Application Note: UserFunction Interface

Overview

SIMetrix version 5 and later provide the ability for users to develop their own binary executable script functions and link them to the main program. The functions must be compiled to a dynamic link.This document describes how to do this. Most of the detailed documentation can be found in the interface header file and a simple supplied example.

User Function SDK

The SDK (software development kit) actually consists of just one header file namely UserFunctionInterface.h. This defines the interface in total.

User programs should be written in C++ and include this header file.

In addition there is a simple example - example.cpp - which demonstrates some trivial functions.

Documentation for the SDK is given within the UserFunctionInterface.h file and there is also additional help in example.cpp

Compiling the DLL

You must compile your code to produce a DLL. If you are a seasoned programmer, you won't need any advise from us in how to compile and link your program using the tools you are familiar with.

If you do not have C/C++ programming tools available, you can use the open-source gcc with Mingw extensions to compile Windows DLLs. This is the basic procedure:

1. Download and install GCC with Mingw. Note that if you are using the 64 bit version of SIMetrix you must install the 64 bit version of GCC – that is the version that builds 64 bit executable files. If you are using a 32 bit version of SIMetrix you must use the 32 bit version of GCC
2. Compile and link your .cpp file using this command:

   ```
   gcc –o mydll.dll –shared sourcefile.cpp
   ```
3. Copy the mydll.dll file to the folder plugins\functions under the SIMetrix root
4. If SIMetrix is running, close it now. The DLL will be automatically loaded on restart.
5. Your function or functions should now be installed. These will be accessible in the same way as any other script function.

Example

An example file is supplied – example.cpp. The simplest function simply multiplies a number by 2. To try the example, compile, link and install example.cpp as explained above then run this command:

```
Show Times2(2)
```

The response should be:

```
Times2(2) = 4
```

We recommend using the example as a starting point to develop your own functions.

Development

When you recompile a DLL to make changes, it isn't necessary to restart SIMetrix provided the following conditions are met:

1. You specify the macro
   REGISTER_INTERFACE_DYNAMIC
   in your source file
2. SIMetrix has previously loaded your dll and any functions in it in the current session.

If you create a new function in your DLL that has not been loaded at least once in the current session then you will need to restart SIMetrix so that it can register it.