



● **FTP is holding your application back.**

If your software relies on file transfer, developing with an FTP library isn't going to cut it. Latency and packet loss cause TCP-based transfers to choke. Interrupted transfers often fail to resume properly. If you have ever seen transfers using only part of your bandwidth, or have ever pleaded for the connection not to drop, you have felt the pain of FTP. When your end users experience this pain, they will not blame the protocol...they will blame your application.

“ **Using FileCatalyst technology meant that we could integrate features like Accelerated file transfer and HTTP fallback.** ”

—**Richard Rees**, Quicklink CEO

● The FileCatalyst platform provides a wide range of options for integration into 3rd party applications and workflows. Java developers can work directly with the FileCatalyst Client and Server APIs, while other developers can take advantage of the Java applet, .NET wrapper, native C++, Command Line Interface or Web Services. Because the FileCatalyst platform is Java based, integration options run flawlessly on all major platforms including Windows, Mac OSX, Linux, and Solaris.

Why settle for FTP when you can bring the speed, security, and reliability of FileCatalyst to your users?



## Development Kit Components

- **Java:** Client and Server Java APIs
- **C++:** Native client-side API
- **Java Applets:** lightweight, easy to embed
- **.NET Wrapper:** access to the client API becomes transparent to the .NET application.
- **Command Line Interface:** script or control with other programming languages
- **Web Services Interface:** 3rd party applications can remotely instantiate FileCatalyst transfers

## Easily implemented methods

```
● FCClient fc = new FCClient("hostname");  
fc.setUsername("user");  
fc.setPassword("pwd");  
fc.initialize();  
fc.connect();  
TransferHook hook = fc.upload(new File("foo"));  
  
while (!hook.isTransferComplete()) {  
    System.out.println(hook.getPercent());  
    Thread.sleep(500);  
}  
if (hook.isTransferError()) {  
    System.out.println(hook.getErrorMessage());  
}  
fc.disconnect();
```

## features

- Development Kit
  - well documented
  - semantic class and method names
  - easily implemented
  - the most development options of any accelerated file transfer SDK
- FileCatalyst Technology
  - transfer data at full line speed
  - immune to the effects of RTT and packet loss
  - transfer file deltas (rsync-like)
  - on-the-fly compression

## benefits

- Resolves the issues associated with TCP-based file transfer
- Maximizes bandwidth utilization for your end users
- Gives confidence that your file transfer component will be secure and reliable
- Flexible licensing options

“File transfers were susceptible to network impairments, especially when transferring overseas. With minimal effort, we were able to plug in FileCatalyst. The performance gains were immediate.”

● **Steve Emerson**, President, Turning Point Integration



**Request development kit**

[www.filecatalyst.com/trial](http://www.filecatalyst.com/trial)