Subject: Major Draw refactoring
Posted by mirek on Tue, 07 Jul 2009 08:04:52 GMT
View Forum Message <> Reply to Message

I am happy to announce that Draw development is getting to its final phase.

Current major step moved all GUI related issues to CtrlCore, leaving Draw "abstract", decoupling X11 (or GDI). Meanwhile, font/text support is still available (in POSIX using raw FontConfig).

Fonts are also much improved:

- new implementation allowed deprecating of FontInfo - you can now get all metrics directly from Font (FontInfo still exists to maintain compatibility).

- new system now supports font replacements - means you can now use CJK glyphs with any font and they will be taken from another font.

- Painter is now able to use both glyph composition and replacements.

Of course, as with any such major redesign, you can expect some issues and bugs...

The last phase should be relatively simple - moving Painter to Draw and fixing POSIX library requirements...

Mirek

Subject: Re: Major Draw refactoring
Posted by gxl117 on Tue, 07 Jul 2009 09:33:24 GMT
View Forum Message <> Reply to Message

luzr wrote on Tue, 07 July 2009 10:04
- new system now supports font replacements - means you can now use CJK glyphs with any font and they will be taken from another font.

Very good news for me!

Subject: Re: Major Draw refactoring
Posted by cbpporter on Thu, 09 Jul 2009 08:03:20 GMT
View Forum Message <> Reply to Message

I'll try out the new Draw with my font replacing/composition system under Linux.

Subject: Re: Major Draw refactoring
Posted by Didier on Thu, 09 Jul 2009 18:23:25 GMT
View Forum Message <> Reply to Message

The new version of dran does not compile under linux due to MT issues. There is a mixup indefines for msc and gcc.

I haven't looked at it right now

Subject: Re: Major Draw refactoring
Posted by mirek on Thu, 09 Jul 2009 19:49:57 GMT
View Forum Message <> Reply to Message

Didier wrote on Thu, 09 July 2009 14:23The new version of dran does not compile under linux due to MT issues. There is a mixup indefines for msc and gcc.

I haven't looked at it right now

Thank you, the problem was that thread__ variables were not POD.

Hopefully fixed.

Mirek

Subject: Re: Major Draw refactoring
Posted by cocob on Fri, 10 Jul 2009 07:09:35 GMT
View Forum Message <> Reply to Message

Does this correction have a link with this ?

http://www.ultimatepp.org/forum/index.php?t=msg&th=4532& amp; amp;start=0&

Subject: Re: Major Draw refactoring
Posted by Didier on Fri, 10 Jul 2009 18:28:45 GMT
View Forum Message <> Reply to Message

Thanks,

Perfect hit, compilation of new font is now OK on linux with MT

Subject: Re: Major Draw refactoring
Posted by mirek on Sun, 12 Jul 2009 07:22:57 GMT
View Forum Message <> Reply to Message

cocob wrote on Fri, 10 July 2009 03:09Does this correction have a link with this ?

http://www.ultimatepp.org/forum/index.php?t=msg&th=4532& amp; amp; amp;start=0&

Unfortunately, not. Waiting for the testcase.

Mirek

Subject: Re: Major Draw refactoring
Posted by nixnixnix on Tue, 14 Jul 2009 17:28:11 GMT
View Forum Message <> Reply to Message

I just updated my version of UPP from the SVN and my code appears to be broken.

It wont compile due to the following errors:

Quote:
D:\MyApps\OpenWind\WindMap.cpp(2745) : error C2664:
'Upp::ImageBuffer::ImageBuffer(Upp::Size)' : cannot convert parameter 1 from 'Upp::ImageDraw'
to 'Upp::Size'
No user-defined-conversion operator available that can perform this conversion, or the
operator cannot be called
D:\MyApps\OpenWind\main.cpp(1080) : error C2039: 'GetPagePixels' : is not a member of
'Upp::Draw'
C:\upp\uppsrc\Draw/Draw.h(412) : see declaration of 'Upp::Draw'
D:\MyApps\OpenWind\main.cpp(1105) : error C2039: 'GetPagePixels' : is not a member of
'Upp::PdfDraw'
C:\upp\uppsrc\PdfDraw/PdfDraw.h(228) : see declaration of 'Upp::PdfDraw'
D:\MyApps\OpenWind\main.cpp(1136) : error C2664: 'Upp::ImageBuffer::ImageBuffer(Upp::Size)'
: cannot convert parameter 1 from 'Upp::ImageDraw' to 'Upp::Size'
No user-defined-conversion operator available that can perform this conversion, or the
operator cannot be called
D:\MyApps\OpenWind\main.cpp(1163) : error C2664: 'Upp::ImageBuffer::ImageBuffer(Upp::Size)'
: cannot convert parameter 1 from 'Upp::ImageDraw' to 'Upp::Size'
No user-defined-conversion operator available that can perform this conversion, or the
operator cannot be called
D:\MyApps\OpenWind\main.cpp(1191) : error C2664: 'Upp::ImageBuffer::ImageBuffer(Upp::Size)'
: cannot convert parameter 1 from 'Upp::ImageDraw' to 'Upp::Size'
No user-defined-conversion operator available that can perform this conversion, or the
operator cannot be called
D:\MyApps\OpenWind\main.cpp(1219) : error C2664: 'Upp::ImageBuffer::ImageBuffer(Upp::Size)'
: cannot convert parameter 1 from 'Upp::ImageDraw' to 'Upp::Size'

No user-defined-conversion operator available that can perform this conversion, or the operator cannot be called
D:\MyApps\OpenWind\WRGLayer.cpp(477) : error C2664:
'Upp::ImageBuffer::ImageBuffer(Upp::Size)' : cannot convert parameter 1 from 'Upp::ImageDraw' to 'Upp::Size'
No user-defined-conversion operator available that can perform this conversion, or the operator cannot be called

Is there any documentation on these changes please and are there equivalent replacement functions for those that have been removed? I thought that UPP was mature enough that this wasn't going to happen.

EDIT: ok well Vista was able to restore my previous version of UPP so I can continue working but of course I am now stuck in a limbo between 2008.1 and the latest version. Will these problems be fixed? I take it that there is another function instead of GetPagePixels (although I am not sure what other name could so perfectly sum up the function) but what about ImageDraw to ImageBuffer? How will that be accomplished in the new version?

Nick

Subject: Re: Major Draw refactoring
Posted by mirek on Wed, 15 Jul 2009 00:52:05 GMT
View Forum Message <> Reply to Message

nixnixnix wrote on Tue, 14 July 2009 13:28I just updated my version of UPP from the SVN and my code appears to be broken.

It wont compile due to the following errors:

Quote:
D:\MyApps\OpenWind\WindMap.cpp(2745) : error C2664:
'Upp::ImageBuffer::ImageBuffer(Upp::Size)' : cannot convert parameter 1 from 'Upp::ImageDraw' to 'Upp::Size'
No user-defined-conversion operator available that can perform this conversion, or the operator cannot be called
D:\MyApps\OpenWind\main.cpp(1080) : error C2039: 'GetPagePixels' : is not a member of 'Upp::Draw'
C:\upp\uppsrc\Draw/Draw.h(412) : see declaration of 'Upp::Draw'
D:\MyApps\OpenWind\main.cpp(1105) : error C2039: 'GetPagePixels' : is not a member of 'Upp::PdfDraw'
C:\upp\uppsrc\PdfDraw/PdfDraw.h(228) : see declaration of 'Upp::PdfDraw'
D:\MyApps\OpenWind\main.cpp(1136) : error C2664: 'Upp::ImageBuffer::ImageBuffer(Upp::Size)' : cannot convert parameter 1 from 'Upp::ImageDraw' to 'Upp::Size'
No user-defined-conversion operator available that can perform this conversion, or the operator cannot be called

D:\MyApps\OpenWind\main.cpp(1163) : error C2664: 'Upp::ImageBuffer::ImageBuffer(Upp::Size)' : cannot convert parameter 1 from 'Upp::ImageDraw' to 'Upp::Size'
No user-defined-conversion operator available that can perform this conversion, or the operator cannot be called
D:\MyApps\OpenWind\main.cpp(1191) : error C2664: 'Upp::ImageBuffer::ImageBuffer(Upp::Size)' : cannot convert parameter 1 from 'Upp::ImageDraw' to 'Upp::Size'
No user-defined-conversion operator available that can perform this conversion, or the operator cannot be called
D:\MyApps\OpenWind\main.cpp(1219) : error C2664: 'Upp::ImageBuffer::ImageBuffer(Upp::Size)' : cannot convert parameter 1 from 'Upp::ImageDraw' to 'Upp::Size'
No user-defined-conversion operator available that can perform this conversion, or the operator cannot be called
D:\MyApps\OpenWind\WRGLayer.cpp(477) : error C2664: 'Upp::ImageBuffer::ImageBuffer(Upp::Size)' : cannot convert parameter 1 from 'Upp::ImageDraw' to 'Upp::Size'
No user-defined-conversion operator available that can perform this conversion, or the operator cannot be called

Is there any documentation on these changes please and are there equivalent replacement functions for those that have been removed? I thought that UPP was mature enough that this wasn't going to happen.

EDIT: ok well Vista was able to restore my previous version of UPP so I can continue working but of course I am now stuck in a limbo between 2008.1 and the latest version. Will these problems be fixed? I take it that there is another function instead of GetPagePixels (although I am not sure what other name could so perfectly sum up the function) but what about ImageDraw to ImageBuffer? How will that be accomplished in the new version?

Nick

I have added "deprecated" backward compatibility version of GetPagePixels.

W.r.t. ImageDraw / ImageBuffer, there was no change. Can you post small code snippet please?

Mirek

Subject: Re: Major Draw refactoring
Posted by nixnixnix on Wed, 15 Jul 2009 04:56:45 GMT
View Forum Message <> Reply to Message

Hi Mirek,

Here is the code fragment that gives the error at main.cpp(1163).

void OpenWind::PrintTIF()

```
{
 FileSel fs;
 fs.Type("TIF files", "*.tif").AllFilesType().DefaultExt("tif");
 fs.ActiveDir(GetCurrentDirectory());

 if(!fs.ExecuteSaveAs("Output TIF file"))
  return;

 CUtils::SetCurrentDirectory_(fs.GetActiveDir());

 // drawgrid
 if(m_optionGrid.GetData())
 {
  Size sz = m_rcClient.Size();
  ImageDraw id(sz);
  id.DrawImage(0,0,sz.cx,sz.cy,m_view);
  DrawGrid(id);
  ImageBuffer ib(id); // I beleive this is the problem right here
  m_view = Image(ib);
 }

 TIFEncoder tif;

 tif.SaveFile(~fs,m_view);
 SaveWorldFile(~fs);
}
```

thanks for taking a look at this.

Nick

Subject: Re: Major Draw refactoring
Posted by mirek on Wed, 15 Jul 2009 08:20:50 GMT
View Forum Message <> Reply to Message

Thanks, I have added (returned) backward compatibility method to Image to solve the issue. I hope it will be OK.

Mirek

Subject: Re: Major Draw refactoring
Posted by nixnixnix on Wed, 15 Jul 2009 16:25:46 GMT

Hi Mirek,

I just updated again but this time I get different errors

Quote:
FontWin32.cpp
C:\upp\uppsrc\Draw\FontWin32.cpp(171) : error C2039: 'LOCAL' : is not a member of 'Upp::Font'
c:\upp\uppsrc\draw\Draw.h(46) : see declaration of 'Upp::Font'
C:\upp\uppsrc\Draw\FontWin32.cpp(171) : error C2065: 'LOCAL' : undeclared identifier

```
static int CALLBACK Win32_AddFace(const LOGFONT *logfont, const TEXTMETRIC *, dword
type, LPARAM param)
{
#ifdef PLATFORM_WINCE
 const wchar *facename = (const wchar *)param;
 if(facename && _wcsicmp(logfont->lfFaceName, facename))
  return 1;
#else
 const char *facename = (const char *)param;
 if(facename && stricmp(logfont->lfFaceName, facename))
  return 1;
#endif

 dword typ = 0;
 if((logfont->lfPitchAndFamily & 3) == FIXED_PITCH)
  typ |= Font::FIXEDPITCH;
 if(type & TRUETYPE_FONTTYPE)
  typ |= Font::SCALEABLE;
 if(!(logfont->lfCharSet == SYMBOL_CHARSET) && logfont->lfCharSet != 0)
  typ |= Font::LOCAL;
#ifdef PLATFORM_WINCE
 FontFaceInfo& f = sFontFace().Add(WString(logfont->lfFaceName).ToString());
 f.name = FromSystemCharset(logfont->lfFaceName);
#else
 FaceInfo& f = sList->Add();
 f.name = FromSystemCharset(logfont->lfFaceName);
 f.info = typ;
#endif
 return facename ? 0 : 1;
}
```

I get this error with MSC9 and a similar error with MINGW.

Nick

Subject: Re: Major Draw refactoring
Posted by mirek on Wed, 15 Jul 2009 16:34:39 GMT
View Forum Message <> Reply to Message

This not what is in svn:

http://code.google.com/p/upp-mirror/source/browse/trunk/upps rc/Draw/FontWin32.cpp

Mirek

Subject: Re: Major Draw refactoring
Posted by nixnixnix on Wed, 15 Jul 2009 19:08:24 GMT
View Forum Message <> Reply to Message

Ah ok yes. There are several files I needed to manually edit to get things to compile and so these files didn't get updated when I updated from the SVN.

These seem to be mostly fixed now - thanks. Is there any way we can sort out that ScalarProduct and VectorProduct in Bezier2Length please? I have lots of users who compile on either Linux or with MINGW and its the same error on both.

Thanks,

Nick

Subject: Re: Major Draw refactoring
Posted by mirek on Thu, 16 Jul 2009 00:59:47 GMT
View Forum Message <> Reply to Message

Hopefully patched.

What are you using Geom for?

Mirek

Subject: Re: Major Draw refactoring
Posted by nixnixnix on Thu, 16 Jul 2009 02:14:44 GMT

View Forum Message <> Reply to Message

Its referenced by the Tomas's geotiff package.

Everything builds now on Linux (64bit) and with MINGW with no alterations.

Thanks Mirek,

Nick

Subject: Re: Major Draw refactoring
Posted by nixnixnix on Thu, 16 Jul 2009 16:32:33 GMT
View Forum Message <> Reply to Message

When I test my software with the latest UPP, my fonts are messed up (when I reload a workbbok from disk) as well as the drop-downs not working (not readable and not dropping down) and the GridCtrl appears to have its y-dimension squashed.

Also, the message boxes (e.g. PromptOK) don't display properly. They are squashed.

Is this intended behaviour and if so, what do I need to do to fix my software?

Nick

p.s. these appear to be Vista issues. It doesn't happen on Ubuntu64 (I can't test on XP 32 just now). I use Vista 64 as my main development platform just now and believe me, on that UPP is not looking good. It was fine before the refactoring though.

I have had to revert again as I need to be able to show a working version of my software. Could you please post in this thread when its fixed. Mirek, if you don't have Vista 64 and want to see the problem I can show you using showmypc.

Subject: Re: Major Draw refactoring
Posted by ncompoop on Fri, 17 Jul 2009 09:01:52 GMT
View Forum Message <> Reply to Message

Just an addition to nixnixnix Vista woes. I appears that UPP can't find any fonts, only STDFONT is available, which is not a pretty sight indeed.

The problem started somewhere after svn.1415.

Subject: Re: Major Draw refactoring
Posted by mirek on Fri, 17 Jul 2009 11:16:04 GMT
View Forum Message <> Reply to Message

This should fix the issue with loading fonts:

```
void Font::Serialize(Stream& s) {
 int version = 1;
 s / version;
 if(version >= 1) {
    enum {
  OLD_STDFONT, OLD_SCREEN_SERIF, OLD_SCREEN_SANS, OLD_SCREEN_FIXED,
  OLD_ROMAN,
  OLD_ARIAL,
  OLD_COURIER,
 };
 int f = GetFace();
 if(f > COURIER)
  f = -1;
 s / f;
 String name;
 if(f == OLD_ROMAN)
  f = ROMAN;
 if(f == OLD_ARIAL)
  f = ARIAL;
 if(f == OLD_COURIER)
  f = COURIER;
 if(f < 0) {
  name = GetFaceName();
  s % name;
 }
 if(s.IsLoading())
  if(f >= 0)
   Face(f);
  else
   FaceName(name);
 }
 else {
  String name = GetFaceName();
  s % name;
  if(s.IsLoading()) {
   FaceName(name);
   if(IsNull())
    Face(COURIER);
  }
 }
 s % v.flags % v.height % v.width;
}
```

Now going to look into Vista issue...

Mirek

Subject: Re: Major Draw refactoring
Posted by koldo on Fri, 17 Jul 2009 11:34:25 GMT
View Forum Message <> Reply to Message

Hello Mirek

Now class Draw has not IsDrawing() method.
IsDrawing() is used in Bazaar package Scatter.

How has it to be done with new Draw ?

Best regards
Koldo

Subject: Re: Major Draw refactoring
Posted by mirek on Fri, 17 Jul 2009 12:18:48 GMT
View Forum Message <> Reply to Message

koldo wrote on Fri, 17 July 2009 07:34Hello Mirek

Now class Draw has not IsDrawing() method.
IsDrawing() is used in Bazaar package Scatter.

How has it to be done with new Draw ?

Best regards
Koldo

Well, the first question is whether you really need to know if Draw is DrawingDraw...

If yes, you can use dynamic_cast<DrawingDraw *>.

If that is not enough, we can consider moving IsDrawing. But I was trying to simplify the interface as much as possible...

Mirek

Subject: Re: Major Draw refactoring
Posted by mirek on Fri, 17 Jul 2009 12:19:57 GMT
View Forum Message <> Reply to Message

ncompoop wrote on Fri, 17 July 2009 05:01Just an addition to nixnixnix Vista woes. I appears that UPP can't find any fonts, only STDFONT is available, which is not a pretty sight indeed.

The problem started somewhere after svn.1415.

Yes, correct. It should now be fixed.

Hopefully, these turbulent times will be over soon

Mirek

Subject: Re: Major Draw refactoring
Posted by Novo on Fri, 17 Jul 2009 15:24:51 GMT
View Forum Message <> Reply to Message

luzr wrote on Fri, 17 July 2009 08:18

If yes, you can use dynamic_cast<DrawingDraw *>.

Is it really necessary to use RTTI with UPP? It makes code bigger and used only at very limited number of places in UPP. Avoiding RTTI (and exceptions) would attract small platform users.

Sorry for the offtopic.

Subject: Re: Major Draw refactoring
Posted by mirek on Fri, 17 Jul 2009 15:43:36 GMT
View Forum Message <> Reply to Message

Novo wrote on Fri, 17 July 2009 11:24luzr wrote on Fri, 17 July 2009 08:18

If yes, you can use dynamic_cast<DrawingDraw *>.

Is it really necessary to use RTTI with UPP? It makes code bigger and used only at very limited number of places in UPP. Avoiding RTTI (and exceptions) would attract small platform users.

Sorry for the offtopic.

RTTI is valid part of C++ standard and while minor feature, it has a nice ability to automatically provide solution to problems similiar to this one.

I do not believe that it makes code *significantly* bigger. There is not that much to store in memory in order to support RTTI (AFAIK).

Mirek

Subject: Re: Major Draw refactoring
Posted by Novo on Fri, 17 Jul 2009 21:54:01 GMT
View Forum Message <> Reply to Message

luzr wrote on Fri, 17 July 2009 11:43Novo wrote on Fri, 17 July 2009 11:24luzr wrote on Fri, 17 July 2009 08:18

If yes, you can use dynamic_cast<DrawingDraw *>.

Is it really necessary to use RTTI with UPP? It makes code bigger and used only at very limited number of places in UPP. Avoiding RTTI (and exceptions) would attract small platform users.

Sorry for the offtopic.

RTTI is valid part of C++ standard and while minor feature, it has a nice ability to automatically provide solution to problems similiar to this one.

I do not believe that it makes code *significantly* bigger. There is not that much to store in memory in order to support RTTI (AFAIK).

Mirek

of overhead because of RTTI (you can see at least mangled names in data sections). This

give much bigger overhead.

During research of this topic I found a very interesting article:
http://connect.microsoft.com/VisualStudio/feedback/ViewFeedb ack.aspx?FeedbackID=471325

executable but increases run-time memory consumption. This explains why all windows applications require so much memory .

Probably, MINGW is not that bad after all.

TheIDE according to dumpbin:

```
  Summary

       142000 .data
       172000 .rdata
         1000 .rsrc
       257000 .text

SECTION HEADER #3
   .data name
  1412FC virtual size
  3CA000 virtual address (007CA000 to 0090B2FB)
   34000 size of raw data
  3CA000 file pointer to raw data (003CA000 to 003FDFFF)
       0 file pointer to relocation table
       0 file pointer to line numbers
       0 number of relocations
       0 number of line numbers
C0000040 flags
         Initialized Data
         Read Write

  Summary

       142000 .data
```

The ".data" segment is 13% of the ".text" segment. Not all ".data" is RTTI but still ...
And for some reason it requires 55% of ".text"'s size at run-time.

Any way, small device guys always compile with RTTI disabled.

Subject: Re: Major Draw refactoring
Posted by mirek on Sat, 18 Jul 2009 00:12:21 GMT
View Forum Message <> Reply to Message

Novo wrote on Fri, 17 July 2009 17:54

The ".data" segment is 13% of the ".text" segment. Not all ".data" is RTTI but still ...
And for some reason it requires 55% of ".text"'s size at run-time.

Any way, small device guys always compile with RTTI disabled.

IMO, most of ".data" is .iml...

Mirek

Subject: Re: Major Draw refactoring
Posted by Novo on Sat, 18 Jul 2009 00:55:42 GMT
View Forum Message <> Reply to Message

luzr wrote on Fri, 17 July 2009 20:12Novo wrote on Fri, 17 July 2009 17:54

The ".data" segment is 13% of the ".text" segment. Not all ".data" is RTTI but still ...
And for some reason it requires 55% of ".text"'s size at run-time.

Any way, small device guys always compile with RTTI disabled.

IMO, most of ".data" is .iml...

Mirek

IMO, that should go in ".rdata" because it is read-only.

Subject: Re: Major Draw refactoring
Posted by andrei_natanael on Sun, 19 Jul 2009 10:58:24 GMT
View Forum Message <> Reply to Message

Novo wrote on Sat, 18 July 2009 03:55luzr wrote on Fri, 17 July 2009 20:12Novo wrote on Fri, 17 July 2009 17:54

The ".data" segment is 13% of the ".text" segment. Not all ".data" is RTTI but still ...
And for some reason it requires 55% of ".text"'s size at run-time.

Any way, small device guys always compile with RTTI disabled.

IMO, most of ".data" is .iml...

Mirek

IMO, that should go in ".rdata" because it is read-only.
I think it's not read-only because you can modify the content of an .iml file, replacing an image from Iml class with your own image.

Subject: Re: Major Draw refactoring
Posted by mirek on Sun, 19 Jul 2009 13:17:35 GMT
View Forum Message <> Reply to Message

andrei_natanael wrote on Sun, 19 July 2009 06:58Novo wrote on Sat, 18 July 2009 03:55luzr wrote on Fri, 17 July 2009 20:12Novo wrote on Fri, 17 July 2009 17:54

The ".data" segment is 13% of the ".text" segment. Not all ".data" is RTTI but still ...
And for some reason it requires 55% of ".text"'s size at run-time.

Any way, small device guys always compile with RTTI disabled.

IMO, most of ".data" is .iml...

Mirek

IMO, that should go in ".rdata" because it is read-only.
I think it's not read-only because you can modify the content of an .iml file, replacing an image from Iml class with your own image.

Actually, that might be unrelated (original data are still in memory, just pointers get updated).

Mirek

Subject: Re: Major Draw refactoring
Posted by nixnixnix on Mon, 10 Aug 2009 18:46:23 GMT
View Forum Message <> Reply to Message

Hey Mirek,

Just a quick note to say thanks for fixing the Vista issues. Everything looks good from where I am.

Cheers,

Nick

Subject: Re: Major Draw refactoring
Posted by nixnixnix on Tue, 11 Aug 2009 15:49:38 GMT
View Forum Message <> Reply to Message

Hey Mirek,

You moved some of the ImageDraw code to CtrlCore! You tried to trick me but I caught you!

N

Subject: Re: Major Draw refactoring
Posted by mirek on Tue, 11 Aug 2009 16:25:47 GMT
View Forum Message <> Reply to Message

nixnixnix wrote on Tue, 11 August 2009 11:49Hey Mirek,

You moved some of the ImageDraw code to CtrlCore! You tried to trick me but I caught you!

N

Well, that was necessary to achieve the basic goal: Make Draw idependent from platform GUI.

There is now, BTW, new class ImageAnyDraw, which would use either ImagePainter (with priority) or ImageDraw (which I have typedefed to SystemImageDraw too, just to make it more clear). With ImagePainter variants, no GDI or X11 calls are necessary to create the Image.

Mirek

Subject: Re: Major Draw refactoring
Posted by nixnixnix on Tue, 11 Aug 2009 18:20:55 GMT
View Forum Message <> Reply to Message

I see. Good idea. You just need to add the Alpha() method then and I can convert to ImageAnyDraw and not include the CtrlCore package in my image processing console apps.

Cheers,

Nick