**Accessing Functions from a C/C++ dll in SAS**

Ariah Klages-Mundt

1. Create a C or C++ dll with exported functions. Code files for an example dll is included in this directory (dlltutorialcpp.cpp, dlltutorialcpp.h, and dlltutorialcpp.def). The functions that are available externally are those stated in the .def file and which include \_\_declspec(dllexport) before the function declaration. Build dlltutorialcpp.dll from these source files.
2. Add the dll directory to the system path environment variable before opening SAS.
3. Put together a SASCBTBL attribute table according to the [documentation](http://support.sas.com/documentation/cdl/en/hostwin/63285/HTML/default/viewer.htm#sascbtbl.htm) for each function you want to access in SAS. Use [this](http://support.sas.com/documentation/cdl/en/hosto390/67326/HTML/default/viewer.htm#p0a727x7johamnn1gmrcg467v6ep.htm) as a guide for specifying argument formats. You can save all of these tables to one .dat file. A dlltutexcpp.dat that contains SASCBTBL attribute tables for the functions in dlltutorialcpp.dll is included in this directory. Notice that function Add1 uses a structure input whereas Add uses two integer inputs. This is dealt with in the SASCBTBL table.
4. In SAS, use a FILENAME statement to assign the SASCBTBL fileref to the .dat file you created. For example:

filename sascbtbl "c:\dlltutexcpp.dat";

1. In SAS, within a DATA step, use a MODULE command to call a dll function according to the [documentation](http://support.sas.com/documentation/cdl/en/hostwin/63285/HTML/default/viewer.htm#win-func-module.htm). For example,

**data** \_null\_;

p=**0**;

call module( '\*i',

"dlltutorialcpp,Add", **1**, **2**, p);

put p=;

**run**;

returns p=3,

**data** \_null\_;

n=modulen( '\*i',

"dlltutorialcpp,Add1", **1**, **2**);

put n=;

**run**;

returns n=3,

**data** \_null\_;

x = **1**;

y = **1.5**;

z = 'ab';

call module( '\*i',

"dlltutorialcpp,Func", x, y, z);

put x= y= z=;

**run**;

returns x=2 y=3.5 z=ba, and

filename sascbtbl "c:\dlltutexcpp.dat";

**data** \_null\_;

length str $**21**;

call module( '\*i',

"dlltutorialcpp,GetString", str);

put str=;

**run**;

returns str=qwertyuiopasdfghjklz.