

EOF-Using EOF-Localization and Internationalization

Overview

Using UTF-8 encoding

Place the following code in your WOApplication's dispatchRequest

```
worequest.setContentEncoding(_NSUtilities.UTF8StringEncoding);  
worequest.setHeader("text/html; charset=UTF-8; encoding=UTF-8", "content-type");
```

This will make every page use UTF8 encoding when sending content back to the browser.

Santi

To use UTF-8 encoding you need four steps:

- You must ensure that your database is using UTF-8 as a storing format. It is the default for FrontBase but you need a bit of tweaking for Oracle. If I remember correctly you need to add these to your connection dictionary in EOModeler:

```
NLS_DATE_FORMAT = "YYYY-MM-DD HH24:MI:SS";  
NLS_LANG = AMERICAN_AMERICA.UTF8;  
databaseEncoding = "UTF-8";
```

- With PostgreSQL, to enable UTF-8 support you have to compile it with the

```
--enable-multibyte
```

flag, so you can create a new database specifying the encoding with:

```
createdb -U postgres -E UNICODE
```

- Encode your page in UTF-8. Set it as the default in WO builder and check the encoding by examining the *.wo file inside the *.wo folder.

```
{"WebObjects Release" = "WebObjects 5.0"; encoding = NSUTF8StringEncoding; }
```

If they are not yet encoded in UTF-8 you can copy-paste the above line into your .wo file. Then you can change the encoding of the .wod and .html files (in pbx) by going to the "Format" menu and choosing "file Encodings".

The reason for doing this step is to minimize conversion at run time. If all your component are UTF-8 encoded then WO does not need to do any conversion during the page generation process. I have never actually tested the performance so it may be unwarranted.

- Add the following META to your headers

```
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=UTF-8">
```

- You also need to overwrite Session.appendToResponse and set the encoding to "UTF8"; before calling super, the same applies to takeValuesFromRequest . You may also want to add a meta in the header giving the encoding (I am not sure this is necessary with modern browsers).

```

public void appendToResponse(WOResponse aResponse, WOContext aContext) {
    // set the encoding before anything is garbled
    aResponse.setContentType( _NSUtilities.UTF8StringEncoding );
    super.appendToResponse(aResponse, aContext);
}

public void takeValuesFromRequest(WORequest aRequest, WOContext aContext) {
    aRequest.setDefaultFormValueEncoding( _NSUtilities.UTF8StringEncoding );
    super.takeValuesFromRequest(aRequest, aContext);
}

```

Beware of the two spelling when setting the encoding in a WOMessage the name is "UTF8"; or use the private function `_NSUtilities.UTF8StringEncoding`. Everywhere else it is "UTF-8"

Is this really necessary? Is not enough to add `WOMessage.setDefaultEncoding("UTF8")` in the application constructor?

And by the way, I don't understand well the difference between `WOMessage.defaultEncoding` and `WORequest.defaultFormValueEncoding`

One is about the URL an the other about the data within the form? I'm a bit confused.

David Paules

The first step in modifying the EOModel's connection dictionary was unnecessary for me. I used OpenBase as my backend database. OpenBase allows you to create a database with one of n different encodings (UTF8 is one of them). The EO Framework was capable of storing and retrieving UTF8 character data from OpenBase without any customization of the connection dictionary (ie, the connection dictionary only had the jdbc driver and database URL and user/password).

Since you must modify the session's `appendToResponse` method, you can simplify the process of adding the META tags to all your pages by doing it in code once. Simply add

```
aResponse.setHeader("text/html; charset=UTF-8; encoding=UTF-8", "content-type");
```

before calling

```
super.appendToResponse(aResponse, aContext);
```

Another person adds: Note, this is NOT the same as adding a meta tag to all your pages at once. It's adding a header to all your responses. But that should be just as good, or better. The META tag is really just meant to simulate the header.

For WebObjects 4.x using ObjectiveC you will need to following code in the WOComponent class:

```

-(void) appendToResponse: (WOResponse *) r inContext: (WOContext *) c
{
    [r setHeader:@"text-html. charsetUTF-8. encoding=UTF8". forKey:@"Content-type"];
    [r setContentType:NSUTF8StringEncoding];
    [super appendToResponse:r inContext:c];
}

-(void) takeValuesFromRequest: (WORequest *) r inContext: (WOContext *) c
{
    [r setFormValueEncodingDetectionEnabled:NO];
    [r setDefaultFormValueEncoding:NSUTF8StringEncoding];
    [super takeValuesFromRequest:r inContext:c];
}

```

You also need to disable the encoding detection... at least I had to do it in my case.