

Purpose

The purpose of this guide is to help a user run LANDIS-II in a debug mode where the program's execution can be traced step by step through the code.

Instructions

Due to the plug-in structure of LANDIS-II, running it through the Visual Studio debugger is not as simple as pressing "Start" in Visual Studio. Doing so requires some technical knowledge of how Visual Studio traces a program's execution.

The first step is to place a line of code near the starting point of the extension to be tested that will cause the program to pause its execution allowing you to attach the debugger. In most cases putting the code `Console.ReadLine();` in `PlugIn.cs` at the beginning of the `Initialize` method in whatever extension being tested will suffice. This code will cause the program to pause and wait for input from the user.

Once this is done, be sure that Visual Studio is configured to build a debug version of the code. To verify this, go to `Build -> Configuration Manager -> Select Debug` from the drop down menu under Configuration for whatever project being tested. This is important because Visual Studio will not build the files necessary for debugging if the project is built in release mode. Build the project from `Build->Build Solution`.

Once this finishes, the output will be placed in `\src\bin\Debug\netstandard2.0` or similar. Within this folder, two files are needed: the `dll` and `pdb` of the code. Look for the files with the name of the project you built followed by `.dll` or `.pdb`. Copy these two files.

These files will need to be placed in the actual LANDIS-II installation location on your computer. The path will be similar to if not exactly `C:\Program Files\LANDIS-II-v7\extensions`. This location is where all installed extensions are ultimately put to be used by LANDIS-II. Paste the two copied files in this location.

Next, run a LANDIS-II scenario as normal, but be sure that the extension you wish to test is part of this scenario. Once it starts, it will eventually pause when it reaches the code that was placed near the initialize method. When this happens, open Visual Studio. Go to `Debug -> Attach to process`. Look through the list for a process called `"dotnet.exe"`. This is a very common name for .NET applications like LANDIS-II but it may vary. Select this and click `"attach"`. It may take Visual Studio a few moments to complete this process. Place a breakpoint in the code where you would like to begin tracing the program's execution by clicking to the left of the code in Visual Studio.

After a short time, return to the LANDIS-II console window which has been paused up to this point. Press any key and then hit enter to bypass the `Console.ReadLine()` code and continue LANDIS-II execution. At this point the program will hit the breakpoint that was placed in Visual Studio and you are now able to trace the program in debug mode.

A common sticking point for users of Visual Studio is that after opening the solution of a given project, Visual Studio has everything it needs to debug that project in any application. This is not so. Visual Studio uses .pdb files for this purpose along with its corresponding .dll. So in fact, you can debug any number of LANDIS-II extensions in the same Visual Studio instance as long as you put each extension's .dll and .pdb file in the C:\Program Files\LANDIS-II-v7\extensions directory.