
Subject: Re: Oracle8: UTF8 charset Patch

Posted by [rylek](#) on Mon, 12 Jul 2010 23:20:32 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi there!

After playing with one of my applications which started to malfunction after Mirek applied tojocky's patch I found out that the patch doesn't solve all related problems and in fact creates new ones. In my commercial practice I write many Oracle-based database applications for Czech companies and institutions and it's quite natural to have the default application charset set to WIN1250. The databases on the servers have the same codepage so no conversion is necessary.

Now, after Mirek applied tojocky's extension, the new OCI9 client method OCIEEnvNlsCreate lets the OCI client to switch to UTF8 encoding (when OCI9 is available). However, when the application character set is set to something else, the necessary conversions were not being made which caused such applications to stop working (in a rather erratic way, depending on which version of the Oracle client was installed on the client computer).

I have written and posted on SVN two fixes which should get rid of the problem. First is a simple extension of the OCI8 client code which remembers (using the utf8_session member variable) whether the UTF8 communication has been established during Login and performs the necessary character set conversions when the application character set is not UTF8. I have found and patched four situations in which character set conversion has to be done:

- 1) passing the statement to OCI parser during Execute
- 2) passing input statement parameters (SetParam)
- 3) retrieving output column values (GetColumn)
- 4) querying error message descriptive texts (OciError)

However, a tricky situation occurs with BLOB / CLOB's. There are two ways to retrieve a LOB during Fetch. Either you can GetColumn with a String& parameter, or you can retrieve the integer LOB descriptor and explicitly use OracleBlob to read the LOB contents using Stream interface. The former case was easy to fix and I did.

However, fixing the latter case would be very difficult because it would require the OCI client to somehow combine the LOB descriptor with a flag saying whether this is a BLOB or a CLOB and the OracleBlob Stream interface would have to automatically convert the character set on-the-fly. I believe it would be perverse to do such a thing silently so I've left OracleBlob as it is, a simple binary stream directly addressing the LOB data. However, when an existing application uses e.g. LoadStream to read a CLOB, the tojocky's UTF8 patch effectively breaks encoding in the fetched LOB. For this reason I've added a pair of member functions to Oracle8:

```
void DisableUtf8Mode(bool dutf8 = true);  
bool IsUtf8Session() const;
```

DisableUtf8Mode turns off the new OCIEEnvNlsCreate-related code and leaves the client

communicating in the native character set; IsUtf8Session lets the application to check the connection mode and possibly perform the character conversion when reading / writing a CLOB stream using OracleBlob directly.

Regards

Tomas
