Installation of a C/C++ Compiler for Altair Activate – September 2019

Altair Activate requires a C/C++ compiler to perform some operations (e.g. compile code for Modelica support).
The Activate documentation lists the supported **Visual Studio** compilers and tools supported.

Compiler	FMI Export	Modelica Support	C-Code Generation
Microsoft® Visual Studio 2019: Community, Professional	Yes	Yes	Yes
Microsoft® Visual Studio 2017: Express, Community, Professional	Yes	Yes	Yes
Microsoft® Visual Studio C++ Build Tools 2019; 2017; 2015 **Note:** The option for **Windows 10 SDK** must be selected.	Yes	Yes	Yes
Microsoft® Visual Studio C++ 2015 (VC 14.0 Community and Professional)	Yes	Yes	Yes
Microsoft® Visual Studio C++ 2013 (VC EXPRESS/VC Pro 12.0)	Yes	Yes	Yes
Microsoft® Visual Studio C++ 2012 (VC EXPRESS/VC Pro 11.0)	Yes	No	Yes
Microsoft® Visual Studio C++ VC 10.0 (VC Pro 10.0)	Yes	Yes	Yes
Microsoft® Visual Studio C++ SDK 7.1 (VC EXPRESS 10.0)	Yes	No	Yes
TCC (default provided with Activate installation)	No	No	Yes

In this document, we describe the steps to download and install "**VS C++ Build Tools 2017**" (second line of the table above).
Build Tools is one of the supported compiler and is probably simpler to use because it provides the required compiler pieces, without adding the Visual Studio IDE.

To install other compilers proposed in the table above, the steps are very similar but can slightly differ for each versions.
The key thing is to make sure the C/C++ compiler and the SDK are checked on in the installer options.

Important Remark

The table above proposes different versions of Microsoft Visual Studio.
Make sure the version you install is approved by your IT department and you have the right license (e.g. if you decide to use Professional Edition, a license is required

Step 0 - Check if a compiler is available and detected by Activate

In the command window of Activate, type this command **vssGetCompilerName**
If you get **tcc** this means no *installed* compiler was found (TCC is the default compiler coming with Activate, sufficient for some compilation tasks, but not appropriate for Modelica or FMU for example).

Step 1 - Go to the Microsoft Visual Studio download page

Link used in this document : https://visualstudio.microsoft.com/fr/visual-cpp-build-tools/

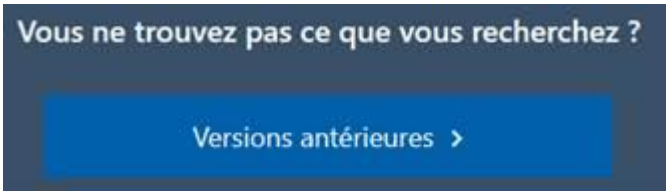
[Warning] The exact link may evolve over time, and may depend on your actual country/location

Step 2 – Select to download the VS C++ Build Tools

Step 3 – Select older versions

The Microsoft download page will offer by default to download the latest version.
For example, at the time of edition of this document, VS 2019 is proposed.

We want to download and install VS C++ Build Tools 2017. So, we go to the bottom of the page, looking for "**Download Older versions**"

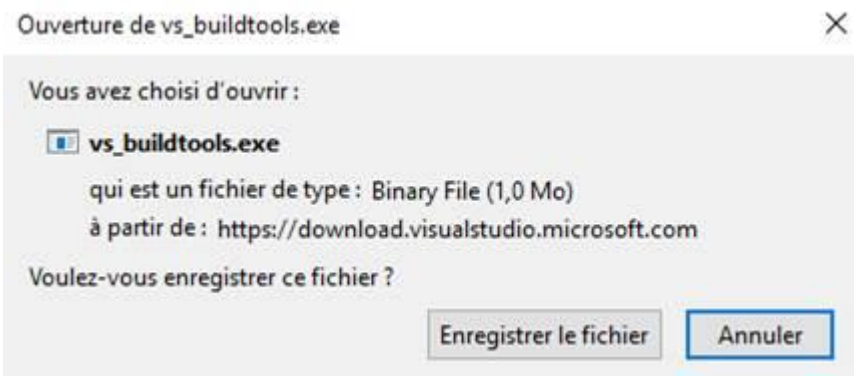


Step 4 – Download VS C++ BT 2017

To reach this page, it is required (this may be optional in some cases or some locations) to connect with a **Microsoft account**.

Step 5 – Start installation

Invoke the executable downloaded at Step 4.



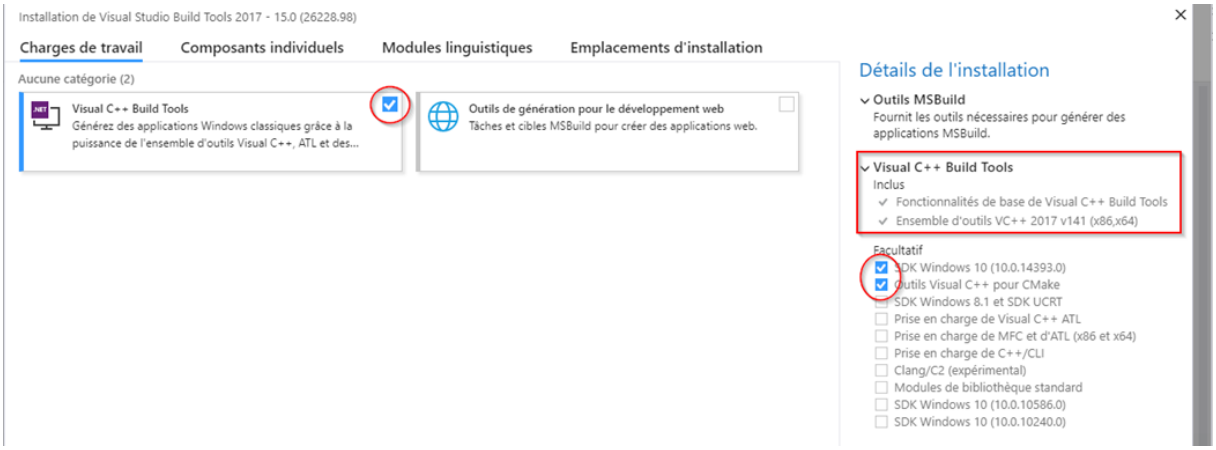
Step 6 – Select options

This is an important step.

Whether you download VS C++ Build Tools or you decide to download another Visual Studio (version listed as supported by Activate), it is important to make sure the **C/C++ compiler is actually selected** in the options.

In the case of **Build Tools**, the default options are OK and shown below. In particular, it is important to check **SDK Windows 10**.

In the case of **Visual Studio**, the list of options may differ in how they are presented, but it is key to make sure the C/C++ compiler is selected



Step 7 - Check if the compiler installation was succesful

In the command window of Activate, type this command : **vssGetCompilerName**

If you get **vc15_buildtools** this means the installation of VS C++ Buildtools 2017 was successful. Running an Activate model using Modelica blocks will confirm the installation is complete and successful.

In case you have installed another (supported version) Visual Studio, you may get a different answer. For example **vc14_x64** for Visual Studio 2015 with C/C++ compiler option enabled.