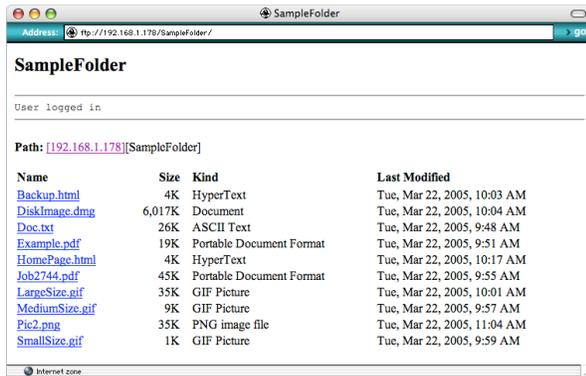


# The Web File Manager

**When clients need to use a Web browser to access your FTP site, use the Web File Manager to provide a more reliable, consistent, and pleasant interface.**



*Typical Web browser access via FTP*

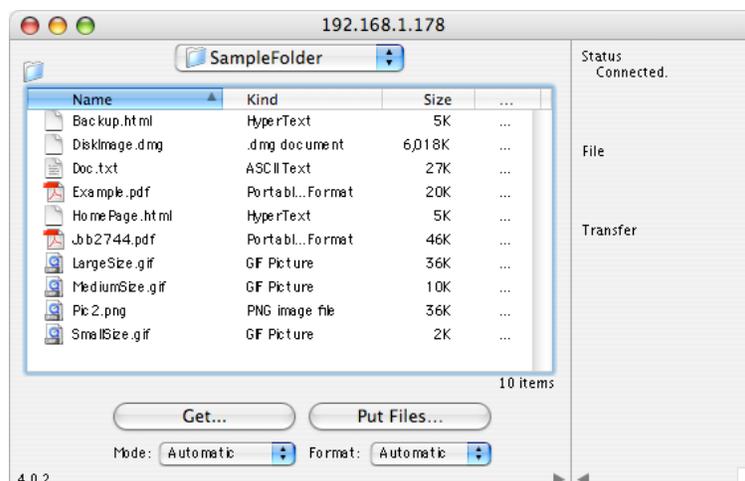


*The same server viewed using the Web File Manager*

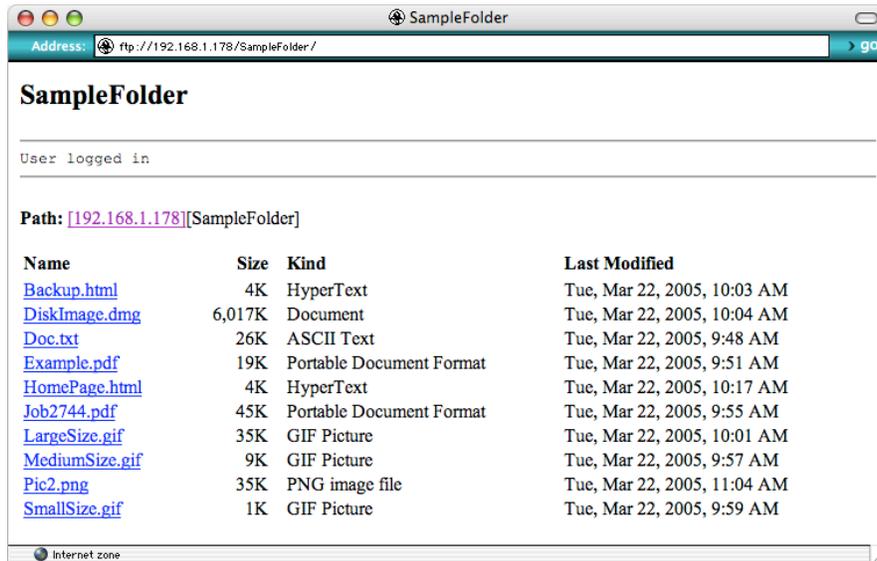
Popular Web browsers have long included basic support for FTP. But because FTP is a secondary task for Web browsers, these clients usually include extremely basic and often bug-ridden FTP implementations. Unfortunately, they are also quite often the most convenient and popular clients for many remote users.

For full-featured, reliable FTP, we strongly recommend that you have your clients use a dedicated FTP client such as Fetch, Transmit, CuteFTP or any number of other inexpensive options. However, in many cases, you will have no choice but to also support Web browsers for file transfers. We have attempted to make Rumpus FTP as Web-browser-friendly as possible, but there is a limit to what can be done to resolve problems in poor client software from the server side. So, Rumpus also includes a purely Web based method of uploading, downloading, and managing files, called the Web File Manager (or WFM, for short).

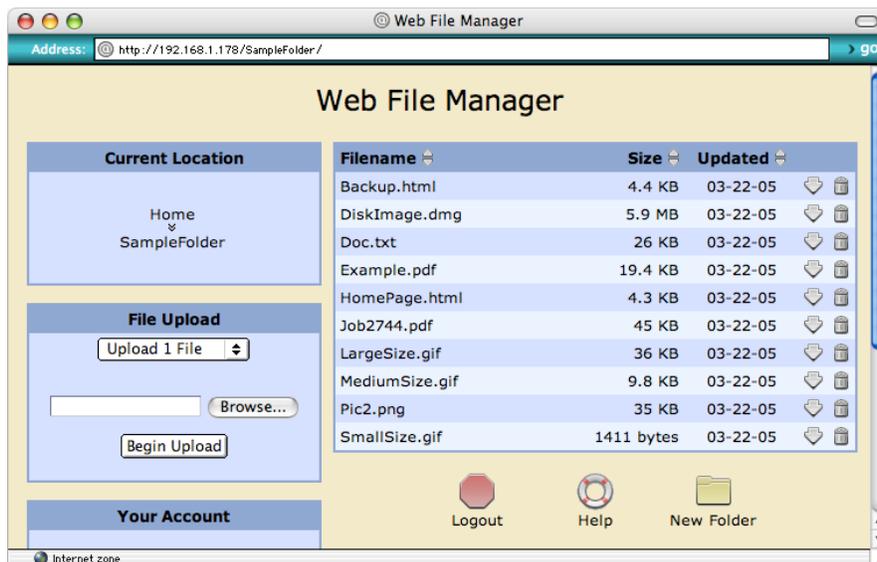
Shown below is a simple FTP site, with an anonymous user logged in using Fetch:



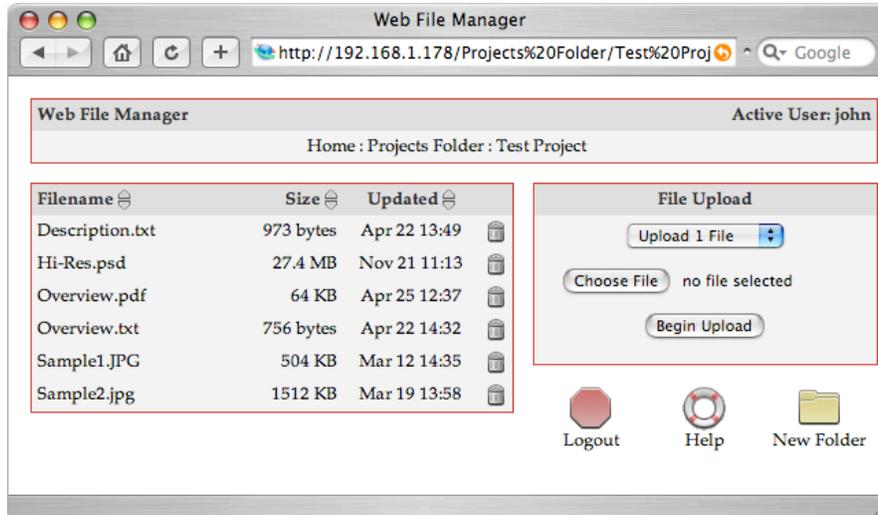
Now, let's see how this same folder looks, with the anonymous user logged in using a Web browser



Not only does the view of your FTP site lack a professional look, but the format of this display is completely out of your hands. Worse yet, some browsers lack features such as the ability to upload files via FTP, or even provide the most basic access to Mac FTP servers. Now, let's look at a view of the same folder in the same browser, but with the user accessing the server through the Rumpus Web File Manager, using the most basic WFM display.



This view offers a better looking interface, and provides a simpler method of secure login with easier file manipulation options. The interface is also fully customizable, as shown below using the alternate "Clean Look" interface included in the Rumpus package, with alternate color scheme and font selections.



Rumpus' WFM actually has very little to do with FTP. When it is enabled, a separate server is activated inside Rumpus to handle HTTP (Web) transactions, and all directory listings, file uploads and downloads, etc. are handled via HTML pages. This not only works around common problems in Web browser FTP implementations, but allows you to present a much nicer and fully-customizable interface to your users.

### **Using the Web File Manager**

The WFM respects all of the user account setup and other security configuration from your FTP site. To use the Rumpus WFM, checking the "enable" checkbox is typically all that's needed. In fact, when Rumpus is first launched, the Setup Assistant allows you to enable the WFM quickly and easily.

Even if you don't use the Setup Assistant to enable the Web File Manager, configuration is simple. To start, open the "Web Settings" window, check the "Enable Web Server" and "Enable Web File Manager" options and choose a port. Note that the standard HTTP port is 80, but Rumpus defaults to 8000 to avoid conflicting with Web services that might already be active on your server. If no other Web server is running, change the port to 80, otherwise, leave the port at 8000 or choose some other suitable port number.

To connect to the server, use a standard HTTP URL in any Web browser, specifying the IP address (or domain name) of the server and the port (if the Rumpus Web server is configured for a port other than the standard port 80). For example:

`http://192.168.1.1:8000/`

In this example, replace "192.168.1.1" with the IP Address or domain name of your server, and replace "8000" with the port number specified in Rumpus. If you set the port number to 80, use the URL:

```
http://192.168.1.1/
```

When using Rumpus' default WFM settings, you will next be presented with the user login page. Enter a valid Rumpus user account name and password to complete the login. If you would prefer to have users immediately connected to the server using the "anonymous" account, uncheck the "Always Prompt For Login" box on the "WFM Options" tab of the "Web Settings" window in Rumpus. Numerous other options are available to customizing the behavior of the Web File Manager. For details on these settings, see the "WFM Settings" help page in the Rumpus application.

The WFM has been designed so that files can be uploaded, downloaded, and managed very easily without a great deal of explanation. A help page is provided, however, that explains how to use the WFM to upload, download, and manage files. Click the "Help" icon below the file listing to access the help page.

### **Customizing The WFM Interface**

The WFM interface can be customized in 2 distinct ways. First, the options available on the "Web Settings" window allow you to change colors, fonts and set the header for the standard WFM look. Alternatively, you can edit the HTML templates yourself, for a completely custom interface. Several example templates are included in the "Extra WFM Templates" folder of the Rumpus package, which you can use as is or as the basis for your own custom Web File Manager.

For details, see the "Customizing The Web File Manager" article in the Rumpus package "Helpful Info" folder.

### **Extended WFM Features**

The basic WFM interface is designed to work well with any reasonably modern Web browser. Convenient WFM features include:

- Multiple file uploads via a single submission
- Progress indicator pop-up windows
- The ability to accept and process additional input fields on the upload form

- The ability to sort directories by name, modification date or size
- A ready made help page
- User account statistics display area
- Separate download links, with “Download Begun” page

Some of these features are not enabled by default, but can be turned on using the options on the “Web Settings” window in Rumpus. Similarly, you may choose to turn off some of the default features, for various reasons. See the “Web Settings” help page in the Rumpus control application for details on each option available.

In particular, note that many of the advanced WFM features implemented by Rumpus rely on client-side javascripts. We have made every attempt to use standard, widely supported javascript in the WFM, but you may find that a few clients have problems with javascript-embedded pages. These clients can be supported by disabling the use of javascripts, though this will, of course, disable WFM features that rely on them.

### **Cookie-Based Login**

Rumpus provides two different mechanisms for user authentication (login). By default, a custom-built login system is used, which is implemented with browser-based cookies. This system allows Rumpus to display a login page that matches the rest of the WFM interface and allows the user to enter their name and password into simple form fields on the page. The cookie-based login has other advantages as well, including user session tracking and more reliable logout.

Cookies are very commonly used on the Web today, and virtually all clients will support the use of cookies and have them enabled. However, if you prefer not to require that clients have cookies enabled, cookie-based login can be turned off. In this case, standard HTTP authentication will be used. Instead of a Web page allowing the user to supply their name and password, a standard dialog box will be opened by the browser when users first connect. Some functions, such as user statistics in Rumpus and user logout, may also not work as reliably, since the users can't be tracked without cookies, and Rumpus has less control over how the browser handles the user session.

**Auto-Login URLs**

At some point, you may have a need to allow a user to log in using a secure user account but bypass the login window. For example, you may wish to send a URL link to someone via e-mail, and have that link complete the login process and take the user immediately to the primary WFM directory listing page. When cookie-based login is enabled, this can be accomplished using a URL of the following form:

```
http://your.rumpus.server:port/?login=username:password
```

Of course, replace “your.rumpus.server” with the name or address of your Rumpus server, and “port” with the WFM port. Essentially, the link is simply the usual WFM access URL, as described above, with “?login=username:password” appended to it, where “username” and “password” represent the actual account name and password you would like the user logged in under.

Obviously, there are security implications to consider when distributing URLs of this form. In particular, anyone that obtains the URL will be able to log in securely through that account, and the name and password are readily visible in plain text. When allowing users the ability to bypass the login process, consider whether using the anonymous user account wouldn’t be a reasonable alternative. And if you do choose to use auto-login URLs, be very careful to restrict the login account as much as possible, and delete the account when it is no longer needed (or have Rumpus automatically expire it).

**Referencing Additional Files**

The Web File Manager also acts as a standard, though very simple, Web server. Files placed in the “WFMTemplates” folder will be served as they would from any Web server. So, for example, images used in your modified WFM templates can be placed in the WFMTemplates folder and linked into the page using the standard “IMG” tag.

Be sure to make all URL references to additional files “root-relative”, which start the URL with a slash (“/”) in the hypertext reference. As the user dives into folders and sub-folders, the browser URL will reflect the folder being viewed, not the WFM template file. You therefore can’t use URLs that are relative to the template file. Hypertext references must instead be relative to the server’s root folder, which, in the case of the Rumpus WFM, is the WFMTemplates folder.

**Multiple Domains With Different WFM Looks**

The Rumpus Web File Manager even supports multiple hosts, each of which can use different template files to provide a customized look for each WFM site. To enable this feature, create a new folder within the selected WFM Templates folder with a name that exactly matches the host name and place the Rumpus template files in the folder. Rumpus will not recognize additional host template folders unless the "Listing.html" file exists in the folder. Not only will Rumpus serve template files from the correct folder based on the host name being accessed, but it will also serve other files in the folder by name.

Secondary host names are limited to 31 characters (the maximum folder name length in Classic Mac OS). You may leave off the preceding "www." portion of a host name, if you choose. For example, Rumpus will correctly recognize a folder named "maxum.com" and serve it's contents for requests specifying "www.maxum.com" as the host.

This feature of Rumpus works by reading the "Host" HTTP request header field, not by the IP address requested. Some very old browsers may not send the host field in the header (though the vast majority of browsers in use today will), in which case Rumpus will serve templates and files from the default WFM Templates folder. In addition, if users access a secondary host using both the IP address and domain name, you will need to create folder entries for both in the Rumpus Prefs folder. Mac OS aliases are supported, allowing you to create one folder with the needed template and additional files and then create an alias to the original folder with the appropriate name. For example, to make a host folder accessible to users accessing the server as both "www.someserver.com" and "192.168.1.23", you would create a folder called "www.someserver.com", populate it with the needed template files, and then create an alias to that folder and name the alias "192.168.1.23".

Rumpus supports up to 15 secondary host folders.

**Accepting User Input Along With File Uploads**

The Rumpus WFM Templates make changing the Web interface relatively easy. You can add images, change the layout, add help text and more to customize the look of the Web File Manager. In addition, you can even add standard HTML input fields to the file upload form, and Rumpus will extract the values and make them accessible in a custom Upload Notice e-mail template.

The first step is to modify the upload forms in the "Listing.html" and "NoListing.html" files. Between the "FORM" and "/FORM" tags, you can add any standard HTML input type, including text fields, checkboxes, radio buttons, and the like. In the examples included in the "Clean Look" Template folder, these pages have file upload forms that include an additional "Comments" text area.

To test out the "user input" capability in Rumpus, open the Web Settings window and click the "choose folder" button next to the "Templates Folder" field, then select the "Clean Look" folder from the "Extra WFM Templates" folder of the Rumpus package. Next, click "Apply Changes" and then the "Reload WFM Templates" button. The next time you access the main listing page, you'll notice that a text input box is included on the file upload form.

The next step is to create an e-mail Upload Notice. Create a new notice, or select an existing one, and make sure all of the fields are set correctly to send an e-mail, as described in the "Upload Notices" article in the Helpful Info folder. Click the "Custom Message Body" button to open the message body sheet, and at the bottom of the body, add a "FORMVAR" tag to include the comments entered by the user on the form. For example, your customized message body might look like this:

```
A file has been uploaded to the Rumpus FTP server.
```

```
File: <file>  
User: <username>  
Address: <address>  
Date: <date>  
Time: <time>
```

```
Comments -
```

```
<FORMVAR Comments>
```

Be sure to set the FORMVAR variable name to "Comments", as that is the name of the input text box included in the "Clean Look" upload form. Again, if you have trouble creating the Upload Notice, see the "Upload Notices" article in the Helpful Info folder.

With the Upload Notice defined, all that's left is to assign the notice to a test user account. When you use that account to log in and upload a file, the text you enter into the Comments field on the upload form will be passed through to the e-mail Upload Notice.

### **"Download Begun" Pages**

The Rumpus WFM includes the option of serving a Web page, rather than the file itself, when a user clicks a file download link. This Web page, the "Download Begun" page, can be used to better present or process the content being served on the browser, or simply to provide better feedback to the user that their download request is being processed. To enable this feature, check the "Present 'Download Begun' Page" option on the "WFM Options" tab of the Web Settings window.

**File Upload Progress Indicator**

When users upload large files, especially over relatively slow Internet connections, file uploads can take considerable time. Because the file being uploaded is part of the HTTP (Web) request sent by the browser, and the fact that a server can't respond to a request until the entire HTTP request has been transmitted, Rumpus is unable to change the Web page to reflect the fact that the transfer has been started. This can cause confusion on the part of the user, as they receive no positive feedback that their file is being uploaded.

Rumpus attempts to provide this user feedback through a couple of optional javascript-based mechanisms. One option is to change the page using HTML divisions. The other, more complicated mechanism is to use javascript to open a second browser window on the client's computer. The second window, which is known as a pop-up window, can then make periodic Web requests which are distinct from the main file transfer. Rumpus then responds to these requests with a status page reflecting the ongoing file transfer. This method of providing feedback is more complicated to implement, and requires both javascript code and special URL handling on the part of Rumpus.

Because they require the use of javascript and pop-up menus, which are not allowed on all clients, progress indicators are not enabled in Rumpus by default. However, for the majority of clients in use the progress indicator will work fine, and can provide valuable feedback for users during long transfers. If you expect that users of your server will be sending large files over relatively slow connections (or even over quick connections), you may find it useful to enable this feature.