

---

Subject: Re: 2020.1 officially released

Posted by [mirek](#) on Wed, 03 Jun 2020 10:46:19 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Sender Ghost wrote on Wed, 03 June 2020 07:37mirek wrote on Tue, 02 June 2020 19:27

Well, we already can add flags via "Common fixed flags". I think all we need is "override" option that would avoid adding platform flags before adding these fixed.

As far as I know, there is a possibility to override platform related flags. This feature is implemented in LocalHost::AddFlags and LocalHost::HasPlatformFlag methods [1, 2, 3].

For example, possible to generate Makefile(s) for WIN32 platform, e.g.:

```
./umk uppsrc ide CLANG -r -M=Makefile +WIN32,GUI
```

or FreeBSD:

```
./umk uppsrc ide CLANG -rs -M=BSDMakefile +POSIX,BSD,FREEBSD,GUI
```

or Linux:

```
./umk uppsrc ide CLANG -rs -M=GNUMakefile +POSIX,LINUX,GUI
```

etc.

There is just a need to use correct build method for them. But I didn't test this for cross-platform/remote build(s), just for Makefile generation.

May suggest to add "OSX" flag to platformFlags also (based on order of appearance in LocalHost::AddFlags method). Patch attached.

Yes, but that ADDs flags. The problem is that there are default, platform defined flags too. We need a method to clear those...

Well, radical solution would be to remove them completely and move them strictly to build method. Less radical is to add an option "No default flags" to build method.

Second problem is that there are some `#ifdef PLATFORM_` conditional compilation flags in GCC builder. Those need to be replaced to actually react to current flag configuration.

Mirek

---