

Use of C++ Dlls in Visual Basic for Applications With Excel 2000

Edward M. Rosen
EMR Technology Group

Background

Microsoft's Excel 2000 is a widely used spreadsheet program. Its macro language, Visual Basic for Applications (VBA) (1,2) , provides an attractive means for performing a variety of engineering computations.

Rosen and Partin (3) have described how legacy chemical engineering technology developed in FORTRAN can be utilized by calling the FORTRAN routines from VBA. In addition to FORTRAN, VBA can also directly utilize functions and/or subroutines written in the popular C++ programming language (4,5).

Visual C++ 6.0 offers a wizard to enable rapid development of C++ DLLs (dynamic link libraries). However, despite the coaching the wizard supplies and the many Internet help sites, the development of the DLL can be confusing. In this article the author describes the use of Visual C++ 6.0 to generate a DLL which can be called directly from the VBA code. An example Excel spreadsheet which utilizes the DLL is described.

Building the Dll for Use in VBA

Here are the steps to build a DLL (Visual C++ 6.0):

Start Visual C++

Create a new project by choosing New from the File menu.

In the window that appears select the following:

A project name (here we will choose ddd).

A project location (here we choose C:\Ctest)

Win32 Dynamic-Link Library

In the next window choose

A simple DLL project

As a result the following files are created:

ddd.cpp: The C++ file in which your coding will be placed

StdAfx.h : Precompiled header file

StdAfx.cpp: Source File

There is no need to modify the two StdAfx files.

Figure 1a shows ddd.cpp file with the coding for the ddd.dll that we are creating. The line starting with double `_stdcall Vari (double indata[], long NumItems)` defines a C++ routine named “Vari” to calculate the variance of a list which is NumItems long.

Figure 1b is an additional file file (ddd.def) that must be inserted into the project. It allows for the export of the “Vari” function.

The project is then compiled using the “Rebuild All” command.

After the file ddd.dll is debugged and built (with Win32 Release) it must be placed where it can be found. Here we have placed it in “C:\Ctest\ddd\Release\ddd.dll”.

Calling the DLL from VBA

The VBA module (Rtest) is shown in Figure 2. Note that DECLARE statement points to the location of the dll library (ddd.dll) The array dd is passed by to the dll by reference and the first element of the array dd is indicated in the call to “Vari”.

Utilization on the Excel Spreadsheet

The spreadsheet can now invoke the Rtest function like any other spreadsheet function. Figure 3 illustrates its use to find the variance of three lists. The variance is first calculated using the spreadsheet function Var. Next the worksheet function Count is used to determine the length of the list. Then the Rtest function is invoked passing the location of the first item of the list and its length to Rtest. For example: `Rtest (B6,B19)`. Note that the “Vari” C++ routine gives the same answer as the spreadsheet routine VAR.

Conclusions

C++ dlls can be called directly from VBA code. However, care must be taken to build the dll properly for use in VBA.

References

1. Walkenbach, J. *Microsoft Excel 2000 Power Programming With VBA*, IDG Books Foster City, CA (1999)
2. Getz, Ken and Mike Gilbert, *VBA Developer's Handbook*, Sybex, San Francisco (1997).
3. Rosen, E. M. and L. R. Partin "A Perspective: The Use of the Spreadsheet for Chemical Engineering Computations", *I&EC Chemistry Research*, 39(6) 1612-1613 June 2000
4. Horton, Ivor, *Beginning Visual C++ 6*, WROX Press, Chicago (1998)
5. McMahon, David, *Rapid Application Development with Visual C++* McGraw-Hill, New York (2000)

```

// ddd.cpp : Defines the entry point for the DLL application.
//

#include "stdafx.h"

BOOL APIENTRY DllMain( HANDLE hModule,
                      DWORD ul_reason_for_call,
                      LPVOID lpReserved
                      )
{
    return TRUE;
}

double _stdcall Vari(double in_data[], long NumItems)
{
    //Function : Vari
    //Description : Returns the variance for array in_data

    long j;
    double sumx=0, sumsquared=0;
    double vari=0;

    for (j=0; j< NumItems; j++)
    {
        sumx += in_data[j];
        sumsquared += in_data[j]*in_data[j];
    }

    vari =(sumsquared - (sumx*sumx/(double)NumItems))/((double)NumItems -1.0);

    return(vari);
}

```

Figure 1a The ddd.cpp File (modified)

```

; DDD.def: Declares the module parameters for the DLL

LIBRARY "DDD"
DESCRIPTION 'DDD Windows Dynamic Link Library'
EXPORTS
    Vari @1

```

Figure 1b The added ddd.def file

```

Option Explicit
Public Declare Function Vari Lib "C:\Ctest\ddd\Release\ddd.dll" _
    (ByRef dd As Double, ByVal kdex As Long) As Double

Public Function Rtest(x, NumItems)

Dim dd() As Double
Dim var As Double
Dim kdex As Long
Dim I As Long

kdex = NumItems

ReDim dd(1 To kdex)

For I = 1 To kdex
dd(I) = x(I)
Next I

var = Vari(dd(1), kdex)

Rtest = var

End Function

```

Figure 2 The VBA Module Invoking “Vari”

Routine to Test C++ dll Vari in VBA

Input Data Values	List 1	List 2	List 3
	5	8	1345
	6	9	1301
	8	20	1368
	9	33	1322
	12	56	1310
	33		1370
	19		1318
	22		1350
			1303
			1299
Variance Calculated from Worksheet Function Var	94.21429	398.7	754.2667
Number of Items Using Worksheet Function Count	8	5	10
Variance Calculated from C++ Dll Vari	94.21429	398.7	754.2667

Figure 3 Excel Spreadsheet Utilizing the C++ DLL