
Subject: [SOLVED][FeatureRequests]Use HttpRequest to upload large file
Posted by [kasome](#) on Tue, 17 Sep 2013 04:06:16 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

class HttpRequest can be used to upload file by using Post member function, e.g.

```
HttpRequest httpRequest;
Upp::int64 offset = 0;
Upp::String uploadFile = "test.mkv";
httpRequest.SSL(true).Url( someUrl ).ClearPost().ClearHeaders().Post(
Upp::LoadFile(uploadFile).Mid(offset) ).Execute();
```

But when the file is large, larger than 1GB, for example, than HttpRequest will need large memory to upload by using Upp::LoadFile

Is there possible to add feature to HttpRequest, something like:

```
HttpRequest httpRequest;
Upp::int64 offset = 0;
Upp::String uploadFile = "test.mkv";
Upp::FileIn in( uploadFile );
in.Seek(offset);
httpRequest.SSL(true).Url( someUrl ).ClearPost().ClearHeaders().Post( in ).Execute();
```

Thanks.

Subject: Re: [FeatureRequests]Use HttpRequest to upload large file
Posted by [mirek](#) on Sat, 21 Sep 2013 16:56:50 GMT
[View Forum Message](#) <> [Reply to Message](#)

Definitely a good idea, added to RM. Might take a while to implement though... (contributions welcome).

Mirek

Subject: Re: [FeatureRequests]Use HttpRequest to upload large file
Posted by [kasome](#) on Sun, 22 Sep 2013 01:37:47 GMT
[View Forum Message](#) <> [Reply to Message](#)

I see. and Thanks.

Subject: Re: [FeatureRequests]Use HttpRequest to upload large file
Posted by [kasome](#) on Mon, 14 Jul 2014 08:08:10 GMT

[View Forum Message](#) <> [Reply to Message](#)

hi,

Inorder to upload large file, I make some modification to HttpRequest to do so.

Here is the usage:

```
Upp::int64 uploadFileSourceStartPosition = 1000;
Upp::String uploadFileSourcePath = "TestVideo.mkv";
_httpRequest.ClearPost().PUT().SSL( true ).Url( hostName ).Path( hostPath
).ClearHeaders().Header( "Content-Length", contentLength ).KeepAlive( true ).ContentType(
contentType ).PostDataStream( uploadFileSourcePath, uploadFileSourceStartPosition
).Execute();
//_httpRequest.ClearPost().PUT().SSL( true ).Url( hostName ).Path( hostPath
).ClearHeaders().Header( "Content-Length", contentLength ).KeepAlive( true ).ContentType(
contentType ).PostData( Upp::LoadFile(uploadFileSourcePath).Mid(uploadFileSourceStar
tPosition) ).Execute();
```

The total modification is as follows:

1. upp\uppsrc\Core\Inet.h

Original (version:7373)

```
class HttpRequest : public TcpSocket {
public:
.....
HttpRequest& PostData(const String& pd) { postdata = pd; return *this; }
.....
HttpRequest& ClearPost() { PostData(Null); multipart.Clear(); GET(); return
*this; }
.....
};
```

Modified

```
class HttpRequest : public TcpSocket {
.....
int64 postdataPos; // Added
Upp::String postdataName; // Added
.....
bool SendingStream(); // Added
public:
```

```

.....HttpRequest& PostData(const String& pd) { postdata = pd; return *this; }
.....HttpRequest& PostDataStream(const String& pdn, const int64 pos = 0) { postdataName =
pdn; postdataPos = pos; return *this; } // Added

.....// HttpRequest& ClearPost() { PostData(Null); multipart.Clear(); GET(); return
*this; } // Delete
.....HttpRequest& ClearPost() { PostData(Null); PostDataStream(Null);
multipart.Clear(); GET(); return *this; } // Added

....};

```

2. upp\uppsrc\Core\Http.cpp

Original (version:7373)

```

bool HttpRequest::Do()
{
.....case REQUEST:
if(SendingData())
break;
StartPhase(HEADER);
break;

}

void HttpRequest::StartRequest()
{
.....if(method == METHOD_GET || method == METHOD_HEAD)
pd.Clear();

.....if(pd.GetCount() || method == METHOD_POST || method == METHOD_PUT)
data << "Content-Length: " << pd.GetCount() << "\r\n";

}

bool HttpRequest::SendingData()
{
for(;; {
int n = min(2048, data.GetLength() - (int)count);
n = TcpSocket::Put(~data + count, n);
if(n == 0)

```

```

        break;
        count += n;
    }
    return count < data.GetLength();
}

```

Modified

```

bool HttpRequest::Do()
{
.....
case REQUEST:
if( IsNull(postdataName) ) {
    if(SendingData())
        break;
}
else {
    if(SendingStream())
        break;
}
StartPhase(HEADER);
break;
.....
}

void HttpRequest::StartRequest()
{
.....
if(method == METHOD_GET || method == METHOD_HEAD){
    pd.Clear();
    postdataName.Clear();
}
.....
if((IsNull(postdataName)? pd.GetCount() : Upp::GetFileLength(postdataName)-postdataPos) ||
method == METHOD_POST || method == METHOD_PUT)
    data << "Content-Length: " << (IsNull(postdataName)? pd.GetCount() :
Upp::GetFileLength(postdataName)-postdataPos) << "\r\n";
.....
}

bool HttpRequest::SendingData()
{
for(;;) {
    int n = min(2048, data.GetLength() - (int)count);
    n = TcpSocket::Put(~data + count, n);
    if(n == 0)

```

```

        break;
        count += n;
    }
    return count < data.GetLength();
}

bool HttpRequest::SendingStream() {
    Upp::FileIn in( postdataName );
    in.Seek( postdataPos );

    int bufferSize = 2048 - (data.GetLength() & 2047);

    Upp::StringBuffer buffer( bufferSize );
    int readingSize = in.Get( buffer, bufferSize );
    int64 postdataSize = in.GetSize() - postdataPos - readingSize;

    data << Upp::String( ~buffer, readingSize );

    for(;;) {
        int n = min(2048, data.GetLength() - (int)count);
        if( data.GetLength() - (int)count < 0 ) {
            int kk = 0;
        }
        n = TcpSocket::Put(~data + count, n);
        if(n == 0)
            break;
        count += n;
    }

    if( data.GetLength() == count ) {
        Upp::StringBuffer buffer( 2048 );
        for(;;) {
            int n = min((int64)2048, (int64)data.GetLength() + postdataSize - (int64)count);
            int readingSize = in.Get( buffer, n );
            n = TcpSocket::Put(~buffer, readingSize);
            if(n == 0)
                break;
            count += n;
        }
    }
    return count < data.GetLength() + postdataSize;
}

```

Hope that helps.

File Attachments

1) [HttpRequest.txt](#), downloaded 344 times

Subject: Re: [FeatureRequests]Use HttpRequest to upload large file

Posted by [kasome](#) on Wed, 16 Jul 2014 05:39:16 GMT

[View Forum Message](#) <> [Reply to Message](#)

Add some error handling to prevent the program from crashing.

```
bool HttpRequest::SendingStream() {
    Upp::FileIn in( postdataName );
    in.Seek( postdataPos );

    int readingSize = 0;
    Upp::String dataPadding = data;
    if( dataPadding.GetLength() & 2047 ) {
        int bufferSize = 2048 - (dataPadding.GetLength() & 2047);

        Upp::StringBuffer buffer( bufferSize );
        readingSize = in.Get( buffer, bufferSize );
        dataPadding << Upp::String( ~buffer, readingSize );
    }
    int64 postdataSize = in.GetSize() - postdataPos - readingSize;

    if( count == 0 ) {
        for(;;) {
            int n = min(2048, dataPadding.GetLength() - (int)count);
            n = TcpSocket::Put(~dataPadding + count, n);
            if(n == 0)
                break;
            count += n;
        }

        if( (count == dataPadding.GetLength()) && postdataSize ) {
            Upp::StringBuffer buffer( 2048 );
            for(;;) {
                int n = min((int64)2048, (int64)dataPadding.GetLength() + postdataSize - (int64)count);
                int readingSize = in.Get(buffer, n);
                n = TcpSocket::Put(~buffer, readingSize);
                if(n == 0)
                    break;
                count += n;
            }
        }
    }
}
```

```
if( count < dataPadding.GetLength() ) {
    for(;;) {
        int n = min(2048, dataPadding.GetLength() - (int)count);
        n = TcpSocket::Put(~dataPadding + count, n);
        if(n == 0)
            break;
        count += n;
    }
}
if( (count >= dataPadding.GetLength()) && postdataSize ) {
    Upp::StringBuffer buffer( 2048 );
    in.Seek( postdataPos + readingSize + (count - dataPadding.GetLength()) );
    for(;;) {
        int n = min((int64)2048, (int64)dataPadding.GetLength() + postdataSize - (int64)count);
        int readingSize = in.Get(buffer, n);
        n = TcpSocket::Put(~buffer, readingSize);
        if(n == 0)
            break;
        count += n;
    }
}
return count < dataPadding.GetLength() + postdataSize;
}
```

Subject: Re: [FeatureRequests]Use HttpRequest to upload large file

Posted by [mirek](#) on Wed, 16 Jul 2014 06:39:22 GMT

[View Forum Message](#) <> [Reply to Message](#)

Would it be possible to post the whole file here as attachment, please? Much easier to compare it with existing one...

Mirek

Subject: Re: [FeatureRequests]Use HttpRequest to upload large file

Posted by [kasome](#) on Wed, 16 Jul 2014 07:01:45 GMT

[View Forum Message](#) <> [Reply to Message](#)

Sorry. The whole file is post as attachment now. (modified from the original upp version 7373)

File Attachments

- 1) [Inet.h](#), downloaded 316 times
-

2) [Http.cpp](#), downloaded 367 times

Subject: Re: [FeatureRequests]Use HttpRequest to upload large file
Posted by [mirek](#) on Mon, 28 Jul 2014 10:27:17 GMT

[View Forum Message](#) <> [Reply to Message](#)

I think this should be more flexible (not coupled to filesystem), something like

HttpRequest& PostStream(Stream& s, int64 len = Null);

I would also like to ask what is the deal with all that padding code?

Mirek

Subject: Re: [FeatureRequests]Use HttpRequest to upload large file

Posted by [mirek](#) on Mon, 28 Jul 2014 11:17:41 GMT

[View Forum Message](#) <> [Reply to Message](#)

I have tried that altered variant, please check it that is OK.

Mirek

Subject: Re: [FeatureRequests]Use HttpRequest to upload large file

Posted by [kasome](#) on Mon, 28 Jul 2014 11:45:36 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi, Mirek

Quote:

I think this should be more flexible (not coupled to filesystem), something like

HttpRequest& PostStream(Stream& s, int64 len = Null);

Great! That is better than mine. 8)

Quote:I would also like to ask what is the deal with all that padding code?

I am just try to make the two different part of the data combined as if they are the same continuous data, but i am really not sure if it is necessary. :blush:

Quote:I have tried that altered variant, please check it that is OK.

I will check it, and thanks a lot. :p
