

# RE: Interrupted FtpWebRequest upload stream

---

*Source:*

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework/2007-04/msg00497.html>

---

- *From:* [wawang@xxxxxxxxxxxxxxxxxxxxxxxx](mailto:wawang@xxxxxxxxxxxxxxxxxxxxxxxx) (Walter Wang [MSFT])
  - *Date:* Thu, 26 Apr 2007 13:08:15 GMT
- 

Hi Dean,

I have done some test on my side and I think the server is really responsible to release the connection, there's really not much we can do from the client side.

Here's my test configuration/steps:

1) I downloaded a ftp server Serv-U from <http://www.serv-u.com> and installed on system B, to ease test, I enabled anonymous user to read/write files on the ftp server

2) On my system A, I created following test program:

=====

```
using System;
using System.Collections.Generic;
using System.Text;
using System.IO;
using System.Net;
using System.Threading;
```

```
namespace ConsoleApplication3
{
class Program
{
static void Main(string[] args)
{
UploadFile(@"f:\test.exe", "ftp://serverB/test.exe, 5000);
Console.WriteLine("Press ENTER");
Console.ReadLine();
}
}
```

```
static bool IsFtpFileExists(string remoteUri, out long remFileSize)
{
FtpWebRequest request =
(FtpWebRequest)WebRequest.Create(remoteUri);
```

RE: Interrupted FtpWebRequest upload stream

FtpWebResponse response:

```
request.Method = WebRequestMethods.Ftp.GetFileSize;  
try  
{  
response = (FtpWebResponse)request.GetResponse();  
remFileSize = response.ContentLength;  
return true;  
}  
catch (WebException we)  
{  
response = we.Response as FtpWebResponse;  
if (response != null && response.StatusCode ==  
FtpStatusCode.ActionNotTakenFileUnavailable)  
{  
remFileSize = 0;  
return false;  
}  
throw;  
}  
}
```

```
static void UploadFile(string localFile, string remoteUri, int  
attempts)  
{  
FileInfo fi = new FileInfo(localFile);  
long remFileSize = 0;  
FtpWebRequest request = null;  
while (--attempts >= 0)  
{  
try  
{  
request = (FtpWebRequest)WebRequest.Create(new  
Uri(remoteUri));  
request.Timeout = 3000;  
if (IsFtpFileExists(remoteUri, out remFileSize))  
{  
Console.WriteLine("File already exists, RESUME");  
request.Method = WebRequestMethods.Ftp.AppendFile;  
}  
else  
{  
Console.WriteLine("File not already exists,  
UPLOAD");  
request.Method = WebRequestMethods.Ftp.UploadFile;  
}  
request.ContentLength = fi.Length - remFileSize;  
request.UsePassive = false;  
  
using (Stream requestStream =  
request.GetRequestStream())
```

RE: Interrupted FtpWebRequest upload stream

```
{  
using (FileStream fs = File.Open(localFile,  
FileMode.Open))  
{  
fs.Seek(remFileSize, SeekOrigin.Begin);  
const int BUFFER_SIZE = 8192;  
byte[] buffer = new byte[BUFFER_SIZE];  
int readBytes = 0;  
do  
{  
readBytes = fs.Read(buffer, 0, BUFFER_SIZE);  
requestStream.Write(buffer, 0, readBytes);  
System.Threading.Thread.Sleep(500);  
} while (readBytes != 0);  
}  
}  
Console.WriteLine("Done");  
using (FtpWebResponse response =  
(FtpWebResponse)request.GetResponse())  
{  
Console.WriteLine(response.StatusCode);  
}  
break;  
}  
catch (WebException we)  
{  
Console.WriteLine(we.Message);  
Thread.Sleep(500);  
}  
}  
}  
}
```

=====

3) Run this test program, it will start upload file f:\test.exe to the ftp server; while it's transferring, disable the network connection "Local Area Connection" to simulate a connection broken error; and re-enable it again after some time. From the ftp server, serv-u administrator shows there's a connection keeps open.

4) Terminate the test program, the connection still opens on the server.

5) Now use another ftp client to test this behavior, you will find the connection is also kept open on the ftp server. Also, in this case, serv-U actually don't use temporary file, therefore you can see the target file exists but while the orphan connection keeps open, you cannot delete or overwrite file --- it's locked by the thread receiving the file on the ftp server.

RE: Interrupted FtpWebRequest upload stream

Since using a commercial ftp client also shows behavior, I think we really cannot do much from client-side. It's totally controlled by the ftp server, the server decides when to times out and dropps the orphan connection.

Regards,

Walter Wang (wawang@xxxxxxxxxxxxxxxxxxxxxx, remove 'online.')

Microsoft Online Community Support

=====  
When responding to posts, please "Reply to Group" via your newsreader so that others may learn and benefit from your issue.  
=====

This posting is provided "AS IS" with no warranties, and confers no rights.

⋮