

Using Arm Compiler 5 in newer Keil μ Vision MDK – UPDATED FOR MDK6/UBL

By Colin Funnell

Introduction

Like with most software tools, Keil's μ Vision MDK software evolves over time to keep up with latest trends and technologies. To keep things lightweight not all legacy tools are included 'out of the box'.

As of μ Vision version 5.37, it no longer ships with Arm Compiler 5. This isn't an oversight, just a gradual progression to give a strong prompt that version 6 is the way forward. In previous versions both compiler versions 5 and 6 have been available as part of the installation. In some cases, projects and existing code still need compiler 5.

Calling the compiler version 5 and version 6 can be a little misleading. Even more so now that version 6 is being renamed "Arm Compiler for Embedded".

With the advent of MDK6, μ Vision IDE is still included but new User Based Licensing (UBL) requires another step to manage for legacy designs.

From compiler 5 to 6

They are two separate tools, each with their own architecture and version numbering. Version 6 is meant to be the successor to 5 but either are still valid to use. While the safety version is under long term maintenance no new features will be added. For new projects you should use the version 6 compiler.

The version 5 compiler (armcc) has been developed over many years and matured into a very reliable system. Version 6 (armclang) jumps to the LLVM compiler infrastructure to take immediate advantage of the work already done in that area and bring it to Arm devices. It has a number of differences, but the main advantage is moving to 64-bit on the compiling host.

A Functional Safety (FuSa) qualified version of both compilers are available but only version 6 is being carried forward.

Although silicon vendor packs are migrating to using the new Compiler version 6, there are still a number of examples and demo code that expect version 5. For the unexperienced, this can be a little daunting seeing a demonstration project which should run 'straight out of the box' instead fill your screen with errors and warnings.

For those who have their existing project but have a clean installation with the latest MDK, it's normally imperative to get back to a usable state with the previous compiler to show all is well. We can put compiler 5 back into our latest μ Vision environment with a little work.

User Based Licensing (UBL)

The new licensing model used by Arm, UBL, is the approach used in MDK6.

It allows a lot of flexibility and covers the main use cases from both node locked and floating licence schemes, which are being withdrawn. Older versions of μ Vision (V5.36 and before) are not aware of UBL. Newer versions are aware and the latest will be UBL only. Compiler 5 is certainly not aware of UBL and will require an old-style licence, even in the latest editions of μ Vision. Arm has catered for this by having a tool to convert your UBL into a legacy licence.

Creation of a legacy licence is performed with the command-line tool: **armlm**. It can be found in your usual Keil installation folder under **/UV4/armlm**. Going to this folder and running **armlm inspect** will show you your current licence. By running **armlm genlic**, this will create a legacy LIC code for Keil products on your machine. Copy the Keil MDK licence key ready for the next step. Note that the genlic command option is new and although you may have armlm already in a slightly older version of μ Vision, the latest is always recommended.

```
C:\Keil_v538a\UV4\armlm>armlm inspect
1 product in your local cache:

Keil MDK Professional
Product code: KEMDK-PRO0
Order Id: ██████████
License valid until: 2025-06-03
Local cache expires: 6 days and 12 hours
Activation code: ██████████ ██████████ ██████████ ██████████
```

Figure 1 - armlm inspect is available in 5.38 but not genlic.

How to install another Arm compiler in μ Vision

We must now download the compiler we wish to use. Although developer.arm.com is eager to point you towards the latest copy of Version 6, we have to go to the Arm Compiler 5 (ACOMP5) download area. A registered account will be needed if you haven't got one already.

At the time of writing, the download can be found at:

<https://developer.arm.com/downloads/view/ACOMP5>

Here we can find the different releases of version 5 (and 4 if you want to go that far back).

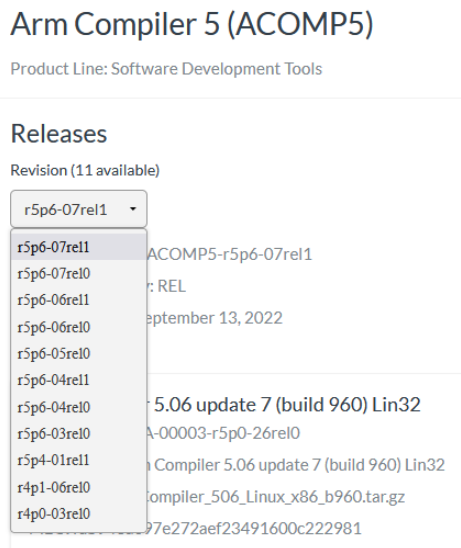


Figure 3- Compiler download releases

The download is a .tgz file which contains a gzipped .tar file. A tool which understands all these filetypes is going to be needed, such as 7-zip under Microsoft Windows.

The installation is straight-forward but I would recommend choosing a more suitable installation folder than the default Program Files (x86) area. For one it's probably better to keep it with where it would normally be, with your

{ μ Vision installation}\ARM\{useful name and version} folder. For me, I used C:\Keil_v537\ARM\ARM_Compiler_5.06u7 but it is traditionally just in the ... \ARM\ARMCC folder to be easily found by μ Vision. Secondly, it's always useful to avoid pathnames with whitespace. Even though Windows has been handling full filenames for many years now, in engineering it still poses problems with 3rd party tools. It wouldn't be the first time a useful tool gets written under *NIX, made to work under Windows, and unexpectedly fails on a particular machine with no useful explanation. Best to give these things a fighting chance so real development problems can be concentrated on instead.

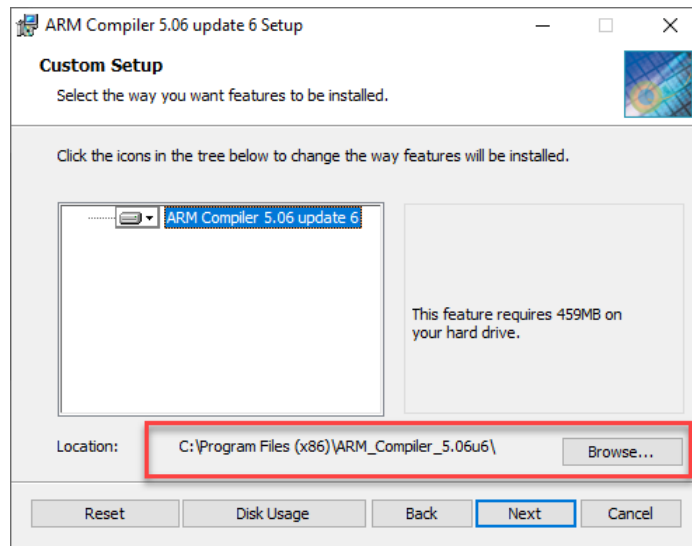



Figure 4 - Installation location

With the compiler installed it is now time for μ Vision to be informed of it. This is done through the menu under Project/Manage/Project Items..., or toolbar button . Here we can now specify where new compilers can be found.

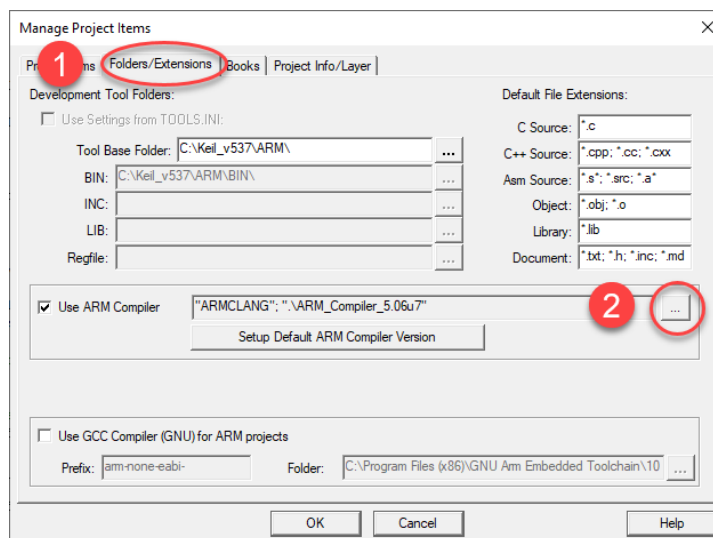


Figure 5 - Adding the new compiler

This brings up a list of known compilers which we can add to the list by selecting the same folder which was set during the compiler installation. If μ Vision finds it, it will now show up on the list. If not, you can give it a helping hand by clicking on the "Add another Arm Compiler Version to list..." button.

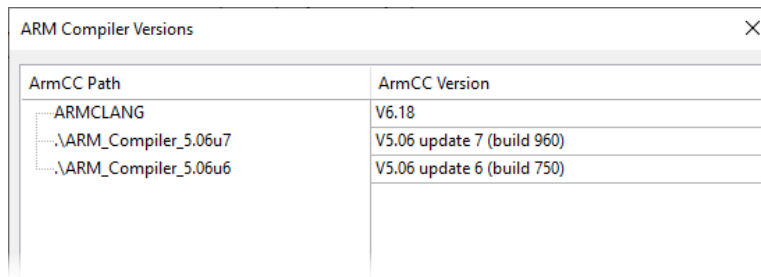



Figure 6 - Compiler list

When changing the “options for target” , you can now select your preferred compiler. This is now available for all projects.

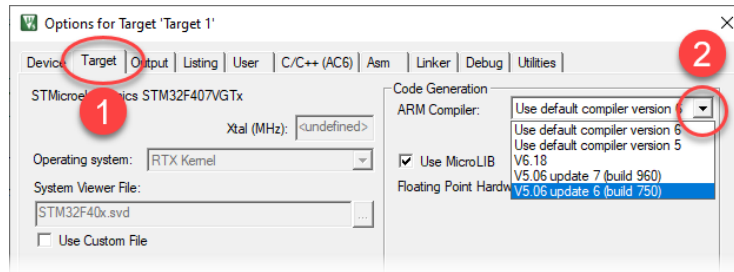


Figure 7 - Selecting the compiler to use

All sorts of different versions can be installed this way, but my main experience is just to re-introduce compiler 5 back into the system. The above is for the official Arm compiler. Note that it requires proper licensing to function. For a μ Vision installation already licensed there is nothing else needed to be done to use compiler 5.

It is possible for μ Vision to make use of the GCC Arm compiler. For this a slightly different approach is needed. Similarly, the compiler must be installed and μ Vision configured to use it. With the GCC case, configuration is done through the lower section of the screenshot in Figure 5.

For further reading see:

Migrate from ARM C/C++ Compiler 5 to ARM Compiler 6 - Keil Application Note 298:

<https://developer.arm.com/documentation/kan298/latest/>

Arm Compiler Version 6:

<https://www2.keil.com/mdk5/compiler/6/>

Keil MDK:

<https://hitex.co.uk/embedded-technology-and-services/development-tools/cortex-development-tools/keil-mdk-arm>

MDK-6 User-based License Activation:

https://www.hitex.co.uk/fileadmin/Knowledgebase/Hitex_Tech_Tip_MDK-V6_User-based_License_Activation.pdf

Licensing User's Guide:

<https://developer.arm.com/documentation/101454/0105/License-Management/User-based-License>

Generate Legacy Licenses from UBL video:

<https://developer.arm.com/Additional%20Resources/Video%20Tutorials/Keil%20Quick%20Tip%20Generate%20Legacy%20Licenses%20from%20UBL>

Manage Arm Compiler Versions:

<https://developer.arm.com/documentation/101407/0537/Creating-Applications/Tips-and-Tricks/Manage-Arm-Compiler-Versions>

Further Information

For more information visit our website: www.hitex.co.uk or get in touch: info@hitex.co.uk. You can also connect with us: [LinkedIn](#)

View our complete library of [Tech Tips](#).