

# UTF-8 encoding in AJAX web application.

---

*Source:*

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2007-03/msg02595.html>

---

- *From:* "Allan Ebdrup" <[ebdrup@xxxxxxxxxxxxxxxxxx](mailto:ebdrup@xxxxxxxxxxxxxxxxxx)>
  - *Date:* Fri, 16 Mar 2007 15:01:45 +0100
- 

I have an ajax web application where I have problems with UTF-8 encoding of chinese chars.

My Ajax webapplication runs in a HTML page that is UTF-8 Encoded. I copy and paste some chinese chars from another HTML page viewed in IE7, that is also UTF-8 encoded (search for "china" on google.com). I paste the chinese chars into a content editable div. My Ajax webservice compiles an XML where the data from the content editable div is placed in a CDATA section and sends it to a webservice on the server. I read the content editable div using `.innerHTML`. I call the webservice using `XMLHttpRequest` in the following way:

```
-----
req.open("POST", strUrl, true);
req.setRequestHeader("Content-Type", "application/x-www-form-urlencoded;
charset=UTF-8");
var strSend = "";
for(var i=0; i<aParameters.length; i+=2)
{
if(strSend.length!=0) strSend += "&";
strSend += aParameters[i] + "=" + encodeURIComponent(aParameters[i+1]);
}
req.send(strSend);
-----
```

where `req` is the `XMLHttpRequest` object. and `aParameters` is an array that contains: `parameterName, parameterValue, parameterName, parameterValue,...`

Before I send the XML I write it to screen and here the chinese chars are displayed correctly.

On the server I use DotNet 2.0. The XML is transformed to SQL in a CDATA section, that is read and executed against a MSSQL 2000 database, where the string with the chinese chars is stored in a text column.

When I load the data again in my Ajax webapplication a webservice is called that returns the string in an XML in a CDATA section, the data is read from the database using a `DataReader`.

## UTF-8 encoding in AJAX web application.

When the text loaded is displayed the chinese chars have turned into questionmarks.

I've tried to change the column in the database to a image and use a byte-array to fetch the data from the database. that didn't work, so I changed it back.

I've added the following to my web.config:

```
-----  
<globalization  
  requestEncoding="utf-8"  
  responseEncoding="utf-8"  
  fileEncoding="utf-8"  
>  
-----
```

I've changed my ToXml method on my object to the following:

```
-----  
// Define the desired encoding of the output  
System.Text.Encoding encodingOfXmlOutput = System.Text.Encoding.UTF8;  
// Create MemoryStream to receive our bytes  
using (System.IO.MemoryStream memoryStream = new System.IO.MemoryStream())  
{  
  // Create XmlTextWriter using our created memoryStream and  
  encodingOfXmlOutput  
  using (System.Xml.XmlTextWriter xmlWriter = new  
  System.Xml.XmlTextWriter(memoryStream, encodingOfXmlOutput))  
  {  
    // Set formatting options for XmlTextWriter  
    xmlWriter.Formatting = System.Xml.Formatting.None; // Output should not be  
    indented  
    //Write XML  
    xmlWriter.WriteStartElement("Question");  
    xmlWriter.WriteStartElement("QuestionText");  
    xmlWriter.WriteCData(this.Text);  
    xmlWriter.WriteEndElement(); //QuestionText  
    xmlWriter.WriteEndElement(); //Question  
    // Force all bytes into memoryStream  
    xmlWriter.Flush();  
    // Create buffer to receive bytes from memoryStream  
    // Some encodings like UTF-8 contains a preamble (bytes to identify the  
    encoding)  
    // having this preamble in our output will invalidate our output, so we wont  
    be grapping that.  
    byte[] buffer = new byte[memoryStream.Length -  
    encodingOfXmlOutput.GetPreamble().Length];  
    // Position cursor correct in memoryStream (which is after the preamble  
    memoryStream.Position = encodingOfXmlOutput.GetPreamble().Length;  
    // Fill data from current position of memoryStream into buffer  
    memoryStream.Read(buffer, 0, buffer.Length);  
    // Return string of the created Xml  
    return encodingOfXmlOutput.GetString(buffer);  
  }  
}
```

## UTF-8 encoding in AJAX web application.

```
}
```

-----  
Still the same problem.

When I transform the xml to sql I use the following function:

```
-----  
public static string Transform(XslCompiledTransform compiledTransform,  
IXPathNavigable document)  
{  
if (compiledTransform == null) throw new  
ArgumentNullException("compiledTransform");  
using (StringWriter writer = new StringWriter())  
{  
string strResult = string.Empty;  
compiledTransform.Transform(document, null, writer);  
strResult = writer.ToString();  
return strResult;  
}  
}
```

-----  
The XSLT has the following encoding

```
-----  
<?xml version="1.0" encoding="UTF-8"?>  
-----
```

So my question is the following: Where does my encoding screw up? How come I can't save and load chinese chars correctly?

Any pointers would be greatly appreciated.

I don't know what other UTF-8 chars don't work correctly, but the danish chars I initially had problems with (æøå) work correctly, I would like my solution to work with any UTF-8 chars.

Kind Regards,  
Allan Ebdrup

.