Development-Examples-Return a File

To return a file from a hyperlink, the easiest way to seems to be over-riding appendToResponse in a WOComponent to return that file as a response, rather than letting the WOComponent generate it's own response from .html/.wod files:

```
public void appendToResponse(WOResponse aResponse, WOContext aContext) {

    super.appendToResponse(aResponse, aContext);

    //Use whatever mime-type you need for content-type:
    //"text/csv" is just an example.

    aResponse.setHeader("text/csv", "content-type");

    //Assuming 'data' is a Java byte[] array. But
    //You can do whatever you want to wind up with
    //an NSData of content.

    aResponse.setContent( new NSData( data ) );
    aResponse.setHeader("filename=MyFilename.xls", "Content-Disposition");

    //or, if you want the link to "force" a SaveAs dialog...
    //aResponse.setHeader("attachment;filename=MyFilename.xls", "Content-Disposition");
}
```

Someone else reports:
We have the following code for downloading a generated pdf...

```
public void appendToResponse(WOResponse aResponse, WOContext aContext)  {
                super.appendToResponse(aResponse, aContext);
                //aResponse.setHTTPVersion("HTTP/1.1");
                aResponse.disableClientCaching();
                aResponse.removeHeadersForKey("Cache-Control");
                aResponse.removeHeadersForKey("pragma");
                aResponse.setHeader("application/pdf", "content-type");
                aResponse.setHeader("inline; attachment; filename=\"" + fileName + ".pdf\"", "content-
disposition");
                aResponse.setHeader(Integer.toString(resultData.length()), "content-length");
                aResponse.setContent(resultData);
    }
```

The lines of interest for fixing the IE problem (for us at least) were...

```
aResponse.disableClientCaching();
aResponse.removeHeadersForKey("Cache-Control");
aResponse.removeHeadersForKey("pragma");
```

You might want to try:

```
public void appendToResponse( WOResponse r, WOContext c ) {
       fileName = "test.txt";
       r.setHeader( contentType + "; name=\"" + fileName + "\"", "Content-Type" );
       r.setHeader( "inline; filename=\"" + fileName + "\"", "Content-Disposition");
       r.setContent( data );
    }
```

Although according to RFC2183, your code should work fine, I've found that certain browsers from a monopoly that shall nameless looks only to the old RFC1341 location for the filename, which is in the Content-Type header, not in (or at least just in) the Content-Disposition header.

If you're primarily interested in having them download the file, you might want to make it:

```
public void appendToResponse( WOResponse r, WOContext c ) {
       fileName = "test.txt";
       r.setHeader( contentType + "; name=\"" + fileName + "\"", "Content-Type" );
       r.setHeader( "attachment; filename=\"" + fileName + "\"", "Content-Disposition");
       r.setContent( data );
  }
```

And another back-and-forth from the mailing list....

"I got a problem with display PDF file _.. I can save file successfully by click save on Download Message Alert But when I click open, Acrobat run and display the error message of "There is
no such file", failed to open the PDF file."

From: "MacMullin, Jake (DCS)" <MacMullin.Jake@saugov.sa.gov.au>;
To: "'Lu Yanmei'" <lu.yanmei@meta.co.jp>, webobjects-dev@omnigroup.com
Subject: RE: could not open PDF file
Date: Fri, 30 May 2003 09:06:57 +0930

I had the exact same problem. Turned out it has to do with the headers you
use. I've found this combination works best:

```
// set the PDF content and header

  response.setContent(outData);
  response.appendHeader("application/pdf", "Content-Type");
  response.appendHeader(outData.length()+"", "Content-Length");
response.appendHeader("inline;filename=\"file.pdf\"", "Content-Disposition");
```

And another back-and-forth from the mailing list....

I got a problem to display a PDF file in a browser _.. I can send simple text files to open in a browser but pdf shows up as garbage. I cant depend on the user to have a plugin, is there a way to show it in the browser? Either that or is there a way to convert pdf to html and show that in the browser.

```
r.appendHeader("application/pdf", "Content-type");
r.setHeader(Integer.toString(myData.length()), "content-length");
r.appendHeader("application;filename=\"dockconcepts.pdf\"", "Content-Disposition");
r.appendContentCharacter('\n');
r.appendContentData(new NSData(myData));
```

If you want to return a file in response to the submission of a form (and have it work in IE on Windows), you need to chnage the 'method' of the form from 'POST' to 'GET' by adding a 'method' binding to the form in WOBuilder.

I would strongly recommend to use the setHeader method instead of the appendHeader method to make sure that an existing content-type will be overwritten by the file return component. If a content-type already exists, appendHeader will not change the content type value which could lead to unpredictable results. --Hschottm 14:24, 16 January 2007 (UTC)