U++ - Feature #1211
tar makefile should build umk first, then use it to build theide

07/27/2015 07:45 PM - Miroslav Fidler

Status:	Approved		**Start date:**	07/27/2015
Priority:	Normal		**Due date:**	
Assignee:			**% Done:**	0%
Category:			**Estimated time:**	0.00 hour
Target version:	Release 2015.2		**Spent time:**	0.00 hour

Description

History

#1 - 07/29/2015 09:50 PM - Jan Dolinár

Would it help anything? It would just make the process longer, by compiling a lot of code twice...

I assume this is because of the build_info.h problem, right? The makefile generated by TheIDE should support the build_info.h file, because it can be used in any project. I'll create another issue for that. I believe this one should be closed.

#2 - 07/29/2015 10:49 PM - Zbigniew Rebacz

It would be faster, because umk is smaller than TheIDE. It dosen't contain GUI component, so compilation should be short. When umk starts compiling ide it will force to use all cores/threads. So whole compilation time can be visibly reduce.

Code compilation twice (Maybe we can reduce this code in compiling ide, by simply moving/copy objects):
- Core
- Esc
- ide/Android
- ide/Builders
- ide/Core
- ide/Java
- plugin/bz2
- plugin/pcre
- plugin/z

BTW, it is nice to have umk out of the box especially on POSIX.

#3 - 07/30/2015 07:31 AM - Jan Dolinár

Zbigniew Rebacz wrote:

> *It would be faster, because umk is smaller than TheIDE. It dosen't contain GUI component, so compilation should be short. When umk starts compiling ide it will force to use all cores/threads. So whole compilation time can be visibly reduce.*

That is not exactly true. Make can use multiple cores as well (using -j option). The only speed-up could come from using BLITZ, which is (if I remember correctly) by default disabled for production builds.

> *Code compilation twice (Maybe we can reduce this code in compiling ide, by simply moving/copy objects):*

Moving the objects is probably not safe, because would be compiled with different flags than theide.

If you really want to speed up the build in source tarball and provide better funcionality, I'd suggest using a makefile generated by uppbox/UppBuilder (see https://code.google.com/p/upp-mirror/source/browse/#svn%2Ftrunk%2Fuppbox%2FUppBuilder). It comes with alternative .upp file parser in single file, that takes about a second to compile and can compile any U++ package just as umk, including support for flags etc. I already use this when building the packages for Debian, Ubuntu, Arch, OpenSuse and others and it works quite well.

#4 - 07/30/2015 12:45 PM - Zbigniew Rebacz

I tried compile ide with "make -j", but it crashed my linux. Not full crashed but cpu usage was so high that I can do nothing. Even when I am compiling using "mkfile" or inside ide with all cores, cpu usage is also 100%, but there is not crash.

On my machine it seems that "make -j" took too many files.

Using makefile (mkfile) for the long term is not good also. It cannot deduce .lay and .iml file changes. Evry time I changed iml or lay file or this file is updated from svn. I need to manually remove "_out/ide/GUI...". Of course even when this directory was not removed ide compiles, but it can leads to crashes like "Invalidate memory access".

I believe that umk is more stable than makefile. So it will be good to compile it from makefile. At least it would be good to have something like "make umk".

#5 - 07/30/2015 02:43 PM - Jan Dolinár

Zbigniew Rebacz wrote:

> *I tried compile ide with "make -j", but it crashed my linux. Not full crashed but cpu usage was so high that I can do nothing. Even when I am*
> *compiling using "mkfile" or inside ide with all cores, cpu usage is also 100%, but there is not crash.*
>
> *On my machine it seems that "make -j" took too many files.*

You should read manual before using unknown argument:

    -j [jobs], --jobs[=jobs]
        Specifies the number of jobs (commands) to run simultaneously. If there is more than one -j option, the
        last one is effective. If the -j option is given without an argument, make will not limit the number of
        jobs that can run simultaneously.

So you tried to run thousands of compilers at once, no wonder it killed your computer :-)

> *Using makefile (mkfile) for the long term is not good also.*

Using make is definitely good. Make is a standard tool, available on any platform and the developers are familiar with it. The possibility to export the project from theide as sources + makefile is very important for any developer who needs to distribute their U++ project in source form, because need to install TheIDE/umk would be scaring away their customers.

> *It cannot deduce .lay and .iml file changes. Evry time I changed iml or lay file or this file is updated from svn. I need to manually remove*
> *"_out/ide/GUI...". Of course even when this directory was not removed ide compiles, but it can leads to crashes like "Invalidate memory access".*

AFAIK the exported makefile is generated to specify .iml and .lay files as dependencies. I will try to check this as soon as I get some spare time.

> *I believe that umk is more stable than makefile. So it will be good to compile it from makefile. At least it would be good to have something like "make umk".*

What I proposed in previous comment would allow "make PKG=anypackage". I wouldn't worry about stability, it builds 50+ packages that I maintain on various Linux distributions, every day for almost two years now, without any substantial problems.

#6 - 07/30/2015 04:53 PM - Zbigniew Rebacz

I found the source of issue. We are talking about two makefiles. On in $(UppRoot) and one in $(UppRoot)/uppsrc. The first one doesn't recognize "-j" option. The second works fine with it. So I think in first there are following problems:

```
.PHONY: all install clean

all:
    # I think we should pass -j parameter. "sh domake" dosen't know how many threads we want to use!
    sh domake

install:
    sh doinstall

clean:
    # Should be checked if this dir/file exists -> make clean can output warning!
    rm -r uppsrc/_out
    rm uppsrc/ide.out
```

Of course domake isn't perfect too (I don't fully understand this file, especially the first part):

```
if which pkg-config
then
    sed -e "s@-I((INCLUDES))@`pkg-config --cflags-only-I gtk+-2.0`@g" uppsrc/Makefile.in >uppsrc/Makefile
    sed -e "s@((INCLUDES))@`pkg-config --cflags-only-I gtk+-2.0|sed -e s/-I//g -e \"s/ /;/g\"`@g" GCC.bm.in >GCC.bm
else
    sed -e "s@((INCLUDES))@@g" uppsrc/Makefile.in >uppsrc/Makefile
    sed -e "s@((INCLUDES))@@g" GCC.bm.in >GCC.bm
fi

if [ ! -f /usr/lib/libdl.so ]
then
    cd uppsrc
    sed -e s/-ldl//g Makefile >Makefile2
    rm Makefile
    mv Makefile2 Makefile
    cd ..
fi

# "-j" is never passed!!!
if which gmake
```

```
    then
        gmake -C uppsrc
    else
        make -C uppsrc
    fi

    # Check if file exists -> print errors
    cp uppsrc/ide.out ./theide
```

I think that "make PKG=anypackage" is killer feature. It allows to build UMK :) But if issues that I was writing about in this and previous message will be solved. The makefile could be an interesting alternative.

#7 - 07/31/2015 11:40 AM - Miroslav Fidler

Well, I think we might to redefine this task to

"tarball creates umk too"

As for UppBuilder: that is one awesome piece of code. However, I am worried about maintaining compatiblity with theide UMK - we are adding new features to .upp that have to be reproduced in UppBuilder...

#8 - 07/31/2015 09:28 PM - Jan Dolinár

Zbigniew Rebacz wrote:

> I found the source of issue. We are talking about two makefiles. On in $(UppRoot) and one in $(UppRoot)/uppsrc. The first one doesn't recognize "-j" option. The second works fine with it.

Oh, right, I totally forgot about that one. It can be easily fixed, I'll do that soon. For all those related files.

> Of course domake isn't perfect too (I don't fully understand this file, especially the first part)

The first part of domake tries to perform some detection and fixes some include paths, which can be different on different Linux distributions. It could actually be improved too, I lately found there is more libraries that were moved in higher versions to locations that the makefile doesn't expect.

Miroslav Fidler wrote:

> Well, I think we might to redefine this task to
>
> "tarball creates umk too"

I'd rather say "modernization of tarball makefiles plus umk support" :-)

> As for UppBuilder: that is one awesome piece of code.

Thanks, but it could be better. It still feels quite a bit "hacky" :-)

> *However, I am worried about maintaining compatiblity with theide UMK - we are adding new features to .upp that have to be reproduced in UppBuilder...*

It doesn't need to have all the features of umk. E.g. it works just fine without BLITZ.

As someone said on the forum lately, theide and U++ should be considered two separate products. IMHO each of them can be used without the other. If this should be possible, then the default build system (theide/umk) should not use features which are not easily implemented in other build systems (generated makefile, my universal UppBuilder makefile, or any other).

I'm leaving for a vacation next week and I don't know how much time I'll have for coding there :-) But I'll definitely want to try to do something about the source tarballs. In their current form, they can be used only for one thing, build theide in default configuration (you can't even change buildflags!). My vision is to make it more flexible, modern and easier to use.

I'll create a branch for this, so the nightly builds are not disrupted by my development, and in couple weeks you can both hopefully evaluate whether the new version is worth merging back into trunk.

#9 - 08/01/2015 10:08 PM - Jan Dolinár

So I started playing with this idea... I'm pretty sure I won't have much time for it in next few days, so I at least post here a little preview of what I got so far :-)

[This modified tarball](#) uses the universal makefile generated by UppBuilder. It still doesn't support make -j parameter correctly, but at least it uses as many parallel process as there is CPU cores. See the attached readme file to get an idea of what is possible now. I'd be glad if you let me know if I go in the right direction - before I get too far :-)

#10 - 09/11/2015 09:26 PM - Zbigniew Rebacz

Can you re-upload the file it seems that it expired.

#11 - 09/15/2015 05:29 AM - Jan Dolinár

Here is a new link: https://drive.google.com/open?id=0Bzq_iqIhK8FjUG5VZENZaXNoZms - it should not expire this time :-)

#12 - 09/15/2015 02:53 PM - Zbigniew Rebacz

I type "make umk" and i have following error:

```
make -f mkfile PKG="umk"
make[1]: Wej?cie do katalogu `/home/klugier/upp_temp/upp-x11-src-8782'
/bin/sh: 5: Syntax error: "then" unexpected (expecting "done")
make[1]: *** [do-build] B??d 2
make[1]: Opuszczenie katalogu `/home/klugier/upp_temp/upp-x11-src-8782'
make: *** [umk] B??d 2
```

#13 - 09/15/2015 08:23 PM - Jan Dolinár

Zbigniew Rebacz wrote:

> *I type "make umk" and i have following error:*
> *[...]*

Oh, good catch. Your default shell is dash, right? This is a unintended bashism in the universal makefile. I'll fix it soon. For now, you can work it around just by telling make to use bash:

    make umk SHELL=bash

#14 - 09/15/2015 10:45 PM - Zbigniew Rebacz

For me it works fine, but i have got comments. It is good design to have -j $(CPU_CORES) by default? I would like to notice that make by default uses one job. For the other hand it is impossible to set jobs in your script:

    make -j 2 umk SHELL=bash

#15 - 09/15/2015 10:45 PM - Zbigniew Rebacz

BTW, coloured output rocks.

#16 - 09/16/2015 06:36 AM - Jan Dolinár

Zbigniew Rebacz wrote:

> *For me it works fine, but i have got comments. It is good design to have -j $(CPU_CORES) by default? I would like to notice that make by default*
> *uses one job. For the other hand it is impossible to set jobs in your script:*
> *   make -j 2 umk SHELL=bash*

Well, in my opinion, it is a good thing to have it on by default. I work with make a lot and it is actually quite tiresome to always havo to type this extra '-j' parameter. Also, many users don't know that the switch exists, they just type in "make <something>" as seen in the documentation and then they would be annoyed by how long it takes, even though the have 8 core machine :-)

The number of processes can be modified by passing JOBS=2 to the make, but I understand that it is nonstandard behavior and not the nicest solution. I remember fighting with this a lot, back when I originally wrote the universal makefile, but at that time, I could not find any reasonable solution... Guess I might give it another try.

Zbigniew Rebacz wrote:

> *BTW, coloured output rocks.*

Thanks, I love it too. It makes the output much more readable.

#17 - 10/26/2015 02:21 PM - Miroslav Fidler

- Status changed from New to Approved

Tempting, but in the end I have decided to just add another Makefile to the tarball...