
Subject: Re: Building applications
Posted by [Xemuth](#) on Wed, 12 Aug 2020 01:49:40 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Normvcr,

To speak about UMK we must first speak about packages. The thing is, an U++ application is made of "packages"

From U++ documentation Packages are centric to U++. An executable application is built from a package. A package can also build into a dynamic link library. A package can be used by other packages. A package corresponds to a single directory with the directory name being the name of the package. The package directory contains the package definition file (a plain text file with a .upp extension), which always has the same name as the package directory. The package definition file contains a list of the source files that make up the package, plus information on what type of package it is, how it should be built and what other packages it uses. The source files for the package are normally located in the package directory and its subdirectories, but they may be in any desired location.

By example, most famous packages of Upp are:

- Core (which also include the plugin/z package) the core package is one of pillars of Upp framework, it provide a lot of interesting think
- CtrlLib (which also include CtrlCore(which also include ...)) the CtrlLib package will provide most of features you need to do real interesting GUI

Package can be configured to allow some option on compilation/linking depending on flags(by example depending on the OS or the build type)

UMK is a tool which will link a compilation toolchain (GCC / CLANG / MSVS) with all the U++ logic (packages, assemblies etc...)

so, when you will need to build an U++ application, instead having to specify to your compiler wich source code he should look for or wich librairy need to be link, Umk will read package information and will use the compiler you chose to build your application / dll ...

The way you link UMK to a compilation toolchain is by creating Build method which can be seen in TheIDE :

As you can see, in the build method editor window of TheIDE you can specify most of thing a compiler need.

Later when you will compile your package (which can contain some others packages), TheIDE will invoke UMK and use your current build method to build your app.

The current build method is here in TheIDE :

When you want to compile your package by command line (imagine you develop your application under windows then compile it on linux without any Graphical interface)

then you must call UMK (as you have seen in the doc) by specifying your build method, package, assemblies etc... ([https://www.ultimatepp.org/appSide\\$umk\\$en-us.html](https://www.ultimatepp.org/appSideumken-us.html))

I'm not sure I answered your question, but maybe with a more precise vision about UMK your problem will be resolve

Xemuth
