Subject: Porting a Delphi Application
Posted by cbpporter on Sat, 15 Sep 2007 01:40:19 GMT
View Forum Message <> Reply to Message

Hi,

I have a Delphi application that I have been (very slowly) developing for quite some time now and for a while I have been thinking about porting the code to some C++ GUI library.

I don't like C++Builders proprietary extensions to C++, GTK is too C-ish (gobject) and can sometimes be very slow on Windows (the GUI), and Qt is one precompiled mess. I researched some other GUI toolkit's too, and I think U++ would be the best for me (even though it's too nonconformist).

So I will try to do the port, at first as a test, and if it works out I will continue add future developments to the U++ version. If it doesn't work, nobody will ever know .

I've gotten quite a good idea of the U++ API, but I can't seem to find and equivalent to Windows MDI Application style windows (TopWindows inserted into TopWindows, with tiling, cascading, ... support). Is MDI supported by U++ or do I have to convert my application to a SDI interface?

Thank you

Subject: Re: Porting a Delphi Application
Posted by cbpporter on Sat, 15 Sep 2007 11:44:02 GMT
View Forum Message <> Reply to Message

So I started porting and managed to create the File menu with relative ease. It takes some time to get used to the package management scheme in TheIDE, it's image editor and the fact that it is it's own little world, which is surprisingly self-contained.

The layout editor is great, but is not well suited for creating really complex layouts and it seems to favour composing a large number of simple layouts to achieve this goal. And it has no frame support (which I think is a design choice) and no menu editor. But it still helps a lot and I'm glad to have it.

But even after this brief time of exposure to U++ I have identified some issues. I will describe them:

1. My applications exe doesn't have the correct icon (it uses the generic one) when I compile my application in Optimal mode. I compiled some of the example applications to, and these also lack the correct icon. But if I compile it in Debug mode, the icons are fine.

2. If the layout designer or image editor is in the main tab of the editor in the moment the application launches (F5), these editors loose their advanced functionality and revert back to a plain text editor.

3. Fullscreen windows (under Windows) get a border and a title bar if the window looses focus.

4. If I create a frame parameterised with a Button (FrameLeft<Button> for example), the Button will have a strange blue-ish skin. But if I create a Frame with a layout which contains a Button, the Button has a normal skin. Is this intentional?

5. How do you activate the MacOS theme under Windows as seen in the screenshot (this is not an issue, just a question).

I think I forgot another one, and if I stumble upon it I will post again.

Thank You.

Subject: Re: Porting a Delphi Application
Posted by cbpporter on Sun, 16 Sep 2007 09:16:58 GMT
View Forum Message <> Reply to Message

I managed to solve the icon problem. Seems I need to have the icon exported into "icon.ico". I thought that those 6 bytes icons were strange...

But I found another bug. If a create a menu with a caption containing the accessor key multiple times (for example "Zoom &Out"), when I press Alt the first of that character will be highlighted (as if I had "Z&oom Out"). This is a minor detail but things like this shouldn't be ignored.

Thank You

Subject: Re: Porting a Delphi Application
Posted by mirek on Sun, 16 Sep 2007 10:24:56 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Fri, 14 September 2007 21:40
(even though it's too nonconformist).

Yes, it is.

Quote:
I've gotten quite a good idea of the U++ API, but I can't seem to find and equivalent to Windows MDI Application style windows (TopWindows inserted into TopWindows, with tiling, cascading, ... support). Is MDI supported by U++ or do I have to convert my application to a SDI interface?

No, there is no MDI support.

Mirek

Subject: Re: Porting a Delphi Application
Posted by mirek on Sun, 16 Sep 2007 10:30:42 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Sat, 15 September 2007 07:44
1. My applications exe doesn't have the correct icon (it uses the generic one) when I compile my application in Optimal mode. I compiled some of the example applications to, and these also lack the correct icon. But if I compile it in Debug mode, the icons are fine.

Debug mode is trange, however to procedure is to activate .ico export and add .rc file that uses it. See ide/.rc file as an example (the icon is BTW usually the only use for .rc in U++).

Quote:
2. If the layout designer or image editor is in the main tab of the editor in the moment the application launches (F5), these editors loose their advanced functionality and revert back to a plain text editor.

That is feature, not error The problem is that these files are, through some macro hackery, in fact valid C++ sources that can be stepped through by debugger.

Press Ctrl+T to get designer back.

Quote:
3. Fullscreen windows (under Windows) get a border and a title bar if the window looses focus.

Well, fullscreen windows are not quite typical, this might be a bug. Cared to post a testcase? (BTW, next time please post problems to more appropriate forums, this is really mostly about interesting apps being developed in U++, not problem solving area... your problem might be unnoticed here).

Quote:
4. If I create a frame parameterised with a Button (FrameLeft<Button> for example), the Button will have a strange blue-ish skin. But if I create a Frame with a layout which contains a Button, the Button has a normal skin. Is this intentional?

Yes, tries to immitate the appearance of ComboBox buttons.

Quote:

5. How do you activate the MacOS theme under Windows as seen in the screenshot (this is not an issue, just a question).

Google: "uxtheme".

Mirek

Subject: Re: Porting a Delphi Application
Posted by mirek on Sun, 16 Sep 2007 10:32:23 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Sun, 16 September 2007 05:16
But I found another bug. If a create a menu with a caption containing the accessor key multiple times (for example "Zoom &Out"), when I press Alt the first of that character will be highlighted (as if I had "Z&oom Out"). This is a minor detail but things like this shouldn't be ignored.

Actually, nobody cares about explicitly defining '&' keys - let U++ runtime generate them for you... and perhaps do it manually in cases when you do not like automatic assignment. You can also assign only some and let U++ do the rest.

Mirek

Subject: Re: Porting a Delphi Application
Posted by cbpporter on Sun, 16 Sep 2007 13:49:31 GMT
View Forum Message <> Reply to Message

Quote:
Actually, nobody cares about explicitly defining '&' keys - let U++ runtime generate them for you... and perhaps do it manually in cases when you do not like automatic assignment. You can also assign only some and let U++ do the rest.

And it seems it does quite a good job assigning the keys. But is still wish it would respect my explicitly assigned position, because a have "Zoom &In" and "Zoom &Out" and I would like to keep the "_"s aligned. But this little detail won't make me think less of U++, and I must admit that I'm impressed.

Quote:
No, there is no MDI support.

Well with owner draw widgets maybe I could emulate somehow MDI behaviour. Or at least I need a Window menu which allows you to switch between all your open documents.

Quote:
BTW, next time please post problems to more appropriate forums, this is really mostly about interesting apps being developed in U++, not problem solving area... your problem might be unnoticed here.

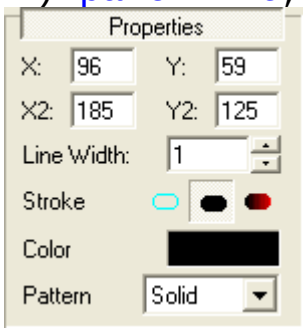
Sorry, I wanted to keep all info regarding this project in one place.

But to keep with the theme of this thread, I will need some custom components. Most of them are highly specialized and I don't see a general use for them, but one could be useful. It is a foldable panel. It looks like a LabelBox with a Button instead of the label, and by clicking it you can open and close it. In my original application this is done by a Panel, a Bevel, a Button and hand-coded folder behaviour. This is not that nice, and I'm going to try to develop a custom component which at least should extend to a property specified size (even better would be if I can make it to auto-detect the correct size and hide, show and align it's children. I attached a picture with this component.

Subject: Re: Porting a Delphi Application
Posted by cbpporter on Sun, 16 Sep 2007 13:51:11 GMT
View Forum Message <> Reply to Message

Forgot to attach...

File Attachments
1) panel.PNG, downloaded 1009 times



Subject: Re: Porting a Delphi Application
Posted by mirek on Sun, 16 Sep 2007 18:53:01 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Sun, 16 September 2007 09:49Quote:
Actually, nobody cares about explicitly defining '&' keys - let U++ runtime generate them for you... and perhaps do it manually in cases when you do not like automatic assignment. You can also assign only some and let U++ do the rest.

And it seems it does quite a good job assigning the keys. But is still wish it would respect my

explicitly assigned position, because a have "Zoom &In" and "Zoom &Out" and I would like to keep the "_"s aligned. But this little detail won't make me think less of U++, and I must admit that I'm impressed.

You can, just do not expect U++ to check your assignments (yet?

Quote:
Or at least I need a Window menu which allows you to switch between all your open documents.

See examples/UWord

Mirek

Subject: Re: Porting a Delphi Application
Posted by mr_ped on Sun, 16 Sep 2007 21:06:09 GMT
View Forum Message <> Reply to Message

Mirek: did you understand his problem with Zoom Out?
It looks to me like you didn't. He doesn't need U++ to check his assignments, he just needs that the underlined "o" will be the one in "Out" word, not the first one in "Zoom", because he didn't specify it there in Zoom, but in Out.

Subject: Re: Porting a Delphi Application
Posted by mirek on Sun, 16 Sep 2007 21:42:21 GMT
View Forum Message <> Reply to Message

I see thanks. (ops)

Subject: Re: Porting a Delphi Application
Posted by cbpporter on Mon, 17 Sep 2007 03:48:47 GMT
View Forum Message <> Reply to Message

Quote:See examples/UWord
Thanks, that served as a quick fix. I don't like the multiple windows but this way I don't have to wory about multiple documents in the same instance of the main window.

Quote:I see thanks. (ops)
If you don't have time, maybe I could find some time in weekend to look over the code, maybe find a way to fix this.

Regarding that drop down panel control, I managed to create one. It works well enough for the

time being, even though the code is rough, functionality is missing and it has problems with auto sizing. But it gets me going, so it’s a good thing. I can imagine a custom Bar component which would accept such controls as children and automatically align them vertically one under the other as their size change.

I attached the control's source. I don't have any experience in writing U++ controls, so maybe I get some feedback. And I couldn't find a centralized place where position or size change is handled (except WndSetPos, but that is OS dependent).

File Attachments

1) DropPane.zip, downloaded 648 times

Subject: Re: Porting a Delphi Application
Posted by mirek on Mon, 17 Sep 2007 07:50:22 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Sun, 16 September 2007 23:48
I attached the control's source. I don't have any experience in writing U++ controls, so maybe I get some feedback. And I couldn't find a centralized place where position or size change is handled (except WndSetPos, but that is OS dependent).

Ctrl::Layout

Mirek

Subject: Re: Porting a Delphi Application
Posted by mirek on Mon, 17 Sep 2007 08:03:12 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Sun, 16 September 2007 23:48
I attached the control's source.

Usual custom here, when you post some sources like this, is to use package and perhaps provide another one as demonstration - that way downloaders can just unpack to some package nest, start theide, choose the test and run it.

This is not a complaint, just a note - I guess it is time to write U++ netiquette summmary

Mirek

Subject: Re: Porting a Delphi Application
Posted by cbpporter on Mon, 17 Sep 2007 17:40:54 GMT
View Forum Message <> Reply to Message

luzr wrote on Mon, 17 September 2007 10:03
Usual custom here, when you post some sources like this, is to use package and perhaps provide another one as demonstration - that way downloaders can just unpack to some package nest, start theide, choose the test and run it.

This is not a complaint, just a note - I guess it is time to write U++ netiquette summmary

I will post another improved version (current one gives error when trying to use it as base for a template layout because it misses a SetRect function with a Rect parameter) once it is ready and I'll try to have a proper package format. Previous post was only so people get an idea of what I'm trying to do (and if can be done easier point me in the right direction).

But I have a functional (barely) DropPane and now I'm trying to insert inside an array of 2x4 ToolButtons (I'm not using button because I need them to be flat-looking), like in the attached picture.

Layout and positioning is very important, so I did not use a ToolBar component, instead I hand placed all the buttons. I also need them to check when the user click on them. I managed to do this, but unfortunately I had to use pointers (old habits die hard). I wonder if there is a better way to do this in a U++ style?

layout

```
LAYOUT(ToolsLayout, 60, 200)
 ITEM(ToolButton, btnMove, LeftPosZ(8, 22).TopPosZ(24, 22))
 ITEM(ToolButton, btnEdit, LeftPosZ(30, 22).TopPosZ(24, 22))
 ITEM(ToolButton, btnLine, LeftPosZ(8, 22).TopPosZ(46, 22))
 ITEM(ToolButton, btnBezier, LeftPosZ(30, 22).TopPosZ(46, 22))
 ITEM(ToolButton, btnRectangle, LeftPosZ(8, 22).TopPosZ(68, 22))
 ITEM(ToolButton, btnEllipse, LeftPosZ(30, 22).TopPosZ(68, 22))
END_LAYOUT
```

code

```
struct ToolsDropPane: public WithToolsLayout<DropPane> {

 typedef ToolsDropPane CLASSNAME;

 ToolsDropPane();

 void SetActive(ToolButton *btn);
};
```

```
ToolsDropPane::ToolsDropPane()
{
 CtrlLayout(*this);

 btnMove.Image(CBPImages::Move()).Text("Modify");
 btnMove.WhenAction = THISBACK1(SetActive, &btnMove);
 btnEdit.Image(CBPImages::Edit()).Text("Edit");
 btnEdit.WhenAction = THISBACK1(SetActive, &btnEdit);
 btnLine.Image(CBPImages::Line());
 btnLine.WhenAction = THISBACK1(SetActive, &btnLine);
 btnBezier.Image(CBPImages::Bezier());
 btnBezier.WhenAction = THISBACK1(SetActive, &btnBezier);
 btnRectangle.Image(CBPImages::Rectangle()).Text("Rectangle");
 btnRectangle.WhenAction = THISBACK1(SetActive, &btnRectangle);
 btnEllipse.Image(CBPImages::Ellipse()).Text("Ellipse");
 btnEllipse.WhenAction = THISBACK1(SetActive, &btnEllipse);

 //btnLine.Check(true);
}

void ToolsDropPane::SetActive(ToolButton *btn)
{
 btn->Check(true);
}
```

On the other hand, you will notice that the third and fourth button doesn't have a Text. This is intentional. When I don't specify a text to a button inserted like this into a layout, I get a weird tool tip for it: "(Alt+Shift+address)". Does this mean anything?

File Attachments

1) panel2.PNG, downloaded 2228 times



Subject: Re: Porting a Delphi Application
Posted by mirek on Mon, 17 Sep 2007 20:34:49 GMT
View Forum Message <> Reply to Message

Oh, no, if you want to follow U++ approach, there is nothing wrong with pointing to things.

What we try to avoid is to use pointers to manage resources.

Your use of pointer here is definitely OK and optimal solution.

BTW, before you will reach that point, drawing primitices in U++ are intentionally very dumb. There is no bezier

Mirek

Subject: Re: Porting a Delphi Application
Posted by mirek on Tue, 18 Sep 2007 18:57:13 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Sat, 15 September 2007 07:44
3. Fullscreen windows (under Windows) get a border and a title bar if the window looses focus.

This is found to be a bug; now fixed.

Quick fix:

```
void TopWindow::SyncCaption()
{
 LLOG("SyncCaption");
 if(fullscreen)     // ADD
  return;    // THIS :)
 HWND hwnd = GetHWND();
 if(hwnd) {
  style = ::GetWindowLong(hwnd, GWL_STYLE);
  exstyle = ::GetWindowLong(hwnd, GWL_EXSTYLE);
 }
```

Mirek

Subject: Re: Porting a Delphi Application
Posted by cbpporter on Tue, 18 Sep 2007 19:16:28 GMT
View Forum Message <> Reply to Message

luzr wrote on Tue, 18 September 2007 20:57
This is found to be a bug; now fixed.

Thanks. It worked. Glad I could help at least in this minor way. What do you think about the strange tooltips I mentioned in my last post?

By the way, how does bug-fixing work? I fixed it in my source-tree, but will others have to wait for next release or are the bugs fixed retroactivelly?

Subject: Re: Porting a Delphi Application
Posted by cbpporter on Wed, 19 Sep 2007 09:07:22 GMT
View Forum Message <> Reply to Message

On the other hand ToolButton doesn't refresh if I call Check(false) after it has been set to true if it is not inserted into a ToolBar. I get the distinct feeling that ToolButton was no intended to be used on it's own. And AFAIK there is no GetCheck or other method to verify if it is checked or not. ButtonOption has such capabilities.

Subject: Re: Porting a Delphi Application
Posted by mirek on Wed, 19 Sep 2007 10:18:28 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Tue, 18 September 2007 15:16luzr wrote on Tue, 18 September 2007 20:57
This is found to be a bug; now fixed.

Thanks. It worked. Glad I could help at least in this minor way. What do you think about the strange tooltips I mentioned in my last post?

Sorry missed it.

ToolButtons show its accelerator in the tip by default. I think this is related.

Quote:
By the way, how does bug-fixing work? I fixed it in my source-tree, but will others have to wait for next release or are the bugs fixed retroactivelly?

It will be in next dev release. At that point, you are supposed to simply replace the whole uppsrc with the new one...

Mirek

Subject: Re: Porting a Delphi Application
Posted by mirek on Wed, 19 Sep 2007 10:21:27 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Wed, 19 September 2007 05:07On the other hand ToolButton doesn't refresh if I call Check(false) after it has been set to true if it is not inserted into a ToolBar. I get the distinct feeling that ToolButton was no intended to be used on it's own.

Your distinct feeling is correct

Quote:
And AFAIK there is no GetCheck or other method to verify if it is checked or not. ButtonOption has such capabilities.

Yes. ToolButton is specifically dedicated to be used with ToolBar and there you never access the button instance.

Mirek

Subject: Re: Porting a Delphi Application
Posted by cbpporter on Wed, 19 Sep 2007 10:49:31 GMT
View Forum Message <> Reply to Message

Well I still need a button with a flat skin and random position. I'll try some experimets with multiple ToolBars and if they don't work maybe I'll fork ButtonOption to have an optional flat skin.

Subject: Re: Porting a Delphi Application
Posted by mirek on Wed, 19 Sep 2007 12:56:12 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Wed, 19 September 2007 06:49Well I still need a button with a flat skin and random position. I'll try some experimets with multiple ToolBars and if they don't work maybe I'll fork ButtonOption to have an optional flat skin.

Well,

1) Maybe you can just use chameleon to skin ButtonOption. ("SetStyle").

2) "Was intended" does not mean that ToolButton cannot be extended for other uses.

Mirek

Subject: Re: Porting a Delphi Application
Posted by cbpporter on Wed, 19 Sep 2007 18:32:39 GMT
View Forum Message <> Reply to Message

I have a little problem with XML loading. I couldn't find any documentation on the site so I looked at the provided examples and the Xml loader source code. Everything seems pretty straightforward and the examples work.

But I need something a little different. I need to keep the entire Xml tree in memory and populate several different trees (consider these trees as "View" and the Xml as the "model" with data computed on base of the Xml by traversing the Xml saved in the memory (multiple times). I also need to modify the Xml and save it.

So I used the ParseXml function to obtain an XmlNode, and used GetCount and assorted functions to traverse recursively it's children. For the first "view" I tried a simple mapping of the structure into a TreeCtrl. But as hard as I tried, I could get anything except a tree with an empty root and a child which has the filename as a child.

Which is the proper way to load a Xml like in the XmlView example, but by maintaining a copy of the Xml in memory?

Subject: Re: Porting a Delphi Application
Posted by zsolt on Wed, 19 Sep 2007 19:42:02 GMT
View Forum Message <> Reply to Message

Check AddressBookXML2 example app.

Subject: Re: Porting a Delphi Application
Posted by cbpporter on Wed, 19 Sep 2007 19:49:27 GMT
View Forum Message <> Reply to Message

zsolt wrote on Wed, 19 September 2007 21:42Check AddressBookXML2 example app.
I did. I guess I have to do more in depth debugging of why it doesn't work.

On the other hand, I changed a method in ToolButton and the control now refreshes itself when Check status is updated. A further optimisation could be to check if it's parent is a ToolBar, and don't refresh in that case.

```
Bar::Item& ToolButton::Check(bool check)
{
 if (checked != check)
 {
  checked = check;
  Refresh();
 }
```

```
 return *this;
}
```

Subject: Re: Porting a Delphi Application
Posted by cbpporter on Wed, 19 Sep 2007 20:26:39 GMT
View Forum Message <> Reply to Message

Nevermind my problem with Xml. It was my fault. I used:

XmlNode node = ParseXML(filename);

instead of:

String d = LoadFile(filename);
XmlNode node = ParseXML(d);

The ParseXml declaration has a char* parameter named s, and I thought if was the filename.

Subject: Re: Porting a Delphi Application
Posted by cbpporter on Thu, 20 Sep 2007 16:24:04 GMT
View Forum Message <> Reply to Message

Does anybody have any experience installing and using Cairo under windows to render into a control? I would use built-in functions, but I need something powerful and Mirek said that the graphic libs are not that powerful anyway.

Subject: Re: Porting a Delphi Application
Posted by mr_ped on Thu, 20 Sep 2007 17:37:18 GMT
View Forum Message <> Reply to Message

There been some experiments with AGG, search the forum.
(And I'm not sure if AGG will suffice to your needs)

I don't remember reading in forums about anyone trying to use Cairo inside Upp.

Subject: Re: Porting a Delphi Application
Posted by cbpporter on Fri, 21 Sep 2007 18:12:15 GMT
View Forum Message <> Reply to Message

Well compiling Cairo under U++ sure is hell, I couldn’t get any significant progress so I decided to use the precompiled Windows dll and try to integrate that into U++ control drawing. But before that I have some problems with accessing TreeControl items.
I am trying to insert an TreeNodes with the key as the tag name of a xml tag and the value as the actual XmlNode. On WhenSel I need to obtain that XmlNode and create an object based on it's content.

But I get a dynamic_cast exception when I try to do this.

```
void MainWindow::Load(int parent, const XmlNode& p)
{
 int t = xml.Add(parent, CBPImages::Move(), p.GetTag(), p);
 for (int i = 0; i < p.GetCount(); i++)
  Load(t, p[i]);
}

void MainWindow::Load(const char *filename)
{
 try
 {
  XmlNode node = ParseXML(LoadFile(filename));
  int t = xml.Add(0, CBPImages::Move(), "Image", node);
  for (int i = 0; i < node.GetCount(); i++)
   Load(t, node[i]);
 }
 catch (XmlError e)
 {
  PromptOK("XMLError");
 }
}

void MainWindow::SelectItem()
{
 const XmlNode& node = ValueTo<XmlNode>(xml.GetValue());
 lbl.SetLabel(node.GetTag());
}
```

Subject: Re: Porting a Delphi Application
Posted by mirek on Fri, 21 Sep 2007 22:22:35 GMT
View Forum Message <> Reply to Message

You cannot add XmlNode directly this way - it is not a Value compatible type.

What indeed happens is that XmlNode -> Value conversion is in fact XmlNode -> operator bool()

-> Value. So in fact, you are storing into tree a bool value that indicates whether XmlNode contains subnodes...

Then, later, bool cannot be converted to XmlNode -> dynamic cast error.

Mirek

Subject: Re: Porting a Delphi Application
Posted by cbpporter on Fri, 21 Sep 2007 22:25:46 GMT
View Forum Message <> Reply to Message

Actually I did some testing and I'm reasonably sure that it is converted to a String (Value::ToString). Did IsType check and ValueTo<String> returns the key in the TreeCtrl.

So how could I store some extra info into nodes?

Subject: Re: Porting a Delphi Application
Posted by mirek on Fri, 21 Sep 2007 23:39:37 GMT
View Forum Message <> Reply to Message

Soerry, my mistake, you are right. You are getting the key value back, which was added as GetTag. The XmlNode bool was added into value (the think which is about to be displayed).

Well, you can add almost anything as Value key to the TreeCtrl. See Value related reference examples.

Mirek
