Subject: bug in DbfView example
Posted by mr_ped on Sat, 28 Jul 2007 13:06:36 GMT
View Forum Message <> Reply to Message

```
GUI_APP_MAIN
{
    SetDefaultCharset(CHARSET_WIN1250);
    SetLanguage(LNG_CZECH);
    Ctrl::NoLayoutZoom();
...
```

The SetLanguage does call (if the language is really changed):
SetDefaultCharset(GetLNGCharset(lang));
so the SetDefaultCharset after GUI_APP_MAIN is useless and misleading anyone who wants to learn UPP from examples.

Subject: Re: bug in DbfView example
Posted by mr_ped on Mon, 30 Jul 2007 03:23:23 GMT
View Forum Message <> Reply to Message

I found out some more problems...

1)

```
 while(dbf.Fetch()) {
  Vector<Value> v;
  for(int i = 0; i < min(4, dbf.GetFieldCount()); i++)
   v.Add(dbf[i]);
  table.Add(v);
 }
```

Does add all record to table, but it does skip over deleted rows, what does cause inconsistency with later

```
void DbfView::EnterRow()
{
 row.Clear();
 if(!dbf.Fetch(table.GetCursor()))
  return;
```

So you get wrong detailed view in such case, and whole index will be moved by -1 to -n (number of deleted rows).

Quick fix:

```
 if(!dbf.Open(~fs))
  Exclamation("Can't open input file");
 for(int i = 0; i < min(4, dbf.GetFieldCount()); i++)
  table.AddColumn(FormatField(dbf.GetField(i)));
 for (int ri = 0; ri < dbf.GetRowCount(); ++ri ) {
  Vector<Value> v;
  if(!dbf.Fetch(ri)) v.Add("DELETED");
  else for(int i = 0; i < min(4, dbf.GetFieldCount()); i++) v.Add(dbf[i]);
  table.Add(v);
 }
```

edit: ARGH, I used the local "i" two times... not very good programming practice... ... so I changed it to "ri".

This will add also deleted rows to the main table, so the cursor is always right than.

2)
I'm unable to show correct czech characters no matter what charset I try. (in my desperation I tried all 26 of them)
My code to convert looks like this (in void DbfView::EnterRow):

```
...
 for(int i = 0; i < dbf.GetFieldCount(); i++) {
  Value val = dbf[i];
  if ( val.GetType() == STRING_V ) {
   String valtxt = val.ToString();
   //WString valwtxt = ToUnicode(valtxt, CHARSET_WIN1250);
   WString valwtxt = ToUnicode(valtxt, dbf.GetCharset() );
   row.Add(FormatField(dbf.GetField(i)), valwtxt);
  } else {
   row.Add(FormatField(dbf.GetField(i)), val);
  }
```

U++ 2007.1, Kubuntu 6.10
Charset for IDE: UTF-8
Charset for application: tried UTF-8, win1250, ISO8859_2, CP852, MJK

No luck so far... I will try it later on windows, maybe there some charset will work??

@Mirek: is it possible the result on windows will be different?

(P.S. I'm trying to do my first commercial project in U++, something very small, ugly, and it should work with DBF files from FoxPro ... the final application will be for Windows, but so far I'm developing just some base and toying around, so I'm staying in my main system so far ... that's the reason why I'm so into this example, I really must be sure I understand the DBF plugin enough)

Subject: Re: bug in DbfView example
Posted by mr_ped on Mon, 30 Jul 2007 04:04:39 GMT
View Forum Message <> Reply to Message

Just another one:

```
fs.AllFilesType();
fs.Type("*.dbf", "dbf");
if(!fs.ExecuteOpen("DBF..")) return;
```

The filter *.dbf does not work in Linux.
Fix (to add asterisk and dot ahead of second dbf):

```
...
fs.Type("*.dbf", "*.dbf");
...
```

I'm not sure if this works for windows and which one is the "correct" way to use in U++. Was it just a problem with example and my fix is correct, or the whole filter thing is broken in Linux and U++ FileSel requires fixing?

Subject: Re: bug in DbfView example
Posted by mrjt on Mon, 30 Jul 2007 10:05:59 GMT
View Forum Message <> Reply to Message

"name", "filter" is the correct format. For example:
fs.Type("Text Files", "*.txt");
Although ".txt" will also work.

Subject: Re: bug in DbfView example

mrjt wrote on Mon, 30 July 2007 06:05"name", "filter" is the correct format. For example:
fs.Type("Text Files", "*.txt");
Although ".txt" will also work.

Sorry about messed example...

This should solve some of problems, please test:

```
#include <CtrlLib/CtrlLib.h>
#include <plugin/dbf/dbf.h>

using namespace Upp;

struct DbfView : public TopWindow {
 Splitter s;
 ArrayCtrl table;
 ArrayCtrl row;
 DbfStream dbf;

 void EnterRow();
 void Perform();

 typedef DbfView CLASSNAME;

 DbfView();
};

String FormatField(const DbfStream::Field& f)
{
 return f.name + Format(" (%c%d)", f.type, f.width);
}

void DbfView::EnterRow()
{
 row.Clear();
 if(!dbf.Fetch(table.GetKey()))
  return;
 for(int i = 0; i < dbf.GetFieldCount(); i++)
  row.Add(FormatField(dbf.GetField(i)), dbf[i]);
}

void DbfView::Perform()
{
```

```
 FileSel fs;
 LoadFromFile(fs);
 fs.AllFilesType();
 fs.Type("dbf", "*.dbf");
 if(!fs.ExecuteOpen("DBF..")) return;
 StoreToFile(fs);
 if(!dbf.Open(~fs))
  Exclamation("Can't open input file");
 table.AddKey();
 for(int i = 0; i < min(4, dbf.GetFieldCount()); i++)
  table.AddColumn(FormatField(dbf.GetField(i)));
 while(dbf.Fetch()) {
  Vector<Value> v;
  v.Add(dbf.GetPos());
  for(int i = 0; i < min(4, dbf.GetFieldCount()); i++)
   v.Add(dbf[i]);
  table.Add(v);
 }
 Run();
}

DbfView::DbfView()
{
 s.Set(table, row);
 s.SetPos(7000);
 Add(s.SizePos());
 Sizeable().Zoomable();
 table.WhenEnterRow = THISBACK(EnterRow);
 row.AddColumn("Column");
 row.AddColumn("Value", 2);
}

GUI_APP_MAIN
{
 SetDefaultCharset(CHARSET_WIN1252);
 Ctrl::NoLayoutZoom();

 DbfView().Perform();
}
```

I am moving this to the "library" area...

Mirek