Subject: The plan....
Posted by mirek on Sun, 24 Aug 2008 16:56:22 GMT
View Forum Message <> Reply to Message

I think it is a good time to summarize all things to be done to make U++ really great platform.

So let us start. Following list will be incomlete and perhaps messy; it is the purpose of this post and thread to create some cozy todo list (and maybe roadmap?).

First, there are infrastructure issues: Make sure svn rights work as intended, do nightly website update, nightly builds, later automated testing etc... Oh, and I forgot about backup solution too

The documentation and class browser. I have done some first steps, I think next thing to do is add "code annotation" capability as the main tool, then support it with better C++ parser and finally do nice code browser.

Of course, a little bit harder part will be to actually write documentation

Next "true unicode" support. That means going wchar -> int, interpreting such WStrings in Draw correctly, also font replacement in X11. Later RTL support.

Core enhancements: Add merssene twister based random to Core, 'cure' Uuid with it, find a way how to support Posix events in Win32. I believe we also need a refactored TcpSocket in Core too. LocalProcess, SysExec.

Software rendering and Draw: Finally do it... most likely based on Cairo (or cairo codebase BTW, ideally, we should be able to e.g. draw PNG images WITHOUT having X11 libraries on the system. That would be perfect for www/server applications. (Speaking about it, in future, I would like to try some nice web toolkit).

OK, feel free to add to the list.

Mirek

Subject: Re: The plan....
Posted by unodgs on Sun, 24 Aug 2008 17:57:40 GMT
View Forum Message <> Reply to Message

From my side:

- GridCtrl2 - to deliver best grid in the market for free . New version will be (is) implemented from scratch. Main goals: grouping, virtual mode, different views (classical and cards), filtering system (I have it implemented in my current project) better integration with SQL package (less coding)

- together with James: docking system in TheIde. We have nv branch in svn, but none of us seem to have time right now to integrate it. But ofcourse it will be done.

- much better quicktabs (now we call it "tabbar"). Now they support drawing in any direction and drag and drop. In "ide" mode they will deliver some great solutions (I finally found how it should work).

- maybe some kind of ORM.

Subject: Re: The plan....
Posted by chickenk on Mon, 25 Aug 2008 09:18:13 GMT
View Forum Message <> Reply to Message

luzr wrote on Sun, 24 August 2008 18:56Software rendering and Draw: Finally do it... most likely based on Cairo (or cairo codebase BTW, ideally, we should be able to e.g. draw PNG images WITHOUT having X11 libraries on the system. That would be perfect for www/server applications. (Speaking about it, in future, I would like to try some nice web toolkit).
May I ask if the Cairo choice is immutable or not ? if there is place for some discussion here, I would simply like to bring to your attention some other great rendering libraries:

 AntiGrain (http://www.antigrain.com) -> some work already done, check for topics in this forum)
 Imlib2 (http://www.enlightenment.org) -> low-level fast rendering, not sure it has some advantages compared to actual cairo
 Evas (http://www.enlightenment.org) -> scene-graph based canvas library, extremely optimized and amazingly fast. Work is in good progress to integrate Win32 and Quartz backends. Already present backends: X11, Xrender, OpenGL, SDL, framebuffer (DirectFB used to be present as well, I am not sure it is maintained).I'm a big fan of EFL (Enlightenment Foundation Libraries), but I would tend to personally choose AntiGrain, which also is very optimized, and most of all extremely accurate (more specifically font-rendering). Unless there is a specific reason to use Cairo.

Just to make sure Ultimate++ will (as usual) do the best to provide high-quality and optimized results.

Yours,
Lionel

Subject: Re: The plan....
Posted by mirek on Mon, 25 Aug 2008 09:39:50 GMT
View Forum Message <> Reply to Message

chickenk wrote on Mon, 25 August 2008 05:18luzr wrote on Sun, 24 August 2008 18:56Software rendering and Draw: Finally do it... most likely based on Cairo (or cairo codebase BTW, ideally, we should be able to e.g. draw PNG images WITHOUT having X11 libraries on the system. That would be perfect for www/server applications. (Speaking about it, in future, I would like to try some nice web toolkit).

May I ask if the Cairo choice is immutable or not ? if there is place for some discussion here, I would simply like to bring to your attention some other great rendering libraries:
[list type=square]
[*] AntiGrain (http://www.antigrain.com) -> some work already done, check for topics in this forum)

Yep, that is potential alternative.

Mirek

Subject: Re: The plan....
Posted by cbpporter on Mon, 25 Aug 2008 10:07:46 GMT
View Forum Message <> Reply to Message

luzr wrote on Sun, 24 August 2008 19:56
Next "true unicode" support. That means going wchar -> int, interpreting such WStrings in Draw correctly, also font replacement in X11. Later RTL support.

I'm a little busy right now and I'll continue to be for the next 2-3 weeks. But after I'll be happy to post a cleaned up version of my patches for Unicode support, which include: surrogate pair aware conversions, text metrics update, Draw, Label and EditField. Also I have some light font substitution mechanism for Win32 for non BMP chars.

Subject: Re: The plan....
Posted by captainc on Mon, 25 Aug 2008 16:45:52 GMT
View Forum Message <> Reply to Message

Quote:- maybe some kind of ORM.
What do you mean? How different from current SQL packages?
I use SqlAlchemy for Python, and it is a great tool; I am much more productive with it. Though, the effort that would go into building such a system would be great.

Subject: Re: The plan....
Posted by unodgs on Mon, 25 Aug 2008 18:08:34 GMT
View Forum Message <> Reply to Message

captainc wrote on Mon, 25 August 2008 12:45Quote:- maybe some kind of ORM.
What do you mean? How different from current SQL packages?
I use SqlAlchemy for Python, and it is a great tool; I am much more productive with it. Though, the effort that would go into building such a system would be great.
I mean something on top sql package. I don't have a clear vision yet (for my current project I wrote simply table->entity class generator for postgres and I thought it'd be great to have something similar but much more flexible), I'm open to any ideas, sugesstions. I'm looking at hibernate, linq,

AR from RoR and I'll take a look at sqlalchemy (never used python so it's new to me) Thanks!

Subject: Re: The plan....
Posted by cbpporter on Mon, 25 Aug 2008 23:08:02 GMT
View Forum Message <> Reply to Message

Other things to consider:
- JPG2000 plugin, maybe via JasPer
- some extensions to Qtf, most noticeably for non-antialiased fonts

Subject: Re: The plan....
Posted by mirek on Tue, 26 Aug 2008 07:09:21 GMT
View Forum Message <> Reply to Message

Quote:
- some extensions to Qtf, most noticeably for non-antialiased fonts

What is wrong with them?

Mirek

Subject: Re: The plan....
Posted by cbpporter on Tue, 26 Aug 2008 07:44:49 GMT
View Forum Message <> Reply to Message

luzr wrote on Tue, 26 August 2008 10:09
What is wrong with them?

AFAIK, there is no Qtf tag to specify that I want a font rendered with or without antialiasing. And on top of that, I would like a mechanism to apply all the formating available in a font to a string and get the corresponding Qtf (and vice-versa maybe too).

Subject: Re: The plan....
Posted by amrein on Wed, 27 Aug 2008 19:33:02 GMT
View Forum Message <> Reply to Message

Clean source code directory:

./examples
./examples/applications
./examples/references

./examples/manual
./manuals
./source
./source/3rdparty
./source/theide
./source/theide/callbackhandler
./source/theide/coff
./source/theide/coff/binobj
./source/theide/coff/uar
./source/theide/coff/uld
./source/theide/crashhandler
./source/theide/sqlcommander
./source/theide/topic
./source/theide/translator
./source/theide/updater
./source/theide/versioning
./source/upp
./source/upp/core
./source/upp/database
./source/upp/database/mysql
./source/upp/database/odbc
./source/upp/database/postgresql
./source/upp/database/sqlite
./source/upp/dbus
./source/upp/gui
./source/upp/gui/draw
./source/upp/gui/controls
./source/upp/gui/dialogs
./source/upp/gui/styles
./source/upp/html
./source/upp/network
./source/upp/opengl
./source/upp/richtext
./source/upp/script
./source/upp/sound
./source/upp/video
./source/upp/xml
./source/templates
./source/plugins

Subject: Re: The plan....
Posted by cbpporter on Wed, 27 Aug 2008 20:50:17 GMT
View Forum Message <> Reply to Message

amrein wrote on Wed, 27 August 2008 22:33Clean source code directory
That kind of breaks the idea of packages and assemblies. But still, I would separate U++ the library from TheIDE, umk and other tools like coff. I see no good reason why they can't have their

own assembly. It would also help when you are inserting a package dependency and are shown all the packages, including TheIDE packages. I know they are just a few, but still: what are the chances of me needing one of those packages if I'm not working on TheIDE (i.e. in MyApps).

Subject: Re: The plan....
Posted by amrein on Wed, 27 Aug 2008 22:08:55 GMT
View Forum Message <> Reply to Message

Quote:
That kind of breaks the idea of packages and assemblies.

In fact, no.

I just need to modify "*.upp" files, they will still find other assembly.

Quote:
But still, I would separate U++ the library from TheIDE, umk and other tools like coff. I see no good reason why they can't have their own assembly.

Because they are not stand alone..

Quote:
It would also help when you are inserting a package dependency and are shown all the packages, including TheIDE packages. I know they are just a few, but still: what are the chances of me needing one of those packages if I'm not working on TheIDE (i.e. in MyApps).

So 7 assemblies:

- myapps
- examples
- templates
- upp
- theide
- 3rdparty
- plugins

Subject: Re: The plan....
Posted by mirek on Thu, 28 Aug 2008 03:21:09 GMT
View Forum Message <> Reply to Message

amrein wrote on Wed, 27 August 2008 18:08
So 7 assemblies:

- myapps
- examples
- templates
- upp
- theide
- 3rdparty
- plugins

I guess you still do not quite get the concept....

Anyway, I agree that theide could perhaps be moved to separate one.

Mirek

Subject: Re: The plan....
Posted by mirek on Thu, 28 Aug 2008 07:02:28 GMT
View Forum Message <> Reply to Message

Another issue that we can perhaps solve after all:

Maybe we should make CtrlCore fully MT safe (in common sense, where you are required to serialize access to single widget by multiple threads).

PostCallback is good, but sometimes more straighforward solution would be even better...

Mirek

Subject: Re: The plan....
Posted by kov_serg on Thu, 28 Aug 2008 07:23:41 GMT
View Forum Message <> Reply to Message

It needs class or component to draw plots and charts and store it to file (jpg,pdf,ps...) or to simply view it on screen or include to reports. No real need in interactive chart simply objects to build and render plots. It could be very usefull to represent results of calculations or thatever.

Subject: Re: The plan....

Posted by unodgs on Thu, 28 Aug 2008 07:56:30 GMT
View Forum Message <> Reply to Message

luzr wrote on Thu, 28 August 2008 03:02Another issue that we can perhaps solve after all:

Maybe we should make CtrlCore fully MT safe (in common sense, where you are required to serialize access to single widget by multiple threads).

PostCallback is good, but sometimes more straighforward solution would be even better...

Mirek
+ you have my vote

Subject: Re: The plan....
Posted by amrein on Thu, 28 Aug 2008 09:16:26 GMT
View Forum Message <> Reply to Message

luzr wrote on Thu, 28 August 2008 05:21amrein wrote on Wed, 27 August 2008 18:08
So 7 assemblies:

- myapps
- examples
- templates
- upp
- theide
- 3rdparty
- plugins

I guess you still do not quite get the concept....

Anyway, I agree that theide could perhaps be moved to separate one.

Mirek

I messed up with packages and assemblies.

What I wanted to say:

- "Assembly" could be renamed "Collection"
- TheIDE and other packages related to "TheIDE only" could be moved into only one Collection (TheIDE collection/assembly)
- Other collections could be - myapps - examples - upp - 3rdparty - plugins.
- upp/source won't be a collection/assembly any more, but the default folder to search for

Ultimate++ official collection.

Any collection/assembly must have a small explanation like packages do at present. Something like "Templates collection : to use one of those packages for your application, create a new package in MyApps and select template x"

That way, when I start TheIDE, I know what I do. I work on MyApps, Examples, on Templates, on TheIDE source, on upp...

If I'm still wrong, please correct me.

Subject: Re: The plan....
Posted by mr_ped on Thu, 28 Aug 2008 10:51:43 GMT
View Forum Message <> Reply to Message

Well, assemblies don't just "collect" packages, but they also handle inclusion of other assemblies in build process, i.e. after default 2008.1 install at Win32 you can't work with Bazaar packages in MyApps assembly (until you add that in Edit assembly).
This also allow you to have two uppsrc (stable / SVN), and assemblies MyApps/MyAppsSVN, which both refer to "Core" package, but from different assembly (version), and work on/build both in the same TheIDE.

In this sense I think the "assembly" is better name than "collection".

Subject: Re: The plan....
Posted by zsolt on Thu, 28 Aug 2008 12:21:08 GMT
View Forum Message <> Reply to Message

luzr wrote on Thu, 28 August 2008 09:02
Maybe we should make CtrlCore fully MT safe (in common sense, where you are required to serialize access to single widget by multiple threads).

PostCallback is good, but sometimes more straighforward solution would be even better...

Mirek

A thread safe Callback system would be much more useful, I think.

Subject: Re: The plan....
Posted by mirek on Thu, 28 Aug 2008 12:46:14 GMT
View Forum Message <> Reply to Message

zsolt wrote on Thu, 28 August 2008 08:21luzr wrote on Thu, 28 August 2008 09:02
Maybe we should make CtrlCore fully MT safe (in common sense, where you are required to

serialize access to single widget by multiple threads).

PostCallback is good, but sometimes more straighforward solution would be even better...

Mirek

A thread safe Callback system would be much more useful, I think.

Callback itself is thread-safe.

And, I guess, PostCallback method can be considered a thread-safe callback system; that is what we have now...

What else should it do? (Note that it is technically impossible to call another thread's routine - that would be "interrupt").

Mirek

Subject: Re: The plan....
Posted by mirek on Thu, 28 Aug 2008 13:31:47 GMT
View Forum Message <> Reply to Message

Another: Win32 release should perhaps include some more free-license libraries, like PostgreSQL, SSL etc.. to simplify setup. Maybe even mysql.

Mirek

Subject: Re: The plan....
Posted by zsolt on Thu, 28 Aug 2008 13:54:30 GMT
View Forum Message <> Reply to Message

luzr wrote on Thu, 28 August 2008 14:46zsolt wrote on Thu, 28 August 2008 08:21luzr wrote on Thu, 28 August 2008 09:02
Maybe we should make CtrlCore fully MT safe (in common sense, where you are required to serialize access to single widget by multiple threads).

PostCallback is good, but sometimes more straighforward solution would be even better...

Mirek

A thread safe Callback system would be much more useful, I think.

Callback itself is thread-safe.

And, I guess, PostCallback method can be considered a thread-safe callback system; that is what we have now...

What else should it do? (Note that it is technically impossible to call another thread's routine - that would be "interrupt").

Mirek

I would like an inter-thread communication framework:

threads are waiting for a job queue
threads can have special job-creating Callback like objects
threads can register their methods to that Callback like objects of other threads (in a type-safe way, like with current callback functions or THISBACK macros)
when a thread calls it's own Callback like object, it creates the job-objects and puts them to the fifos of registered threads
threads are executing jobs from fifos on their own threads

Subject: Re: The plan....
Posted by mr_ped on Thu, 28 Aug 2008 15:11:38 GMT
View Forum Message <> Reply to Message

luzr wrote on Thu, 28 August 2008 15:31Another: Win32 release should perhaps include some more free-license libraries, like PostgreSQL, SSL etc.. to simplify setup. Maybe even mysql.

Mirek

Yes, in ideal world I would personally deliver "online" and "offline" install.exe. The online would be just couple of kB file, which would let user configure what should be installed. (U++, MINGW, SDL, PSQL, ...) Those parts would be downloaded than, and installed.
Offline installer would contain everything (so it would be several MB big file), but it would work without internet connection. (visually it would behave in the same way as online installer)

I think right now we are at "offline" installer, with having several different .exe with different things bundled (right now just MINGW vs non-MINGW). Sticking to this path may lead to 4-5 installers maybe (some common configurations). As the build process will be fully automated, I think this is still viable to maintain and usable. The online/offline model would just lessen the strain on newcomers, so they would need to choose just from 2 options, not 4-5 of them.

Edit:
and the "online" installer should be usable to ADD some package later after the U++ is already installed without it. Or even "remove" some part of it (although I think that's really just adding some polish, not really needed function, the ADD is important part).

Subject: Re: The plan....
Posted by mirek on Thu, 28 Aug 2008 15:14:05 GMT
View Forum Message <> Reply to Message

zsolt wrote on Thu, 28 August 2008 09:54
I would like an inter-thread communication framework:

threads are waiting for a job queue
threads can have special job-creating Callback like objects
threads can register their methods to that Callback like objects of other threads (in a type-safe way, like with current callback functions or THISBACK macros)
when a thread calls it's own Callback like object, it creates the job-objects and puts them to the fifos of registered threads
threads are executing jobs from fifos on their own threads

I believe this is very specific scenario. Sometimes, this can be handy, but I doubt it is something common.

Mirek

Subject: Re: The plan....
Posted by amrein on Thu, 28 Aug 2008 17:15:12 GMT
View Forum Message <> Reply to Message

mr_ped wrote on Thu, 28 August 2008 12:51Well, assemblies don't just "collect" packages, but they also handle inclusion of other assemblies in build process, i.e. after default 2008.1 install at Win32 you can't work with Bazaar packages in MyApps assembly (until you add that in Edit assembly).
This also allow you to have two uppsrc (stable / SVN), and assemblies MyApps/MyAppsSVN, which both refer to "Core" package, but from different assembly (version), and work on/build both in the same TheIDE.

In this sense I think the "assembly" is better name than "collection".

Well, for me, assemblies are collection of packages. Packages refer to other packages from uppsrc/ (uppsrc is the main assembly) and not to other assemblies.

So, for me my proposition is still valid.

luzr wrote on Thu, 28 August 2008 15:31Another: Win32 release should perhaps include some more free-license libraries, like PostgreSQL, SSL etc.. to simplify setup. Maybe even mysql.

Mirek

Well: www.graphicsmagick.org or www.imagemagick.org to support reading, writing, and manipulating images (over 88 or more major image formats supported).

Subject: Re: The plan....
Posted by captainc on Thu, 28 Aug 2008 18:07:36 GMT
View Forum Message <> Reply to Message

Quote:Well: www.graphicsmagick.org or www.imagemagick.org to support reading, writing, and manipulating images (over 88 or more major image formats supported).
That is a GREAT idea. 'Motion Seconded'!

Subject: Re: The plan....
Posted by mirek on Thu, 28 Aug 2008 19:34:37 GMT
View Forum Message <> Reply to Message

amrein wrote on Thu, 28 August 2008 13:15mr_ped wrote on Thu, 28 August 2008 12:51Well, assemblies don't just "collect" packages, but they also handle inclusion of other assemblies in build process, i.e. after default 2008.1 install at Win32 you can't work with Bazaar packages in MyApps assembly (until you add that in Edit assembly).
This also allow you to have two uppsrc (stable / SVN), and assemblies MyApps/MyAppsSVN, which both refer to "Core" package, but from different assembly (version), and work on/build both in the same TheIDE.

In this sense I think the "assembly" is better name than "collection".

Well, for me, assemblies are collection of packages. Packages refer to other packages from uppsrc/ (uppsrc is the main assembly) and not to other assemblies.

Assemblies are ordered sets (ok, collections as good term as set) of "nests". Nests are directories where packages reside.

Well, in practice, assembly has often just two nests in it (second being uppsrc), but that is not a rule.

Mirek

Subject: Re: The plan....
Posted by mirek on Fri, 29 Aug 2008 11:49:17 GMT
View Forum Message <> Reply to Message

More:

Decimal type (we have begun working on it; my version is in uppdev/Dec, I am now stuck with mul/div

Move SHA1 and MD5 to Core; perhaps derive it from Stream.

Mirek

Subject: Re: The plan....
Posted by unodgs on Fri, 29 Aug 2008 12:05:04 GMT
View Forum Message <> Reply to Message

luzr wrote on Fri, 29 August 2008 07:49More:

Decimal type (we have begun working on it; my version is in uppdev/Dec, I am now stuck with mul/div

Move SHA1 and MD5 to Core; perhaps derive it from Stream.

Mirek
Just one note about decimal. Unfortunatelly MSC8/9 32bit doesn't support int128 type...

Subject: Re: The plan....
Posted by mirek on Fri, 29 Aug 2008 18:12:40 GMT
View Forum Message <> Reply to Message

amrein wrote on Wed, 27 August 2008 18:08
So 7 assemblies:

"nests", right?

Quote:
- myapps
- examples
- templates
- upp
- theide
- 3rdparty
- plugins

- What is the point of joining all 3 kinds of examples into single dir?

- What is "templates" good for?

- Having separate nests for 3rdparty and plugins would lead to more complicated assemblies - you would have to add 3 nests instead of one. IMO it is better to have it as subdir in the nest, which is current status.

OTOH, to truly separate 3rdparty stuff, we MIGHT consider putting 3rdparty folder in uppsrc and keep only U++ interfacing stuff (-> our core) in "plugin". The grand question is whether is it worth the trouble...

Mirek

Subject: Re: The plan....
Posted by mirek on Fri, 29 Aug 2008 18:19:28 GMT
View Forum Message <> Reply to Message

Hm, maybe even better solution to 3rdparty problem is to have direct license support in package system...

I mean, packages could know what is their license and TheIDE can offer license overview or something like that...

Mirek

Subject: Re: The plan....
Posted by amrein on Fri, 29 Aug 2008 20:06:14 GMT
View Forum Message <> Reply to Message

luzr wrote on Fri, 29 August 2008 20:12amrein wrote on Wed, 27 August 2008 18:08
So 7 assemblies:

"nests", right?

Yes.

For me, those two words "Assembly" and "Nest" are complicated and should disappear. "Assembly", "Nest" and "Package" organisation could be break into "Collection","Directory" and "Package" (when someone say "Garbage collector" in java, everyone understand what he means). I don't hate birds ("Nest"...) but a few newcomers are still confused by those names (me included, I had to read the doc 4 times).

Example: in the first TheIDE window (when you open it), there are two rows "Assembly" and "Packages". That create confusion.

In fact, with this window, you are watching packages from the Nests of the default TheIDE

Assembly right?
(If I'm wrong about all this, I hope I won't find a gun too close to me)
If you want to create another Assembly, you need at present to do it manually.

Now, lets break this into "Collection","Directories" and "Packages":

- The main title of the window become "Select main package from "Default Collection" ".
- First row title: "Directories"
- Second row title: "Packages"

This doesn't break the "Assembly" original idea. Anyone could create is own Collection/Assembly if he needs to.

Quote:Quote:
- myapps
- examples
- templates
- upp
- theide
- 3rdparty
- plugins

- What is the point of joining all 3 kinds of examples into single dir?

To clean directory listing in c:/upp. Could become for example:

- upp/myapps (your own packages)
- upp/exemples (training packages)
- upp/manual (documentation for learning)
- upp/source (the core: libupp, tools and plugins)

With examples: "applications" = complete apps, "references" = small apps to demonstrate one feature, "manual" = examples from the manual.

With 3 kinds of examples in one directory, their is a need perhaps to add something else: be able to group subdir. The main window in TheIDE use symbol like "(+)" to close or open group of files. Here, the idea is for all Nests without packages but having subdir with packages to be able to open or close groups to see those sub-packages with "(+)".

Quote:
- What is "templates" good for?

It's where people can create of fix templates for TheIDE. It's the default TheIDE template directory. Could be in TheIDE directory though.

When you create your new application, TheIDE already asks you to choose one of them at present.

Another reason is to transform TheIDE into TheLinuxWindowsIDE with templates and support for many other toolkits. A tool you will want to use for any projects from DLL to .so, from Qt applications to GTK+, MFC, wxWindows.

I don't remember who said something like this: "Keep your friend close to you. Keep your enemies even closer.". They are not enemies, I know. The idea is to open TheIDE doors to all kind of developers. Become the most used IDE. If someone wants to create a new application, he should think TheIDE first.

Quote:
- Having separate nests for 3rdparty and plugins would lead to more complicated assemblies - you would have to add 3 nests instead of one. IMO it is better to have it as subdir in the nest, which is current status.

- When they are from U++ team, sub directories in a packages are better.
- "3rdparty" is a directory containing packages not from U++ team. They could be useful for other applications.
- "Plugins" could be external sql plugins, image format plugins, codecs plugins. The idea of plugins is to open U++ to third party plugins and be able to handle them dynamically at runtime. They could link upplib to "3rdparty" too for example.

Last idea, with 3rdparty and plugins, is to be able to link TheIDE dynamically with 3rdparty lib not in source/3rdparty but in /usr/lib. Why? Because 3rdparty applications receive security fix from main Linux distributor. This is a bit far from Windows security. No need to rebuild your application if the linux distributor has patched one of the dynamically linked library. No need to wait from next U++ release to get those packages fix.

Quote:
OTOH, to truly separate 3rdparty stuff, we MIGHT consider putting 3rdparty folder in uppsrc and keep only U++ interfacing stuff (-> our core) in "plugin". The grand question is whether is it worth the trouble...

Mirek

The main idea is be able to say: U++ is a "Collection" or "Directories" containing "Packages". TheIDE is one of those directories. IDE and Topic++ are packages from TheIDE directory.

More complicated:
A directory contain a complete project with many packages.
Subdirectories are named packages when they contain something to build with a .upp file.
Subsubdirectories are subpackages of packages.

Subject: Re: The plan....
Posted by mirek on Fri, 29 Aug 2008 20:38:35 GMT
View Forum Message <> Reply to Message

Quote:
In fact, with this window, you are watching packages from the Nests of the default TheIDE Assembly right?

Incorrect. You are seeing predefined assemblies.

Quote:
Now, lets call them "Collection","Directories" and "Packages":

I am responding to this in another thread.

Quote:
- The main title of the window become "Select main package from "Default Collection" ".
- First row title: "Directory"
- Second row title: "Packages"

Wrong.

Quote:
To clean directory listing in c:/upp. Would become:

- upp/myapps (your own packages)
- upp/exemples (training packages)
- upp/manual (documentation for learning)
- upp/source (the core: libupp, tools and plugins)

Well, I see no big advantage, but you can do this and maintain original 3 nests too. OTOH, I think the directory structure is not that relevant, once predefined assemblies are setup.

Quote:
With examples: "applications" = complete apps, "references" = small apps to demonstrate one feature, "manual" = examples from the manual.

BTW, now you suggest 2 instead of 3. In previous post you suggested 1 instead of 3

Quote:
With 3 kinds of examples in one directory, their is a need to add something else: be able to group subdir. The main window in TheIDE use symbol like "(+)" to close or open group of files. Here, the idea is for all Nests without packages but having subdir with packages to be able to open or close groups to see those sub-packages with "(+)".

Great. So we are going to go back to old tired concepts?

I agree that it is less confusing for beginners, but I am not going to sacrifice my productivity

Quote:
Quote:
- What is "templates" good for?

It's where people can create of fix templates for TheIDE. It's the default TheIDE template directory. Could be in TheIDE directory though.

What is template for TheIDE?

Quote:
When you create your new application, TheIDE already asks you to choose one of them at present.

Ops. But that logically belongs to packages - and that is why all templates are related to packages. E.g. GUI project templates reside in CtrlLib, because that is the package you need for GUI.

Quote:
Another reason is to transform TheIDE into TheLinuxWindowsIDE with templates and support for many other toolkits. A tool you will want to use for any projects from DLL to .so, from Qt applications to GTK+, MFC, wxWindows.

OK. That is valid point... Current status is that such templates really are in Config directory (~/.upp/theide in linux now AFAIK).

Quote:
They could link upplib to "3rdparty" too for example.

Last idea, with 3rdparty and plugins, is to be able to link TheIDE dynamically with 3rdparty lib not in source/3rdparty but in /usr/lib.

If you are going to use .so, you do not need to provide source code at all.

Mirek

Subject: Re: The plan....
Posted by amrein on Fri, 29 Aug 2008 21:57:41 GMT
View Forum Message <> Reply to Message

luzr wrote on Fri, 29 August 2008 22:38Quote:
In fact, with this window, you are watching packages from the Nests of the default TheIDE Assembly right?

Incorrect. You are seeing predefined assemblies.

Where is my gun.
I will continue this discussion in the other thread to let people talk here about what is coming next in U++.

Subject: Re: The plan....
Posted by cocob on Sat, 30 Aug 2008 10:31:10 GMT
View Forum Message <> Reply to Message

From my side

- A powerfull graphical profiler
- A graphical fronted for valgrind
- Better debugger integration
- Always more and more performances .... !

cocob

Subject: Re: The plan....
Posted by Oblivion on Mon, 06 Oct 2008 18:50:20 GMT
View Forum Message <> Reply to Message

It may sound silly but how about embedding a configurable feed (RDF/RSS/Atom) reader (which would function like the VisualStudio's News reader, and could be opened in a a tab) using the power of QTF into theIDE ? By this way, it may be possible to reach U++ forum content online too. I'm already working on two feed readers (a simple feed ctrl and a "Rich" feed ctrl), so I can handle that part.

Subject: Re: The plan....
Posted by koldo on Tue, 07 Oct 2008 06:20:26 GMT
View Forum Message <> Reply to Message

Hello all

Things that I would add:

- A String type that would be truly unicode, by default. I have found some mistakes in U++ code because it has been used String instead of WString.

- A multimedia (Video) control

- Also interesting to have an option to download Postgre, MySql, etc. so they fit perfectly with Upp.

Perhaps we may choose the main things listed in this post and let the volunteers to choose which one is going to implement.

Thank you very much
Koldo

Subject: Re: The plan....
Posted by mirek on Tue, 07 Oct 2008 09:07:01 GMT
View Forum Message <> Reply to Message

koldo wrote on Tue, 07 October 2008 02:20Hello all

Things that I would add:

- A String type that would be truly unicode, by default. I have found some mistakes in U++ code because it has been used String instead of WString.

I am afraid you have to be more specific about this....

Also, FYI, WString is goint to be 32-bit soon.

Mirek

Subject: Re: The plan....
Posted by koldo on Tue, 07 Oct 2008 09:59:19 GMT
View Forum Message <> Reply to Message

Hello luzr

I mean, as String seems to be the by default option, it tends to produce some mistakes as programmers tend to use it in functions that usually are going to be used worldwide.

Perhaps it would be better that the by default String would be international. In addition it would be possible to have other types with smaller sizes for special things like binary files management.

Perhaps there could be compatibility problems with old code.

Best regards
Koldo

Subject: Re: The plan....
Posted by mirek on Tue, 07 Oct 2008 10:32:20 GMT
View Forum Message <> Reply to Message

koldo wrote on Tue, 07 October 2008 05:59Hello luzr

I mean, as String seems to be the by default option, it tends to produce some mistakes as programmers tend to use it in functions that usually are going to be used worldwide.

It works quite well as long as you use UTF-8.

Quote:
Perhaps it would be better that the by default String would be international. In addition it would be possible to have other types with smaller sizes for special things like binary files management.

Actually, I would say that String is used to store texts in only about 50% of usages, i18n texts in only 20%

Anyway, I believe that the correct way is UTF-8. Note also that there is HUGE performace difference between String and WString (more than magnitude). So if you e.g. have any map of names (of e.g. employees), your software might run 10x faster if they are stored as UTF-8.

Of course, this also means WString<->String flupping if you are about to manipulate individual characters, but IME this is not that big problem.

Mirek

Subject: Re: The plan....
Posted by koldo on Tue, 07 Oct 2008 11:13:37 GMT
View Forum Message <> Reply to Message

Hello luzr

Sorry, I thought that to comply with UTF8 it was necessary to use WString.

In some minutes I am going to report an error in FileSel.cpp related with this, really confirming it:

The struct FileIconMaker has a member declared String file. If the file name has non english characters the function virtual Image Make() does not work.

To solve it I simply change this in FileSel.cpp:

```
for(int i = 0; i < 2; i++) {
    SHFILEINFO info;
    SHGetFileInfo(file, dir ? FILE_ATTRIBUTE_DIRECTORY : FILE_ATTRIBUTE_NORMAL,
&info, sizeof(info),
```

with this;

```
WString wfile = file.ToWString();
for(int i = 0; i < 2; i++) {
    SHFILEINFOW info;
    SHGetFileInfo(wfile, dir ? FILE_ATTRIBUTE_DIRECTORY : FILE_ATTRIBUTE_NORMAL,
&info, sizeof(info),
```

and it works.

It is not a problem with String type. It is just that we have to check our code with international samples!.

Best regards
Koldo

Subject: Re: The plan....
Posted by Sender Ghost on Wed, 17 Dec 2008 18:18:03 GMT
View Forum Message <> Reply to Message

luzr wrote on Sun, 24 August 2008 22:56
OK, feel free to add to the list.

Mirek

Firstly, we can use static code analysis tools to improve existing Ultimate++ code base, such as (open source; in alphabetical order): Clang Checker, CppCheck, FlawFinder, Sparse, Splint, Yasca and so on. Or send request to Coverity as open source software (e.g. the LAMP stack). Example of FlawFinder v1.27 output you can see in the file attachment.
Also, we can create own static analysis tool inside TheIDE based on (future preprocessed) Assist++ information and existing (probably converted) analysis rules. Examples: Eclipse PDT, JetBrains IntelliJ IDEA.

Secondly, add support for LLVM front end for GNU GCC as compiler and linker.

File Attachments

1) upp_report.zip, downloaded 376 times