

FTP Component

File Transfer Protocol (RFC 959, RFC 1579)

Java™ edition



COPYRIGHT © 2004-2005 WiseSoft Ltd. ALL RIGHTS RESERVED

1. Overview

1.1 Introduction

The FTP (File Transfer Protocol) component provides an interface for transmitting data to/from an FTP server.

1.2 Target audience

This document is intended to serve as a technical guide to using the FTP component. Users should understand the Java programming language and be proficient in writing Java applications. A basic understanding of FTP and networking protocols is also helpful.

1.3 General features

- I** User can connect to any FTP server
- I** Work with directory on server - get current directory name, change working directory, create or remove directory on FTP server
- I** User can get file list of current working directory from FTP server
- I** User can upload a file, download a file, rename a file or remove a file from FTP server
- I** Provides method to disconnect from FTP server

2. Programming Guide

2.1. Creating an instance of FTPClient

Example:

```
FTPClient ftpClient = new FTPClient();;
```

2.2. Configuring host name and port

Example:

```
// This sets host name to "localhost".
String hostName = "localhost";
ftpClient.setHostName(hostName);

// This sets port to 21 (default port is 21)
int port = 21;
ftpClient.setPort(port);
```

2.3. Configuring user name and password

Example:

```
//Sets the user name
String userName = "username";
ftpClient.setUserName(userName);

//Sets the password
String password = "password";
ftpClient.setPassword(password);
```

2.4. Connecting to FTP server

Example:

```
//To connect the ftp server, Returns a Boolean value indicates whether connection process is success or not  
boolean success = ftpClient.connect();
```

2.5. Getting File list from server

Example:

```
String fileList = ftpClient.getFileList();
```

2.6. Getting current working directory from server

Example:

```
//Gets the current working directory of the ftp server  
String directoryOfServer = ftpClient.getCurrentWorkingDirectory();
```

2.7. Changing working directory on server

Example:

```
String directoryName = "/ebooks/java";  
//To change the working directory of the ftp server  
ftpClient.changeWorkingDirectory(directoryName);
```

2.8. Changing to parent directory on server

Example:

```
//To change the current working directory of the ftp server to parent directory  
ftpClient.changeToParentDirectory();
```

2.9. Add a new remote directory on server

Example:

```
//To create a new directory in the ftp server
```

```
ftpClient.createRemoteDirectory ("/ebooks/public");
```

2.10. Removing remote directory from server

Example:

```
//To delete the remove the directory of the ftp server  
ftpClient.removeRemoteDirectory("/ebooks/private");
```

2.11. Uploading local file to server

Example:

```
String remoteFileName = "/ebooks/java/thinkinjava.paf";  
String localPath = "c:/my_ebooks/thinkinjava.pdf";  
//To upload file named thinkinjava.pdf to ftp server  
ftpClient.uploadLocalFile(remoteFileName, localPath);
```

2.12. Downloading remote file

Example:

```
String remotePath = "/ebooksr/java/thinkinjava.pdf";  
String localPath = "c:/my_ebook/thinkinjava.pdf";  
//To download the remote file named thinkjnjava.pdf to local  
ftpClient.downloadRemoteFile(remotePath, localPath);
```

2.13. Renaming remote file

Example:

```
String oldPath = "/ebooks/thinkinjava.pdf";  
String newPath = "/ebooks/think.pdf";  
//To rename the remote file name  
ftpClient.renameRemoteFile(oldPath, newPath);
```

2.14. Deleting remote file

Example:

```
String remotePath = "/ebooks/think.java";  
ftpClient.deleteRemoteFile(remotePath);
```

2.15. Disconnecting from FTP server

Example:

```
//To disconnect  
ftpClient.disconnect();
```