

How_to_download_a_file_on_the_device

The following article shows how to use the HTTP Client to download and store a file on the device.

Before the transaction

The file download can be initiated using the GET Method of the HTTP Client. But to store the file being downloaded the following needs to be done: (in the get method itself)

- Create a file on the device.

```
_LIT(KXMLFilePath, "c:\\mytempfiles\\");
iCurrentFileName.Append(_L("default.xml"));
TInt err=iRFileObj.Open(iFsSession,iCurrentFileName,EFileWrite);
if (err==KErrNotFound) // file does not exist - create it
{
    err=iRFileObj.Create(iFsSession,iCurrentFileName,EFileWrite);
}
```

once created the file contents are appended as the download happens.

- Initiate the transaction for the file to download

A file download can happen as a GET to a particular URL, e.g. <http://www.foo.com/myfile.xml>, here the file to download is myfile.xml. This file will be downloaded and stored as default.xml in our example.

During the transaction

Once the GET has started use the following code in **MHFRunL** to store the file

```
case THTTPEvent::EGotResponseBodyData:

// Get text of response body
MHTTPDataSupplier* dataSupplier = aTransaction.Response().Body();
TPtrC8 ptr;
dataSupplier->GetNextDataPart(ptr);

TInt aPos=0;
iRFileObj.Seek(ESeekCurrent, aPos);
iRFileObj.Write(ptr); //save the file being downloaded

HBufC* buf = HBufC::NewLC(ptr.Length());
buf->Des().Copy(ptr);

if (!iResponseBuffer)
{
iResponseBuffer = buf->AllocL();
}
else
{
iResponseBuffer = iResponseBuffer->ReAllocL(iResponseBuffer->Length()+buf->Length());
iResponseBuffer->Des().Copy(*buf);
}
}
```



How_to_download_a_file_on_the_device

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

// Release the body data
dataSupplier->ReleaseData();
```

Now the file has been stored and can be accessed at "C:\\mytempfiles\\default.xml"