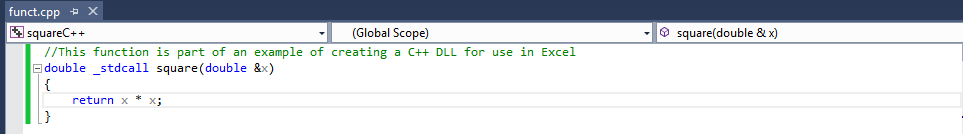
# **Ayshahvez Konowalchuk 1103538**

**APPENDIX**

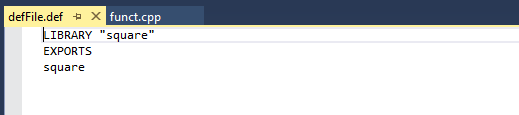
**Case #1: Language Interoperability demonstration**

The Technical Director requested demonstration of interoperability between environments of how *C/C++* languages would be linked to VB scripting in order to invoke C/C++ functions as macros in Microsoft Office Excel. Below is a demonstration of technicality of proving interoperability as was mention:

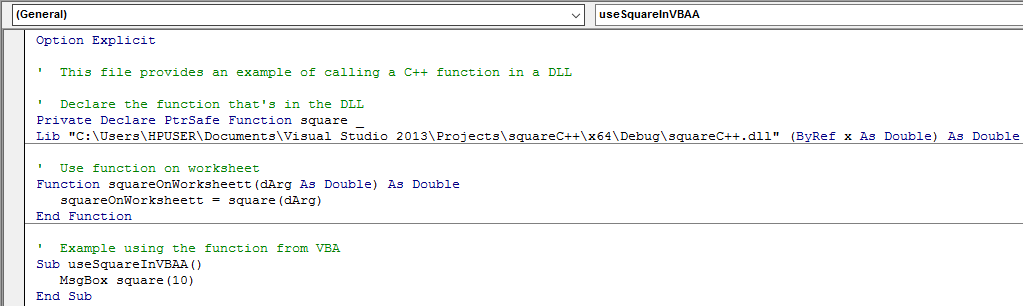
Here is simple C++ function which simply calculates square of any number. The simple demo will just demonstrate how the C++ function can be as a macro in Excel with help of VB scripting. **\_stdcall** is basically telling the compiler that this function will be used by another application which is Excel via VB scripting.



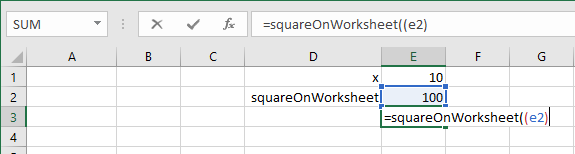
In order for the C++ function to be invoked by scripting features of VB scripting, a definition file was also to be created; we called this defile.def. LIBRARY “square” stipulates square is just the name of the library; EXPORTS mean to export function square to another application, and square is just the name of function.



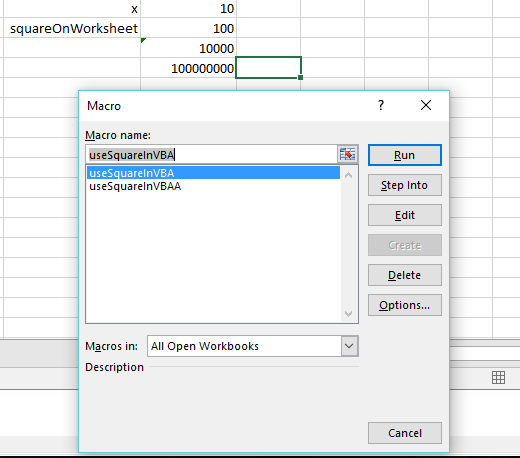
Below is the code written in VB Script to demonstrate how to include the C++ function DLL file which is the “squareC++.dll”. There is function which can be used in excel and also a macro which can be run anytime when necessary. "PtrSafe" basically tells VBA that the function is safe for 64-bit. The VBA function squareOnWorksheet takes a double as an argument, calls square function using this double as an argument, and then returns the value it gets back from square function. This way square function gets the double it is expecting and the worksheet gets the result it is expecting.



Now we can call the function which was written in C++ language in excel. As shown below, the squareOnWorksheet function was what we called it and we can use it to simply square any number as stipulated



Now you can also call the macro as was written in VBA for the C/ C++ function. The macro useSquareInVBA was written in C and useSquareInVBAA was written in C++. Both executes fast as both languages are compiled languages.

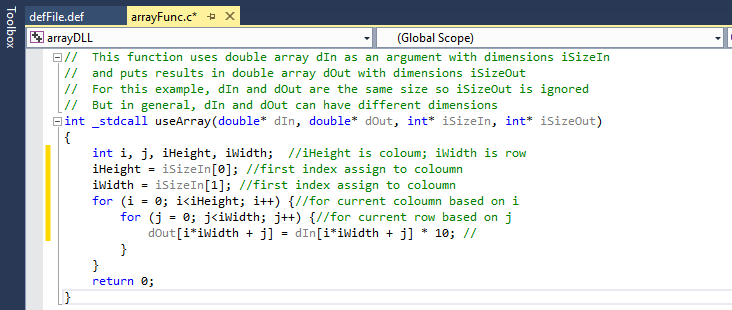


**Case #2: Language Interoperability demonstration**

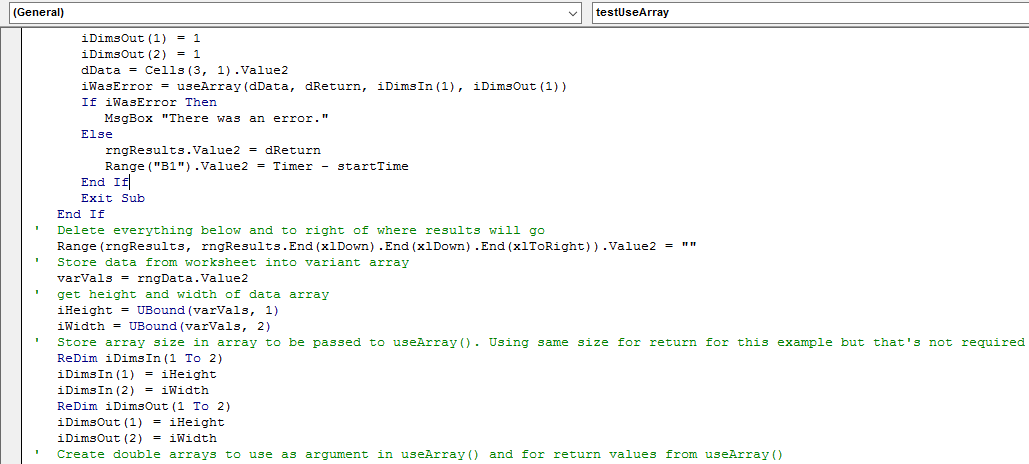
Below is a function (useArray) which uses array data structure as well as pointers and loops.

The C function will be exported to the VB scripting environment where a sub-routine will utilize the fast run-time execution speed of the said function in order to carry out calculation in Excel.

To demonstrate the fast run-time execution time of this function in the C compiled environment, a routine in VB script call upon the function for its fast processing power.

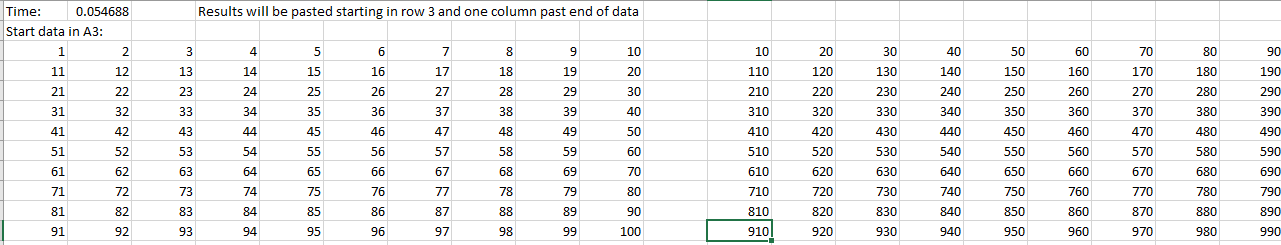


Here below is a snippet of the routine written in VB script code, where the arguments were passed to the useArray function which provides the fast run-time environment for the operation to be completed in 0.054688s.



The operation was to loop through the list of the numbers on the left starting from cell A3, calls the C function and apply a multiplication of 10 to the list and then puts the returned results (the original values multiplied by 10) in the cells to the right of the original values. This operation was performed in run-time of 0.054688s in Microsoft Excel. It varies between computers but it normally takes less than 0.1s.

This proves the fast processing power of C language compiled environment which provided the function to which excel utilized as a macro via VB scripting environment.



References

Retrieved January 25, 2016 from [*https://sites.google.com/site/jrlhost/links/excelcdll*](https://sites.google.com/site/jrlhost/links/excelcdll)

Retrieved January 25, 2016 [*http://www.tutorialspoint.com/apache\_poi*](http://www.tutorialspoint.com/apache_poi)

Retrieved January 25, 2016 *http://www.libxl.com/read-excel-data.html*