

# DLL - TIPS

[http://www.tutorialspoint.com/dll/dll\\_tips.htm](http://www.tutorialspoint.com/dll/dll_tips.htm)

Copyright © tutorialspoint.com

Keep the following tips in mind while writing a DLL –

- Use proper calling convention *Corstdcall*.
- Be aware of the correct order of arguments passed to the function.
- NEVER resize arrays or concatenate strings using the arguments passed directly to a function. Remember, the parameters you pass are LabVIEW data. Changing array or string sizes may result in a crash by overwriting other data stored in LabVIEW memory. You MAY resize arrays or concatenate strings if you pass a LabVIEW Array Handle or LabVIEW String Handle and are using the Visual C++ compiler or Symantec compiler to compile your DLL.
- While passing strings to a function, select the correct type of string to pass. C or Pascal or LabVIEW string Handle.
- Pascal strings are limited to 255 characters in length.
- C strings are NULL terminated. If your DLL function returns numeric data in a binary string format *forexample, viaGPIBortheserialport*, it may return NULL values as a part of the data string. In such cases, passing arrays of short 8 – bit integers is most reliable.
- If you are working with arrays or strings of data, ALWAYS pass a buffer or array that is large enough to hold any results placed in the buffer by the function unless you are passing them as LabVIEW handles, in which case you can resize them using CIN functions under Visual C++ or Symantec compiler.
- List DLL functions in the EXPORTS section of the module definition file if you are using `_stdcall`.
- List DLL functions that other applications call in the module definition file EXPORTS section or to include the `_declspec dllexport` keyword in the function declaration.
- If you use a C++ compiler, export functions with the `extern "C"{}`  statement in your header file in order to prevent name mangling.
- If you are writing your own DLL, you should not recompile a DLL while the DLL is loaded into the memory by another application. Before recompiling a DLL, ensure that all applications using that particular DLL are unloaded from the memory. It ensures that the DLL itself is not loaded into the memory. You may fail to rebuild correctly if you forget this and your compiler does not warn you.
- Test your DLLs with another program to ensure that the function *andtheDLL* behave correctly. Testing it with the debugger of your compiler or a simple C program in which you can call a function in a DLL will help you identify whether possible difficulties are inherent to the DLL or LabVIEW related.

Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js