Subject: Postgres library improvements
Posted by Zbych on Mon, 27 Jul 2009 08:44:41 GMT
View Forum Message <> Reply to Message

1. Client code page

Functions PostgreSQLConnection::ErrorMessage and PostgreSQLSession::ErrorMessage use FromSystemCharset to convert code page. Unfortunately PQerrorMessage AFAIK does not use client's system code page but the code page that is declared in lc_messages (postgresql.conf), or the code page explicit set by the client (function PQsetClientEncoding).

So I think those functions should be modified:

```
String PostgreSQLConnection::ErrorMessage()
{
 // no code page conversion, leave it to postgres
 return AsString(PQerrorMessage(conn));
}

String PostgreSQLSession::ErrorMessage()
{
 // no code page conversion, leave it to postgres
 return AsString(PQerrorMessage(conn));
}
```

And of course postgres should be informed about the code page:

```
bool PostgreSQLSession::Open(const char *connect)
{
 Close();
 conn = PQconnectdb(connect);
 if(PQstatus(conn) != CONNECTION_OK)
 {
  SetError(ErrorMessage(), "Opening database");
  Close();
  return false;
 }
 level = 0;

 // set client's code page
 int stat = PQsetClientEncoding(conn, CharsetName(GetDefaultCharset()));
 ASSERT(stat == 0);
 LOG( String("Postgresql client encoding: ") + pg_encoding_to_char( PQclientEncoding(conn) ) );

 return true;
```

```
}
```

I made some tests with UTF-8 and WIN1250, and it appears that postgres accepts code page names from ultimate

2. Re-establishing connection

It would be nice to have function that reconnects to database, when connection is lost:

```
bool PostgreSQLSession::ReOpen()
{
 PQreset(conn);
 if(PQstatus(conn) != CONNECTION_OK)
 {
  SetError(ErrorMessage(), "Opening database");
  return false;
 }
 level = 0;

 return true;
}
```

Subject: Re: Postgres library improvements
Posted by mirek on Wed, 29 Jul 2009 06:40:26 GMT
View Forum Message <> Reply to Message

Thanks, applied. I hope this will not cause any trouble (but it seems quite reasonable).

Mirek

Subject: Re: Postgres library improvements
Posted by Zbych on Wed, 29 Jul 2009 12:52:48 GMT
View Forum Message <> Reply to Message

luzr wrote on Wed, 29 July 2009 08:40Thanks, applied. I hope this will not cause any trouble (but it seems quite reasonable)

I see two possible sources of troubles:

1. not all code page names from CharsetName() are accepted by postgres (that is why I've added ASSERT). Code page names: iso8859-xx, windows-125x, koi8-r, UTF-8 are accepted and cp852, mjk, cp850 are rejected.

2. I don't know what code page is used before connection is established (in my case all messages at this time are in english, probably because of this bug).

Subject: Re: Postgres library improvements
Posted by Zbych on Thu, 30 Jul 2009 20:12:32 GMT
View Forum Message <> Reply to Message

I've made some additional tests and it appears that code page and client language are one big mess in windows version of postgresql:

1. Before client is connected, error messages (for example when server is down) use win-1250 code page (environment variable LC_LOCALE is ignored, bug)

2. When client tries to connect, but connection is refused (for example when client has no right to connect to database), error messages use code page and language set in configuration of server (in my case utf-8)

3. When client is connected and code page is set by PQsetClientEncoding, error messages use correct code page.

I tried to fix the problem:

```
String PostgreSQLSession::ErrorMessage()
{
   if (PQclientEncoding(conn) >= 0) return PQerrorMessage(conn);
   return FromSystemCharset(PQerrorMessage(conn));
}
```

but it only solves case 1 i 3.

Any ideas how to handle this correctly?

Subject: Re: Postgres library improvements
Posted by mirek on Fri, 31 Jul 2009 08:22:26 GMT
View Forum Message <> Reply to Message

Zbych wrote on Thu, 30 July 2009 16:12I've made some additional tests and it appears that code page and client language are one big mess in windows version of postgresql:

1. Before client is connected, error messages (for example when server is down) use win-1250 code page (environment variable LC_LOCALE is ignored, bug)

2. When client tries to connect, but connection is refused (for example when client has no right to connect to database), error messages use code page and language set in configuration of server (in my case utf-8)

3. When client is connected and code page is set by PQsetClientEncoding, error messages use correct code page.

I tried to fix the problem:

```
String PostgreSQLSession::ErrorMessage()
{
   if (PQclientEncoding(conn) >= 0) return PQerrorMessage(conn);
   return FromSystemCharset(PQerrorMessage(conn));
}
```

but it only solves case 1 i 3.

Any ideas how to handle this correctly?

For now, patch applied.

I guess that perhaps the issue is not that serious. error messages tend to be in basic ascii, especially connection messages.

Moreover, you do not expect regular users to understand them anyway...

Mirek

Subject: Re: Postgres library improvements
Posted by mirek on Fri, 04 Sep 2009 12:24:50 GMT
View Forum Message <> Reply to Message

So it happened, we achieved to broke important existing application...

The reason is handlong of SQL_ASCII.

I have reworked the whole issue:

- now, the db encoding is detected first.
- if it is SQL_ASCII, no further conversions are performed.
- if it is anything else, we set encoding to UTF-8 and convert everything from us to UTF-8 and back
- to workaround the rest, there is now SetCharset method to force it

Mirek

Subject: Re: Postgres library improvements
Posted by Zbych on Tue, 08 Sep 2009 14:02:41 GMT
View Forum Message <> Reply to Message

luzr wrote on Fri, 04 September 2009 14:24So it happened, we achieved to broke important existing application...

Sorry to hear that.
But your solution does the conversion twice (first one on database server side, and the second one in the application). To be honest, I don't see much sense in this solution. It would be much easier to give user a choice by adding new parameter to PostgreSQLSession::Open(const char *connect, bool autoconvert = false);
By default conversion can be turned off (for compatibility with older applications).

Another problem is in ErrorMessage function:

Now:
```
{
 if(PQclientEncoding(conn) >= 0)
  /* Client is connected, you should use FromCharset here */
  return PQerrorMessage(conn);
 /* Client is disconnected, use system code page */
 return FromCharset(PQerrorMessage(conn));
}
```

Proposition:
```
{
 if(PQclientEncoding(conn) >= 0)
  return FromCharset(PQerrorMessage(conn))
 return FromSystemCharset(PQerrorMessage(conn));
}
```

Subject: Re: Postgres library improvements
Posted by mirek on Thu, 10 Sep 2009 08:12:55 GMT
View Forum Message <> Reply to Message

Zbych wrote on Tue, 08 September 2009 10:02luzr wrote on Fri, 04 September 2009 14:24So it happened, we achieved to broke important existing application...

Sorry to hear that.
But your solution does the conversion twice (first one on database server side, and the second one in the application).

Well, I bet on the fact that most apps will use either UTF-8 or ASCII in DB and UTF-8 in app.

Subject: Re: Postgres library improvements
Posted by mirek on Thu, 10 Sep 2009 08:20:53 GMT
View Forum Message <> Reply to Message

Zbych wrote on Tue, 08 September 2009 10:02
Another problem is in ErrorMessage function:

Now:
```
{
 if(PQclientEncoding(conn) >= 0)
  /* Client is connected, you should use FromCharset here */
  return PQerrorMessage(conn);
 /* Client is disconnected, use system code page */
 return FromCharset(PQerrorMessage(conn));
}
```

Proposition:
```
{
 if(PQclientEncoding(conn) >= 0)
  return FromCharset(PQerrorMessage(conn))
 return FromSystemCharset(PQerrorMessage(conn));
}
```

Yes, thanks. Actually, I believe this fix will do:

```
String PostgreSQLConnection::ErrorMessage()
{
 return FromCharset(PQerrorMessage(conn));
}
```

and not using ErrorMessage for connection failure.

Mirek
