
Subject: Re: LoadFile problem with accented chars
Posted by [koldo](#) on Thu, 12 Feb 2009 00:13:52 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello luzr and all

Here I inclose you the "String" version of the functions.

LoadStreamBOM now handles UTF-16 LE & BE, UTF-8 and ISO8859_1 text files and is more optimized but more complex than the first version.

Best regards
Koldo

```
String LoadStreamBOM(Stream& in)
{
    if(in.IsOpen()) {
        in.ClearError();
        int size = (int)in.GetLeft();
        if((dword)size != 0xffffffff) {
            unsigned char header[3];    // Get 3 bytes header
            if (!in.GetAll(&header, 3))
                return String::GetVoid();
            if ((header[0] == 0xFF) && (header[1] == 0xFE)) { // Check header
                StringBuffer s(size-2);    // UTF16 Little Endian
                s[0] = header[2];    // This char is not header
                if (!in.GetAll(s.Begin()+1, size-3))
                    return String::GetVoid();    // Conversion
                return ToUtf8((wchar *)s.Begin(), (size-2)*sizeof(char)/sizeof(wchar));
            } else if ((header[0] == 0xFE) && (header[1] == 0xFF)) {
                StringBuffer s(size-2);    // UTF16 Big Endian
                s[0] = header[2];    // This char is not header
                if (!in.GetAll(s.Begin()+1, size-3))
                    return String::GetVoid();
                for (int i = 0; i < size-2; i += 2) { // Change from big to little endian
                    unsigned char aux = s[i];    // by changing byte order
                    s[i] = s[i+1];
                    s[i+1] = aux;
                }    // Conversion
                return ToUtf8((wchar *)s.Begin(), (size-2)*sizeof(char)/sizeof(wchar));
            } else if ((header[0] == 0xEF) && (header[1] == 0xBB) && (header[2] == 0xBF))
                return in.Get(size-3);    // UTF8. No conversion required
            else {
                StringBuffer s(size);    // Maybe ISO8859-1
                s[0] = header[0];    // Three chars are not header
                s[1] = header[1];    // so inserted into the StringBuffer
                s[2] = header[2];
            }
        }
    }
}
```

```

    if (!in.GetAll(s.Begin()+3, size-3))
        return String::GetVoid();
    return ToUtf8(ToUnicode(s.Begin(), size, CHARSET_ISO8859_1)); // Conversion
}
}
}
return String::GetVoid();
}
String LoadFileBOM(const char *filename)
{
    FileIn in(filename);
    return LoadStreamBOM(in);
}
bool SaveBOMUtf8(Stream& out, const String& data) {
    if(!out.IsOpen() || out.IsError())
        return false;
    unsigned char bom[] = {0xEF, 0xBB, 0xBF};
    out.Put(bom, 3);
    out.Put((const char *)data, data.GetLength());
    out.Close();
    return out.IsOK();
}
bool SaveFileBOMUtf8(const char *path, const String& data)
{
    FileOut out(path);
    return SaveBOMUtf8(out, data);
}

```
