# Building and running pCEP in Visual Studio

This has been developed and tested on the following version of Visual Studio:

Microsoft Visual Studio Community 2017

Version 15.7.1

VisualStudio.15.Release/15.7.1+27703.2000

Microsoft .NET Framework

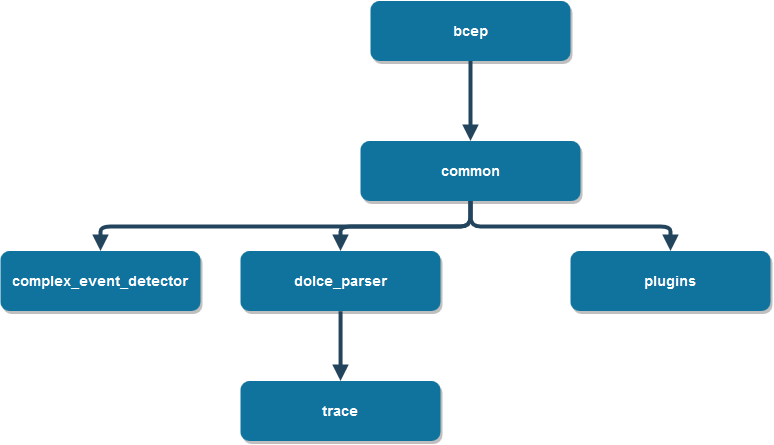
Version 4.7.03062

The community edition is free (formerly Visual Studio Express) and can be downloaded from [here](https://visualstudio.microsoft.com/es/downloads/?rr=https%3A%2F%2Fwww.google.com%2F).

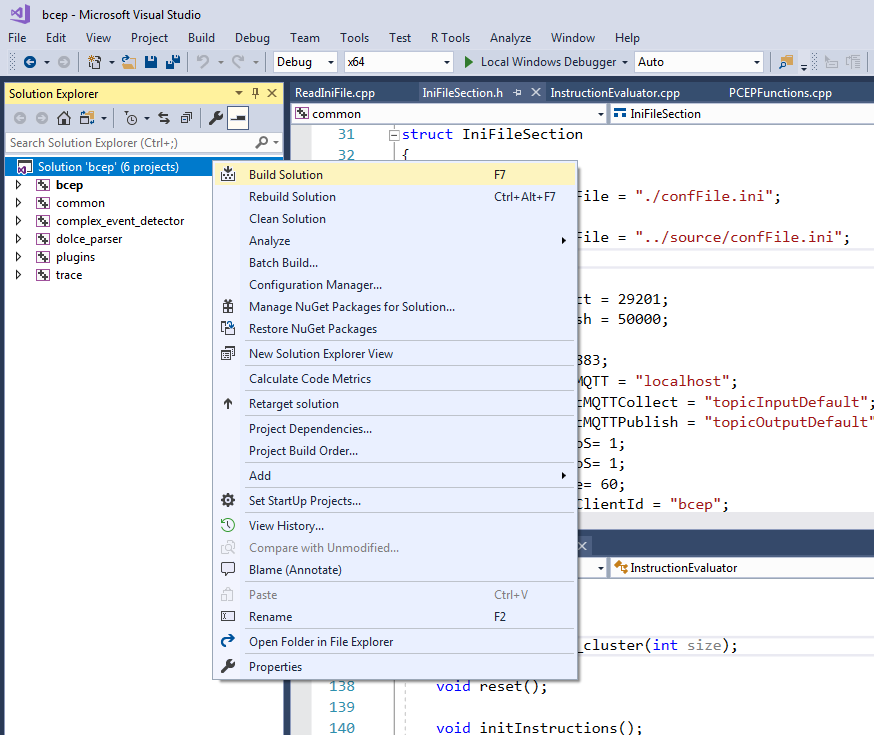
Visual Studio’s equivalent of Unix/Linux Makefiles are solutions and projects. The **bcep** solution (…/bcep/bcep.sln) is made up of 6 projects:

* **bcep**: …/bcep/bcep.vcxproj
* **common**: …/common/common.vcxproj
* **complex\_event\_detector**: …/bcep/complex\_event\_detector /complex\_event\_detector.vcxproj
* **dolce\_parser**: …/dolce\_parser/dolce\_parser.vcxproj
* **trace**: …/bcep/trace/trace.vcxproj
* **plugins**: …/bcep/plugins/plugins.vcxproj

When loading the bcep.sln file, the 6 projects with their dependencies are loaded. In the figure below, the projects and their dependencies are depicted. Note that bcep, the top most box, is the bcep project and not the solution. The whole diagram is the bcep solution.



To build the whole solution, right click on the solution and select Build Solution or simply press F7:



To build individual projects, right click on a project and hit Build.

## Running bcep

To run bcep, click on the green play icon Local Window Debugger, or simply press F5, however, to correctly execute bcep, it must be passed a dolce file. To set command line arguments, right-click on the project **bcep**, select Properties 🡪 Debugging and in the Command Arguments field type in your command line arguments, e.g. :

-d "C:\Users\a696457\dev\gitlab\bcep\source\test\detect1.dolce"

Or

-d "C:\Users\a696457\dev\gitlab\bcep\source\test\detect1.dolce" > "C:\Users\a696457\dev\gitlab\branches\bcep\source\test\out.txt"

If you want to send output to the …\out.txt file.

## Testing bcep

To test pCEP, build the Test configuration. Similarly to “make test” for the Linux version, this configuration will call the (windows specific) python test file (…\bcep\source\test\regtest\_win.py) with the appropriate arguments.

## Debugging bcep

My favorite way of debugging is running the Debug configuration of bcep from within VS, (F5), and sending UDP messages using the python script …\bcep\source\test\send\_udp\_msgs.py. If you set a breakpoint (F9) in the code which receives the messages, you should be able step through the code (F10) step into (F11)… etc., and of course inspect the variables, the call stack… etc. The best thing about Visual Studio in my opinion, is its debugging functionalities and the only reason for using Windows (again… in my opinion) is because of Visual Studio.