Subject: Oracle8: UTF8 charset Patch
Posted by tojocky on Fri, 28 May 2010 11:59:45 GMT
View Forum Message <> Reply to Message

Hello all,

I propose to apply this Oracle patch for use UTF-8 (AL32UTF8) or UTF8 charset encoding, acording by this documentation. I read legal documentation but without success.

Ok, the patch is (with long raw patch):

```
Index: uppsrc/Oracle/Oci8.cpp
===================================================================
--- uppsrc/Oracle/Oci8.cpp (revision 2421)
+++ uppsrc/Oracle/Oci8.cpp (working copy)
@@ -69,7 +69,7 @@
  struct Item {
   T_OCI8&        oci8;
   int            type;
-  int16          len;
+  int            len;
   sb2            ind;
   ub2            rl;
   ub2            rc;
@@ -142,6 +142,7 @@
  void      SetParam(int i, Date d);
  void      SetParam(int i, Time d);
  void      SetParam(int i, Sql& refcursor);
+ void      SetRawParam(int i, const String& s);

  void      AddColumn(int type, int len);
  void      GetColumn(int i, String& s) const;
@@ -296,6 +297,13 @@
  PrepareParam(i, r.GetOraType(), r.GetMaxLen(), 0, r.GetType());
 }

+void OCI8Connection::SetRawParam(int i, const String& s) {
+ int l = s.GetLength();
+ Item& p = PrepareParam(i, SQLT_LBI, l, 0, VOID_V);
+ memcpy(p.Data(), s, l);
+ p.ind = l ? 0 : -1;
+}
+
 class Oracle8RefCursorStub : public SqlSource {
 public:
  Oracle8RefCursorStub(SqlConnection *cn) : cn(cn) {}
@@ -320,6 +328,9 @@
   SetParam(i, String());
  else
```

```
   switch(q.GetType()) {
+  case SQLRAW_V:
+   SetRawParam(i, SqlRaw(q));
+   break;
   case STRING_V:
   case WSTRING_V:
    SetParam(i, String(q));
@@ -842,21 +853,36 @@
   | OCI_THREADED
 #endif
  ;
- if(oci8.OCIEnvCreate) {
-  if(oci8.OCIEnvCreate(&envhp, accessmode, 0, 0, 0, 0, 0, 0)) {
-   OCIInitError(*this, "OCIEnvCreate");
-   return false;
-  }
+
+ // ILUPASCU(tojocky): try to put utf8 charset
+ bool v_without_NLS = true;
+
+ if(oci8.OCIEnvNlsCreate) {
+  if((v_without_NLS)&&(oci8.OCIEnvNlsCreate(&envhp, accessmode, 0, 0, 0, 0, 0, 0,
OCI_NLS_NCHARSET_ID_UT8, OCI_NLS_NCHARSET_ID_UT8))){
+   LLOG("OCI8: error on initialization utf8 NLS");
+   v_without_NLS = true;
+  }else v_without_NLS = false;
+  if((v_without_NLS)&&(oci8.OCIEnvNlsCreate(&envhp, accessmode, 0, 0, 0, 0, 0, 0,
OCI_NLS_NCHARSET_ID_AL32UT8, OCI_NLS_NCHARSET_ID_AL32UT8))){
+   LLOG("OCI8: error on initialization utf8 NLS");
+   v_without_NLS = true;
+  }else v_without_NLS = false;
  }
- else {
-  if(oci8.OCIInitialize(accessmode, 0, 0, 0, 0)) {
-   OCIInitError(*this, "OCIInitialize");
-   return false;
+ if(v_without_NLS){
+  if(oci8.OCIEnvCreate){
+   if(oci8.OCIEnvCreate(&envhp, accessmode, 0, 0, 0, 0, 0, 0)) {
+    OCIInitError(*this, "OCIEnvCreate");
+    return false;
+   }
+  } else {
+   if(oci8.OCIInitialize(accessmode, 0, 0, 0, 0)) {
+    OCIInitError(*this, "OCIInitialize");
+    return false;
+   }
+   if(oci8.OCIEnvInit(&envhp, OCI_DEFAULT, 0, 0)) {
```

```
+    OCIInitError(*this, "OCIEnvInit");
+    return false;
+   }
   }
-  if(oci8.OCIEnvInit(&envhp, OCI_DEFAULT, 0, 0)) {
-   OCIInitError(*this, "OCIEnvInit");
-   return false;
-  }
  }
  if(!AllocOciHandle(&errhp, OCI_HTYPE_ERROR)) {
   OCIInitError(*this, "OCI_HTYPE_ERROR");
Index: uppsrc/Oracle/Oci8.dli
===================================================================
--- uppsrc/Oracle/Oci8.dli (revision 2421)
+++ uppsrc/Oracle/Oci8.dli (working copy)
@@ -18,6 +18,19 @@

 FN(sword,   OCIEnvInit, (UPP::OCIEnv **envp, ub4 mode, size_t xtramem_sz, dvoid
**usrmempp))

+FN0(sword, OCIEnvNlsCreate, (OCIEnv **envp, ub4 mode, const dvoid *ctxp, \
+ const dvoid *(*malocfp)(dvoid *ctxp, size_t size), \
+ const dvoid *(*ralocfp)(dvoid *ctxp, dvoid *memptr, size_t newsize), \
+ const void (*mfreexp)(dvoid *ctxp, dvoid *memptr), \
+ size_t xtramem_sz, dvoid **usrmempp, ub2 charset, ub2 ncharset))
+
+FN0(sword, OCINlsEnvironmentVariableGet, (dvoid *val, size_t size, ub2 item,\
+ ub2 charset, size_t *rsize))
+
+FN0(ub2, OCINlsCharSetNameToId, (dvoid *hndl, const OraText *name))
+
+FN0(sword, OCINlsCharSetIdToName, (dvoid *hndl, OraText *buf, size_t buflen, ub2 id))
+
 FN0(sword,   OCIEnvCreate, (OCIEnv **envp, ub4 mode, const dvoid *ctxp, \
  const dvoid *(*malocfp)(dvoid *ctxp, size_t size), \
  const dvoid *(*ralocfp)(dvoid *ctxp, dvoid *memptr, size_t newsize), \
Index: uppsrc/Oracle/OciCommon.h
===================================================================
--- uppsrc/Oracle/OciCommon.h (revision 2421)
+++ uppsrc/Oracle/OciCommon.h (working copy)
@@ -36,6 +36,9 @@
   // The following *TWO* are only valid for OCICreateEnvironment call
  OCI_NO_UCB     = 0x40,  // No user callback called during init
  OCI_NO_MUTEX   = 0x80,  // the environment handle will not be
+  // NLS
+ OCI_NLS_NCHARSET_ID_UT8 = 871,                 /* AL32UTF8 charset id */
+ OCI_NLS_NCHARSET_ID_AL32UT8 = 873,             /* AL32UTF8 charset id */
   // protected by a mutex internally
```

```
  OCI_SHARED_EXT = 0x100, // Used for shared forms
  OCI_CACHE      = 0x200, // used by iCache
```

I hope to be helpful!

File Attachments

1) oracle_utf8_charset_patch.patch, downloaded 396 times

Subject: Re: Oracle8: UTF8 charset Patch
Posted by mirek on Fri, 28 May 2010 12:54:15 GMT
View Forum Message <> Reply to Message

I am sorry, but can you just add plain files (whole files, no patches) please?

Subject: Re: Oracle8: UTF8 charset Patch
Posted by tojocky on Fri, 28 May 2010 17:15:13 GMT
View Forum Message <> Reply to Message

luzr wrote on Fri, 28 May 2010 15:54I am sorry, but can you just add plain files (whole files, no patches) please?

Sure,

File Attachments

1) Oracle.7z, downloaded 344 times

Subject: Re: Oracle8: UTF8 charset Patch
Posted by tojocky on Fri, 28 May 2010 20:19:48 GMT
View Forum Message <> Reply to Message

luzr wrote on Fri, 28 May 2010 15:54I am sorry, but can you just add plain files (whole files, no patches) please?

Mirek, Can you change with places OCI_NLS_NCHARSET_ID_AL32UT8 and OCI_NLS_NCHARSET_ID_UT8 like this:

```
 if(oci8.OCIEnvNlsCreate) {
  if((v_without_NLS)&&(oci8.OCIEnvNlsCreate(&envhp, accessmode, 0, 0, 0, 0, 0, 0,
OCI_NLS_NCHARSET_ID_AL32UT8, OCI_NLS_NCHARSET_ID_AL32UT8))){
   LLOG("OCI8: error on initialization utf8 NLS");
   v_without_NLS = true;
```

```
  }else v_without_NLS = false;
  if((v_without_NLS)&&(oci8.OCIEnvNlsCreate(&envhp, accessmode, 0, 0, 0, 0, 0, 0,
OCI_NLS_NCHARSET_ID_UT8, OCI_NLS_NCHARSET_ID_UT8))){
   LLOG("OCI8: error on initialization utf8 NLS");
   v_without_NLS = true;
  }else v_without_NLS = false;
 }
```

The problem is: charset AL32UT8 is created in 9.1 database release, but UTF8 is created in 8.0. UTF8 it is not a real Unicode charset.

Subject: Re: Oracle8: UTF8 charset Patch
Posted by mirek on Sun, 30 May 2010 19:15:49 GMT
View Forum Message <> Reply to Message

Applied, thanks.

Mirek

Subject: Re: Oracle8: UTF8 charset Patch
Posted by tojocky on Mon, 31 May 2010 06:27:05 GMT
View Forum Message <> Reply to Message

luzr wrote on Sun, 30 May 2010 22:15Applied, thanks.

Mirek

Thank you!

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 OCIEnvNlsCreate 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 OCIEnvNlsCreate-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

Subject: Re: Oracle8: UTF8 charset Patch
Posted by tojocky on Tue, 13 Jul 2010 06:36:17 GMT
View Forum Message <> Reply to Message

rylek wrote on Tue, 13 July 2010 02:20Hi 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.

Hello Tomas,

Thank you for testing my patch and improve the functionality.

Quote:have the default application charset set to WIN1250. The databases on the servers have the same codepage so no conversion is necessary.

When you load the library and try to connect to the database you do not know the DB encoding! The solution I found to set utf8 charset.

Your patch is very useful. My default charset of application is UTF8 and I did not need to convert the data to/from utf8. Thanks!

But I can't understand:

```
void  OCI8Connection::SetParam(int i, const Value& q) {
.................
case STRING_V:
    SetParam(i, WString(q));
    break;
   case WSTRING_V:
    SetParam(i, String(q));
    break;
.....................
}
```

You convert into string if type is WString and vice-versa. Why?
Quote:DisableUtf8Mode turns off the new OCIEnvNlsCreate-related code and leaves the client communicating in the native character set

How to prevent the user if he tries to DisableUtf8Mode after connected. Nothing happening?

The problems with Oracle 8 is: the charset OCI_NLS_NCHARSET_ID_UT8 is not true UTF8 and may create some problems. I test my patch with Oracle 10 (uses OCI_NLS_NCHARSET_ID_AL32UT8) and works fine with my default charset of application UTF8!

Thanks for patch.

Subject: Re: Oracle8: UTF8 charset Patch
Posted by rylek on Tue, 13 Jul 2010 23:39:30 GMT
View Forum Message <> Reply to Message

Hi Tojocky!

Thanks, that was just a silly typo, of course the cases should match (WSTRING_V with WString and STRING_V with String). As concerns database character set: if I remember the documentation right, you normally don't need to know the database character set at all; it's considered a part of the internal communication protocol between the client and the server. The client receives the data from the server (in some encoding) and converts it to a "default" client-side encoding set up during client installation (when it's not initialized with OCIEnvNlsCreate).

I agree that using the UTF8 encoding is more general as it's independent of the client installation settings and can store Unicode characters, I just believe that in practice a specific territory-dependent character set is quite sufficient when you know the database / client encoding in advance.

As concerns DisableUtf8Mode(), I see it as an ugly hack for special cases and as one which ipso facto requires a programmer knowing it must be used before Login. It might ASSERT that the session is not open, all right.

Regards

Tomas

Subject: Re: Oracle8: UTF8 charset Patch
Posted by tojocky on Wed, 14 Jul 2010 06:56:10 GMT
View Forum Message <> Reply to Message

rylek wrote on Wed, 14 July 2010 02:39As concerns database character set: if I remember the documentation right, you normally don't need to know the database character set at all; it's considered a part of the internal communication protocol between the client and the server. The client receives the data from the server (in some encoding) and converts it to a "default" client-side encoding set up during client installation (when it's not initialized with OCIEnvNlsCreate).

I agree that using the UTF8 encoding is more general as it's independent of the client installation settings and can store Unicode characters, I just believe that in practice a specific territory-dependent character set is quite sufficient when you know the database / client encoding in advance.

As concerns DisableUtf8Mode(), I see it as an ugly hack for special cases and as one which ipso facto requires a programmer knowing it must be used before Login. It might ASSERT that the session is not open, all right.

Regards

Tomas

Tomas,

I used this patch for load oci.dll/so without install oracle client. Just provide the library with application.

About DisableUtf8Mode(), I thing is not ASSERT that the session is not open. It is not a system error and not need force to close the application.

Best Regard, Ion

Subject: Re: Oracle8: UTF8 charset Patch
Posted by rylek on Wed, 21 Jul 2010 20:48:38 GMT
View Forum Message <> Reply to Message

Hello Tojocky!

I can easily live without an ASSERT in DisableUtf8Mode (I originally proposed it because I believed you had asked for it in an earlier message). I, however, don't think the main reason is that it's not a "system error". The main purpose of ASSERT / NEVER, I believe, is primarily not to catch situations which would otherwise trigger a hard error and application abort, but to bang the programmer over the head that they're doing something wrong in the sense they're using interfaces as they shouldn't. Which, I believe, includes calling DisableUtf8Mode at a wrong time, i.e. after establishing a connection.

But that's only a minor philosophical contemplation. What's worse, I've encountered a new error in a commercial database application of mine (WinZPV, surface water data processing I wrote for the Czech Hydrological Institute), which I found out to be related to our UTF8 patches.

The cause of trouble is of course the damned CLOB. I might understand it but I still hate it, however OCI8 interface counts CLOB length in characters, not bytes. Namely, OCILobGetLength returns number of characters in a CLOB and OCILobRead reads given number of characters at a given character offset.

This, of course, wreaks complete havoc when using UTF8 encoding, because the relation between byte and character offsets and lengths is not linear. Currently the only way around this problem I see is dividing OracleBlob into two classes, OracleBlob and OracleClob, where OracleClob would map the CLOB to Unicode byte pairs. This allows directly mapping the character-based count/offset OCI functions to file offsets (* 2) and random access to the input stream.

OracleBlob should then ASSERT that the handle it processes is really a BLOB, not a CLOB. Or it should do so at least in the UTF8 mode; in a "native encoding" mode this is not necessary, because the character set width is fixed and the relation "character = byte" holds.

Some internal hackery would of course be necessary to read the UTF8 stream data into an intermediate buffer and convert them on-the-fly to Unicode which would then be fed into the stream read queue. A similar procedure would have to be applied on output.

To be honest, I currently have no energy to write this (although I accept readiness to do it sometimes next week), moreover I believe this to be a partially backwards-incompatible modification and therefore I would appreciate your opinion and perhaps others' as well, at least Mirek's. Perhaps you can devise a smarter way to get along with things as they are, I can't.

Regards

Tomas

Subject: Re: Oracle8: UTF8 charset Patch
Posted by mirek on Wed, 21 Jul 2010 21:49:57 GMT
View Forum Message <> Reply to Message

rylek wrote on Wed, 21 July 2010 16:48therefore I would appreciate your opinion and perhaps others' as well, at least Mirek's

I totally support Tom's position.

Mirek

Subject: Re: Oracle8: UTF8 charset Patch
Posted by tojocky on Mon, 26 Jul 2010 16:04:03 GMT
View Forum Message <> Reply to Message

sorry for delay Tomas!

Quote: Perhaps you can devise a smarter way to get along with things as they are, I can't.
I didn't work with BLOB and CLOB in U++, but:
Quote:dividing OracleBlob into two classes, OracleBlob and OracleClob, where OracleClob would map the CLOB to Unicode byte pairs. This allows directly mapping the character-based count/offset OCI functions to file offsets (* 2) and random access to the input stream.
seems to be a nice idea!
Quote:I would appreciate your opinion
Thank you for your effort!

Best Regards, Ion.
