Subject: ide: main.conf
Posted by mirek on Mon, 26 Aug 2013 11:57:48 GMT
View Forum Message <> Reply to Message

New feature of ide (and umk) build system:

All files named "main.conf" from all packages are appended together (in unspecified order), result placed on output directory of main package into main.conf.h file (always, even if it is empty). Path to this directory is placed to the include set. Ide/umk supporting this feature defines MAIN_CONF macro.

The purpose of this is to make possible a configuration of common packages (like Draw or CtrlCore) by adding other packages. For example, we can have package that provides custom font subsystem, this package would put "#define CUSTOM_FONTSYS" in its main.conf, which in turn would exclude generic font handling in Draw.

Mirek

Subject: Re: ide: main.conf
Posted by dolik.rce on Sun, 08 Sep 2013 08:10:48 GMT
View Forum Message <> Reply to Message

mirek wrote on Mon, 26 August 2013 13:57New feature of ide (and umk) build system:

All files named "main.conf" from all packages are appended together (in unspecified order), result placed on output directory of main package into main.conf.h file (always, even if it is empty). Path to this directory is placed to the include set. Ide/umk supporting this feature defines MAIN_CONF macro.
Hi Mirek

This will make it a bit harder again to use U++ sources without TheIDE (or umk). As I try to maintain the universal makefile fully compatible with TheIDE, this is not really a good news for me So I have few questions:

Do you plan to use this in any of the core packages soon? I've found no mention of this in svn so far.

Will it always be guarded by #ifdef MAIN_CONF for backwards compatibility? Or at least for some period of time, to allow transition to newer build system (be it TheIDE or makefile)?

Wouldn't it be better to use existing "Add/Remove flags" functionality of Package manager for this? To use your example, the package that wants to include it own font handling could just have "flags CUSTOM_FONTSYS;" in .upp file and Draw package would then turn off the generic font handling based on #ifdef flagCUSTOM_FONTSYS. I'm aware that main.conf as you designed it will allow much more than this, but I'm not sure if it is really necessary.

Best regards,

Honza

Subject: Re: ide: main.conf
Posted by mirek on Sun, 08 Sep 2013 09:04:39 GMT
View Forum Message <> Reply to Message

dolik.rce wrote on Sun, 08 September 2013 04:10mirek wrote on Mon, 26 August 2013 13:57New feature of ide (and umk) build system:

All files named "main.conf" from all packages are appended together (in unspecified order), result placed on output directory of main package into main.conf.h file (always, even if it is empty). Path to this directory is placed to the include set. Ide/umk supporting this feature defines MAIN_CONF macro.
Hi Mirek

This will make it a bit harder again to use U++ sources without TheIDE (or umk). As I try to maintain the universal makefile fully compatible with TheIDE, this is not really a good news for me So I have few questions:

I understand that. I think that theide will always be possible to build without this feature.

Quote:
Do you plan to use this in any of the core packages soon? I've found no mention of this in svn so far.

Will it always be guarded by #ifdef MAIN_CONF for backwards compatibility? Or at least for some period of time, to allow transition to newer build system (be it TheIDE or makefile)?

I plan to something like

```
#ifdef MAIN_CONF
#include "main.conf.h"
#endif
```

into Core/Core.h soon - and that is about it, nothing else is ever required... Which means that for now, no changes are required as well.

Quote:
Wouldn't it be better to use existing "Add/Remove flags" functionality of Package manager for this? To use your example, the package that wants to include it own font handling could just have "flags CUSTOM_FONTSYS;" in .upp file and Draw package would then turn off the generic font handling based on #ifdef flagCUSTOM_FONTSYS. I'm aware that main.conf as you designed it will allow much more than this, but I'm not sure if it is really necessary.

I do not really see a difference with breaking stuff one way or another...

Add/remove packages work for package compiled and do not allow to define values. I can sure invent some new "Add/remove global flags", but I really am not sure how would that improve the situation.... Build system has to be adjusted anyway and it should be no harder to implement main.conf than new flags, I think...

All that said, main.conf is right now just experimental attempt to solve given issue. I might be possible that we will find another solution to this...

Mirek

Subject: Re: ide: main.conf
Posted by dolik.rce on Sun, 08 Sep 2013 14:50:01 GMT
View Forum Message <> Reply to Message

mirek wrote on Sun, 08 September 2013 11:04Quote:Wouldn't it be better to use existing "Add/Remove flags" functionality of Package manager for this? To use your example, the package that wants to include it own font handling could just have "flags CUSTOM_FONTSYS;" in .upp file and Draw package would then turn off the generic font handling based on #ifdef flagCUSTOM_FONTSYS. I'm aware that main.conf as you designed it will allow much more than this, but I'm not sure if it is really necessary.

I do not really see a difference with breaking stuff one way or another...
The difference is that the stuff with flags is already implemented in the Makefile The main.conf.h would mean to add that feature. It sure can be done, it just additional obstacle.

mirek wrote on Sun, 08 September 2013 11:04Add/remove packages work for package compiled and do not allow to define values.
Yes, it does, perhaps not intentionally You can define a flag as XYZ=42 in package manager and it results in -DflagXYZ=42 being appended to the compiler command line.

Anyway, if you decide to use it somewhere in the core packages, please let me know that I have to implement matching functionality. I use the universal makefile not only for compilation of theide in the linux packages but also for my own projects (e.g. to deploy them on OpenShift).

Honza

Subject: Re: ide: main.conf
Posted by Sender Ghost on Mon, 09 Sep 2013 04:00:23 GMT
View Forum Message <> Reply to Message

Hello, Mirek and Jan.
mirek wrote on Mon, 26 August 2013 13:57The purpose of this is to make possible a configuration of common packages (like Draw or CtrlCore) by adding other packages. For example, we can

have package that provides custom font subsystem, this package would put "#define CUSTOM_FONTSYS" in its main.conf, which in turn would exclude generic font handling in Draw.
dolik.rce wrote on Sun, 08 September 2013 10:10Wouldn't it be better to use existing "Add/Remove flags" functionality of Package manager for this? To use your example, the package that wants to include it own font handling could just have "flags CUSTOM_FONTSYS;" in .upp file and Draw package would then turn off the generic font handling based on #ifdef flagCUSTOM_FONTSYS.
I also think, that this is already possible with using current flags system.

The Draw package might accept CUSTOM_FONTSYS flag and check it with following source code:

```
#include "Draw.h"

#ifndef flagCUSTOM_FONTSYS
// Use generic implementation
Size Draw::GetPixelsPerInch() const
{
 return IsNative() ? GetNativeDpi() : Dots() ? Size(600, 600) : Size(96, 96);
}
// ...
#endif
```

The DrawCustom package might use Draw package with own implementation:

```
#include <Draw/Draw.h>

// Use custom implementation
Size Draw::GetPixelsPerInch() const
{
 return Size(96, 96);
}
// ...
```

Then main application (or some Rainbow package) could use DrawCustom package, when CUSTOM_FONTSYS enabled for main package configuration and its implementation will be active.

I created sample application which prints different Draw package description, when CUSTOM_FONTSYS flag enabled. The Draw and DrawCustom packages has its own implementation of this.

Overall, it depends from concrete implementation and external interface.

Edit: Added example with possible Rainbow configuration (for CustomRainbow nest), which doesn't require CUSTOM_FONTSYS for main package configuration, when using Rainbow package.

File Attachments

1) CustomDraw.zip, downloaded 276 times

Subject: Re: ide: main.conf
Posted by mirek on Mon, 09 Sep 2013 11:15:16 GMT
View Forum Message <> Reply to Message

I do not know, I might be doing something wrong, but all I am getting is

Draw.obj : error LNK2005: "char const * __cdecl DrawDescription(void)" (?DrawDescription@@YAPBDXZ) already defined in Draw.obj

which is understandable, as add/remove flags only affects the package where it is defined, AFAIK

Subject: Re: ide: main.conf
Posted by Sender Ghost on Mon, 09 Sep 2013 14:27:13 GMT
View Forum Message <> Reply to Message

mirek wrote on Mon, 09 September 2013 13:15Draw.obj : error LNK2005: "char const * __cdecl DrawDescription(void)" (?DrawDescription@@YAPBDXZ) already defined in Draw.obj
Looks like compiler/linker issue.
I tested on Windows XP with Microsoft Visual C++ 9.0 (MSC9) and Nuwen GCC 4.8.1, where it works (by linking unimplemented functions to final executable in correct order).
mirek wrote on Mon, 09 September 2013 13:15which is understandable, as add/remove flags only affects the package where it is defined, AFAIK
At least, the example with main package configuration should work.

Ok, this is just an example. The full implementation might require changes to flags system (to support such cases).

Subject: Re: ide: main.conf
Posted by mirek on Tue, 10 Sep 2013 16:58:25 GMT
View Forum Message <> Reply to Message

Sender Ghost wrote on Mon, 09 September 2013 10:27mirek wrote on Mon, 09 September 2013 13:15Draw.obj : error LNK2005: "char const * __cdecl DrawDescription(void)" (?DrawDescription@@YAPBDXZ) already defined in Draw.obj
Looks like compiler/linker issue.
I tested on Windows XP with Microsoft Visual C++ 9.0 (MSC9) and Nuwen GCC 4.8.1, where it works (by linking unimplemented functions to final executable in correct order).
mirek wrote on Mon, 09 September 2013 13:15which is understandable, as add/remove flags only affects the package where it is defined, AFAIK
At least, the example with main package configuration should work.

Ok, this is just an example. The full implementation might require changes to flags system (to support such cases).

Yes, and initially I planned to do just that. But main.conf idea seems to be more generic...

Mirek