Symbian C++ : Multipart/form-data

The following code can be used to upload any file, image, text, video etc.. as a binary file to a server.

The HTTPEngine should have **MHTTPDataSupplier** as its parent class, as the following functions need to be implemented for the upload process

```
TBool GetNextDataPart(TPtrC8& aDataPart);
void ReleaseData();
TInt Reset();
TInt OverallDataSize();
```

The content-type for the post should be, hence add following lines to ClientEngine.h and declare variables under the CClientEngine class

```
_LIT8(KPostContentType, "multipart/form-data; boundary=AaB03x");

TBuf<255>                        iFileName;
 HBufC8    *iPostDataImage;
 HBufC16    *iResponseBuffer;
TBool iDataAvailable
```

In the **MHFRunL** the case **EGotResponseBodyData** should do this

```
MHTTPDataSupplier* dataSupplier = aTransaction.Response().Body();
TPtrC8 ptr;
iDataAvailable = dataSupplier->GetNextDataPart(ptr);


// Convert to 16-bit descriptor
HBufC* buf = HBufC::NewLC(ptr.Length());
buf->Des().Copy(ptr);

// Append to iResponseBuffer
if (iResponseBuffer==NULL)
{
iResponseBuffer = buf->AllocL();
}
else
{
iResponseBuffer = iResponseBuffer->ReAllocL(iResponseBuffer->Length()+buf->Length());
iResponseBuffer->Des().Append(*buf);
}

// Release buf
CleanupStack::PopAndDestroy(buf);

if(!iDataAvailable)
{
 iRunning=EFalse;
 iTransaction.Close();
}
else
{
```


(C) Copyright Nokia 2014. All rights reserved.

```
        dataSupplier->ReleaseData();
}
```

POST code Get the file you want to upload in binary format

```
RFs aFs;
User::LeaveIfError(aFs.Connect());
CleanupClosePushL(aFs);

RFile aFile;

User::LeaveIfError(aFile.Open(aFs, iFileName, EFileShareAny));

TInt aSize;
User::LeaveIfError(aFile.Size(aSize));

iPostDataImage = HBufC8::NewL(aSize);
TPtr8 aPtr = iPostDataImage->Des();

//Obviously we have to read in the data to the iPostDataImage (ray)
aFile.Read(0, aPtr, aSize);

aFile.Close();
aFs.Close();
CleanupStack::PopAndDestroy(&aFs);
```

Using **EFileShareAny** ensures that if the file can be opened by any other application. The file is read into **iPostDataImage** as a binary file.

Now create the upload format

```
_LIT8(KDataStart,"--AaB03x");
_LIT8(KCrlf,"\r\n");
_LIT8(KContent,"Content-Disposition: form-data; name='userfile'; filename='");
_LIT8(KFileCompletion,"'");

_LIT(KContent2,"Content-Type: image/gif");
_LIT(KContent3,"Content-Transfer-Encoding: binary");
_LIT8(KDataEnd,"--AaB03x--");

iPostData = HBufC8::NewL(650+aPtr.Length());


TPtr8 iPostDataPtr = iPostData->Des();
iPostDataPtr.Zero();

iPostDataPtr.Append(KCrlf);
iPostDataPtr.Append(KDataStart);
iPostDataPtr.Append(KCrlf);
iPostDataPtr.Append(KContent);
iPostDataPtr.Append(iFileName);
iPostDataPtr.Append(KFileCompletion);
iPostDataPtr.Append(KCrlf);
iPostDataPtr.Append(KContent2);
iPostDataPtr.Append(KCrlf);
```


(C) Copyright Nokia 2014. All rights reserved.

```
iPostDataPtr.Append(KContent3);
iPostDataPtr.Append(KCrlf);
iPostDataPtr.Append(KCrlf);
iPostDataPtr.Append(aPtr); //the file in binary
iPostDataPtr.Append(KCrlf);
iPostDataPtr.Append(KDataEnd);
iPostDataPtr.Append(KCrlf);
```

Submit the transaction

```
RHTTPHeaders hdr = iTransaction.Request().GetHeaderCollection();
SetHeaderL(hdr, HTTP::EUserAgent, KUserAgent);
SetHeaderL(hdr, HTTP::EAccept, KAccept);
SetHeaderL(hdr, HTTP::EContentType, KPostContentType);

MHTTPDataSupplier* dataSupplier = this;
iTransaction.Request().SetBody(*dataSupplier);

iTransaction.SubmitL();
```

Implement DataSupplier functions

```
TBool CHTTPExampleEngine::GetNextDataPart(TPtrC8& aDataPart)
{
 TBool retVal = EFalse;
    aDataPart.Set(iPostData->Des());
    retVal = (aDataPart.Length() == 0);
    iDataAvailable=retVal;
    return retVal;
}

TInt CHTTPExampleEngine::OverallDataSize()
{
 if(iPostData)
  return iPostData->Length();
 else
  return KErrNotFound ;
}

void CHTTPExampleEngine::ReleaseData()
{
        if(iDataAvailable)
 iTransaction.NotifyNewRequestBodyPartL();
}
```

Any binary file can be uploaded with this upload code.

(C) Copyright Nokia 2014. All rights reserved.

(C) Copyright Nokia 2014. All rights reserved.