

The purpose of this document is to describe one method through which one can interface a simple C++ program with Microsoft Excel. The end result allows for data to be sent between the program and an Excel spreadsheet. This method is especially helpful in that it allows for data generated in C++ to be displayed graphically without the use of complicated programming. The process shown uses the Microsoft Visual C++ 2005 compiler, as well as Microsoft Office Excel 2007. Note, the same can be achieved using newer versions of either software with minor changes.

For this example, a C++ program will generate the values of the function $Y=X^2$ for the domain $0 \leq X \leq 5$. An Excel spreadsheet will then be used to display these numbers graphically.

The entire process requires only three files:

- A C++ Program [.cpp]
- A data file [in this case, a .csv]
- An Excel spreadsheet [.xlsx]

For the sake of keeping things easy, it is best to keep these three files in the same folder.

The overall procedure is simple: A variable **x** is initialized with a value of 0. The C++ program squares this number, and then stores the new value in variable **y**. The program then immediately outputs the value of **y** to the data file and starts a new line. After the program writes the value to the data file, it then proceeds to increment **x**. The process then repeats until all values have been written to the data file. The Excel spreadsheet will be set up to take inputs from the data file in the form of cells, and a graph will be generated to display these inputs.

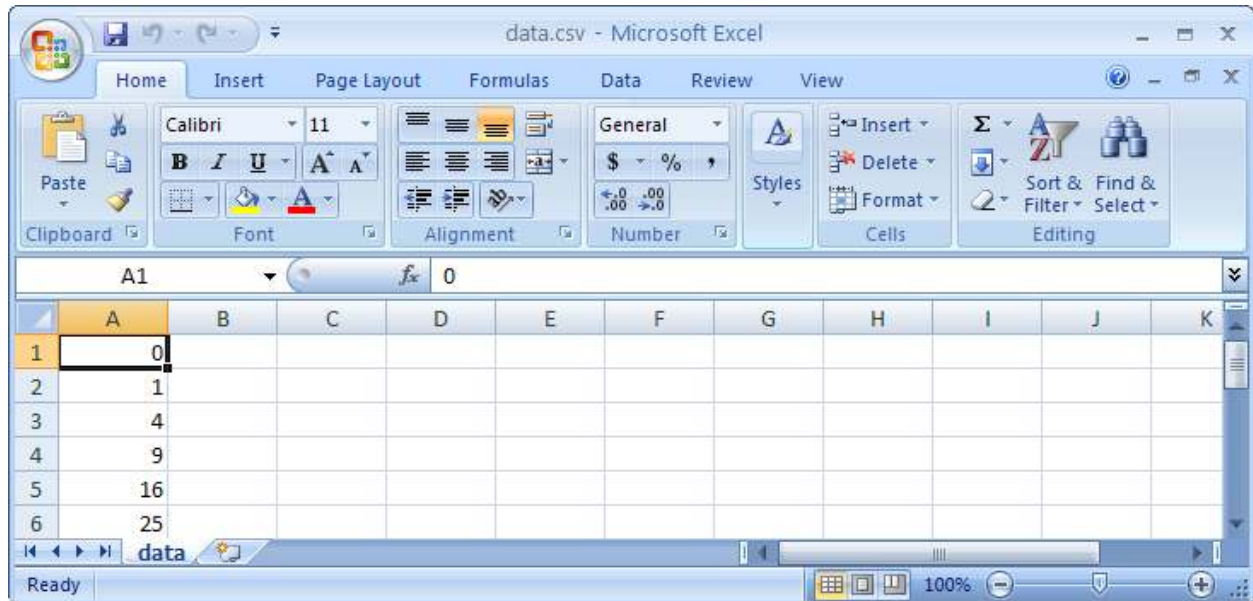
The Program

```
#include<fstream>           // fstream is required for data to be written to external files
#include<windows.h>
#include<math.h>             //math.h is necessary to use the power function, use <cmath> with 2008 compiler
using namespace::std;
void main()
{
    int y; // x [declared later] will be squared and stored here
    ofstream output("data.csv",ios::out); // Declaring ofstream handle "output." Using this handle
                                           // will write argument to the file "data.csv."

    for(int x=0; x<=5;++x)
    {
        y=pow(x,2); // raise x to the power of 2. Store value in variable y
        output << y; // write y to "data.csv"
        output << endl; // start a new line in "data.csv" following y
    }; // Go back to store additional values until all have been written.
    output.close(); // Close access to "data.csv." Allows other programs to access this file.
    system("Graph.xlsx"); //OPTIONAL. This command opens file "Graph.xlsx"
};
```

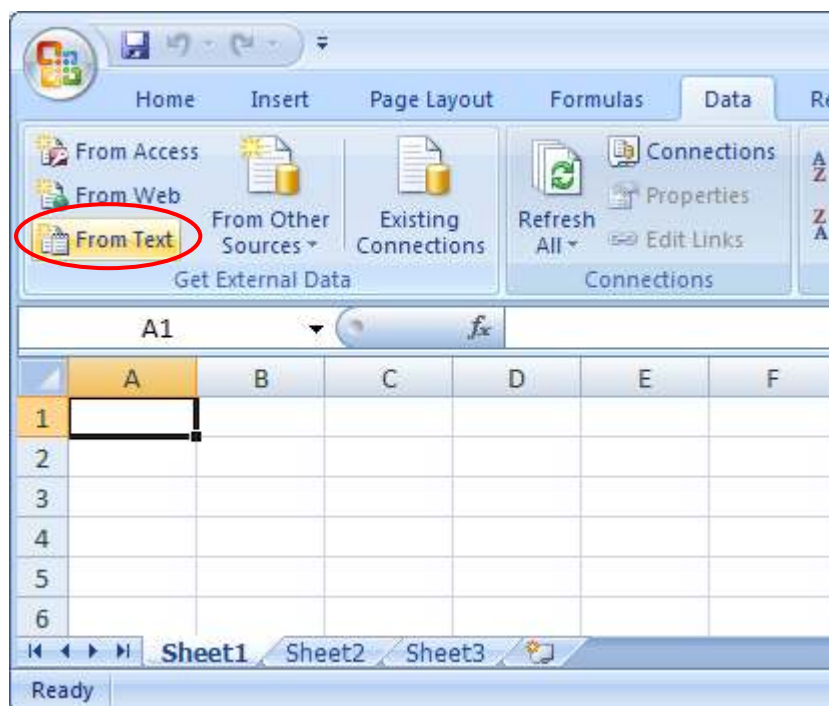
The Data File

Assuming the above program was run properly, the values 0, 1, 4, 9, 16, 25 will have been sent to the data file. The file should look like this:

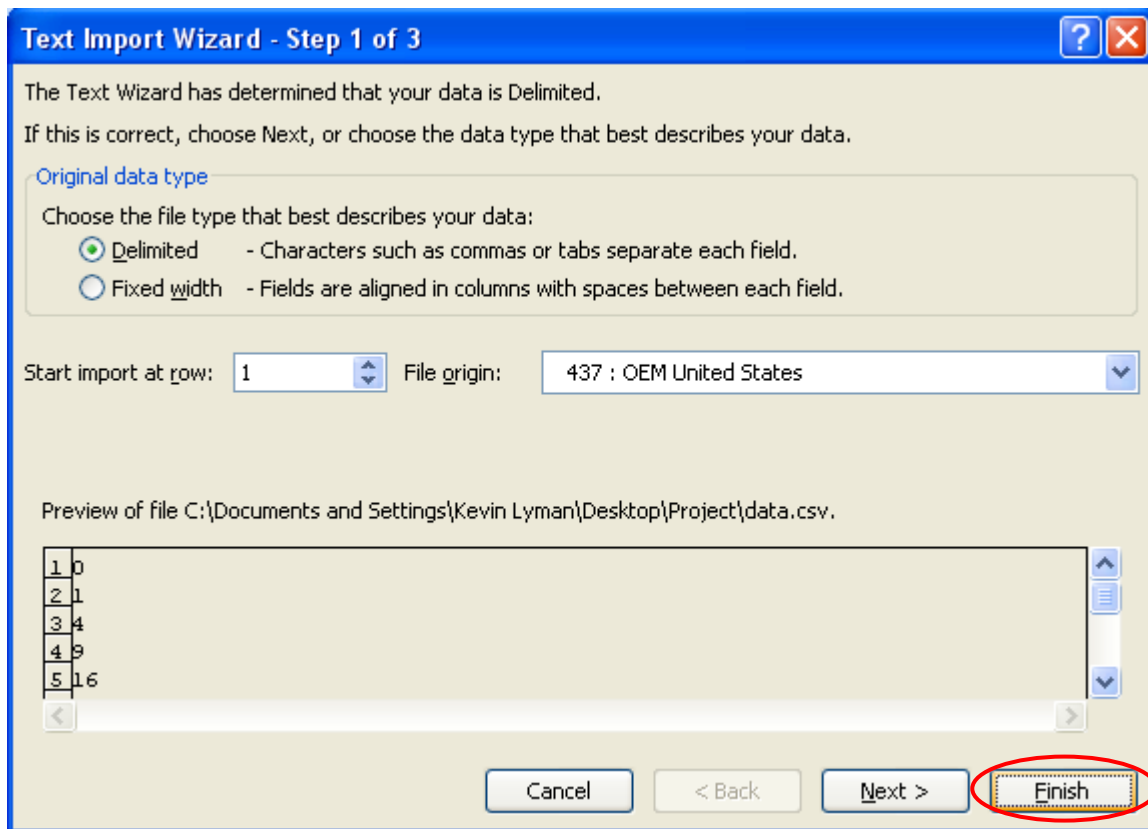


The Excel Spreadsheet

Create a new Excel spreadsheet. Select cell A1, go to the “Data” tab, and press “Get External Data >> From Text.” This will bring up a file select interface. Go to the folder which contains your data file, “Data.csv,” and select it.



You will then be prompted with the “Text Import Wizard.” Because this example uses new lines as row delimiters, just hit “Finish.” Hit “Ok” when the “Import Data” box appears. Your Excel spreadsheet should now look like the data file did previously.



The Text Wizard has determined that your data is Delimited.
If this is correct, choose Next, or choose the data type that best describes your data.

Original data type
Choose the file type that best describes your data:

- ☒ Delimited - Characters such as commas or tabs separate each field.
- ☐ Fixed width - Fields are aligned in columns with spaces between each field.

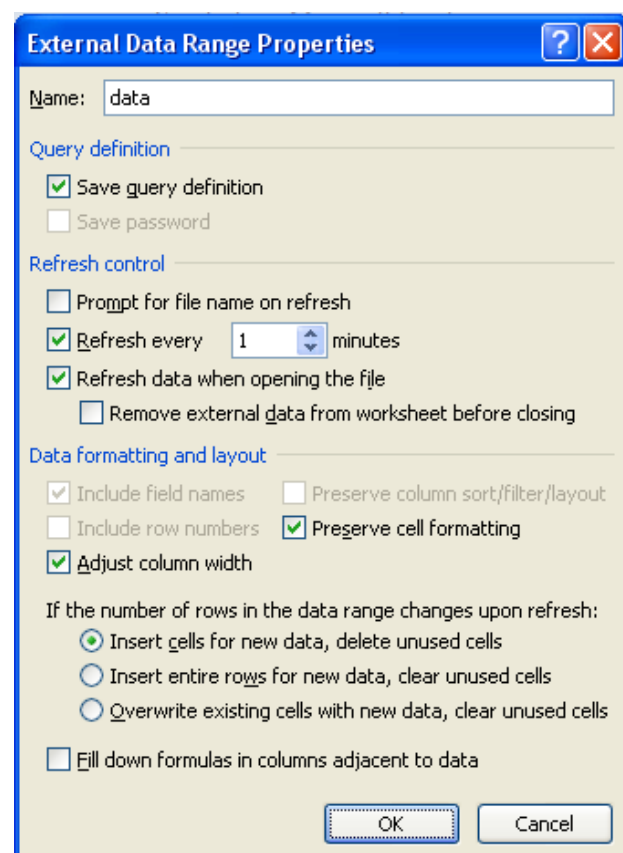
Start import at row: 1 File origin: 437 : OEM United States

Preview of file C:\Documents and Settings\Kevin Lyman\Desktop\Project\data.csv.

1	0
2	1
3	4
4	9
5	16

Buttons: Cancel, < Back, Next >, **Finish**

The next step is to ensure that the spreadsheet always has up-to-date data. Highlight all of the cells which contain input data [in this case, A1:A6]. Go to the “Data” tab again, and select “Properties.” Set the properties as they are displayed on the right. These settings ensure that the spreadsheet is updated every minute, and every time the spreadsheet is opened.



External Data Range Properties

Name: data

Query definition

- ☒ Save query definition
- ☐ Save password

Refresh control

- ☐ Prompt for file name on refresh
- ☒ Refresh every 1 minutes
- ☒ Refresh data when opening the file
- ☐ Remove external data from worksheet before closing

Data formatting and layout

- ☒ Include field names
- ☐ Preserve column sort/filter/layout
- ☐ Include row numbers
- ☒ Preserve cell formatting
- ☒ Adjust column width

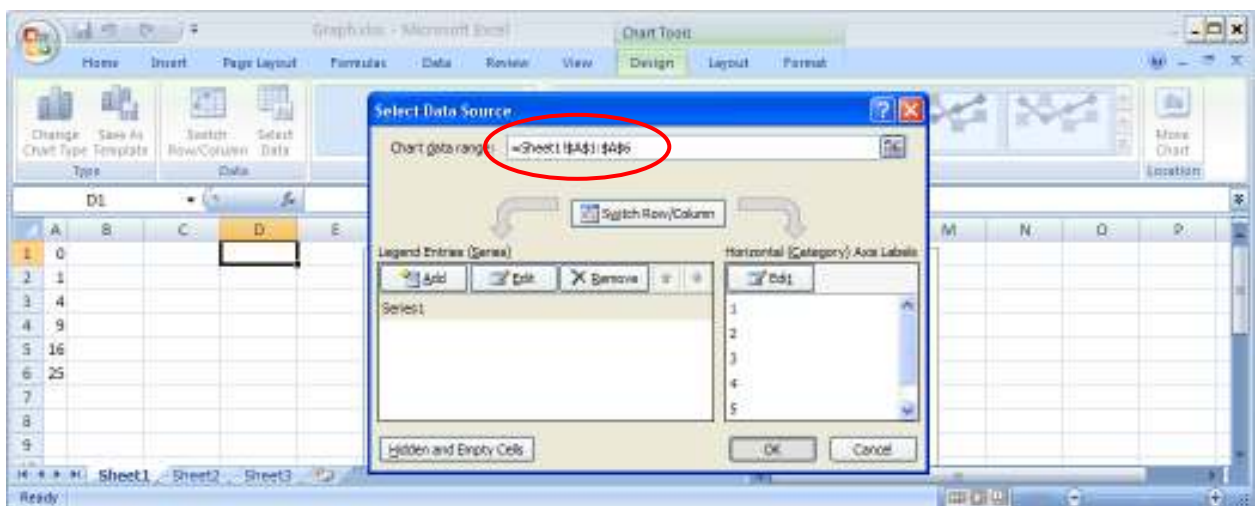
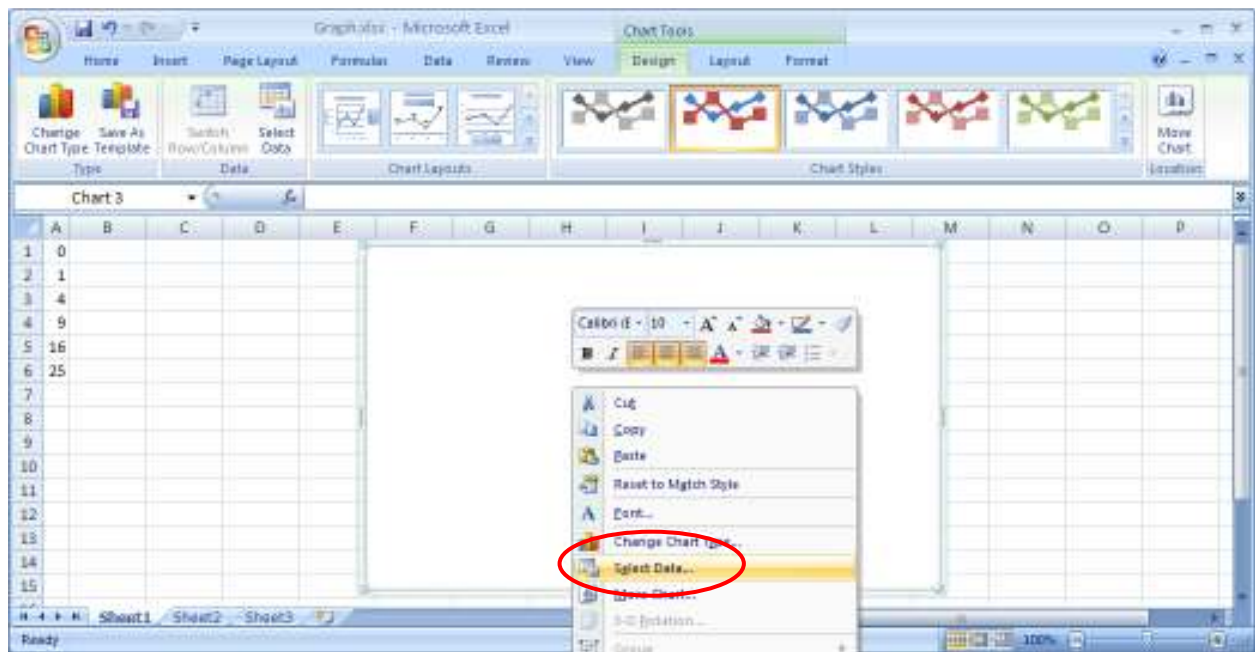
If the number of rows in the data range changes upon refresh:

- ☒ Insert cells for new data, delete unused cells
- ☐ Insert entire rows for new data, clear unused cells
- ☐ Overwrite existing cells with new data, clear unused cells

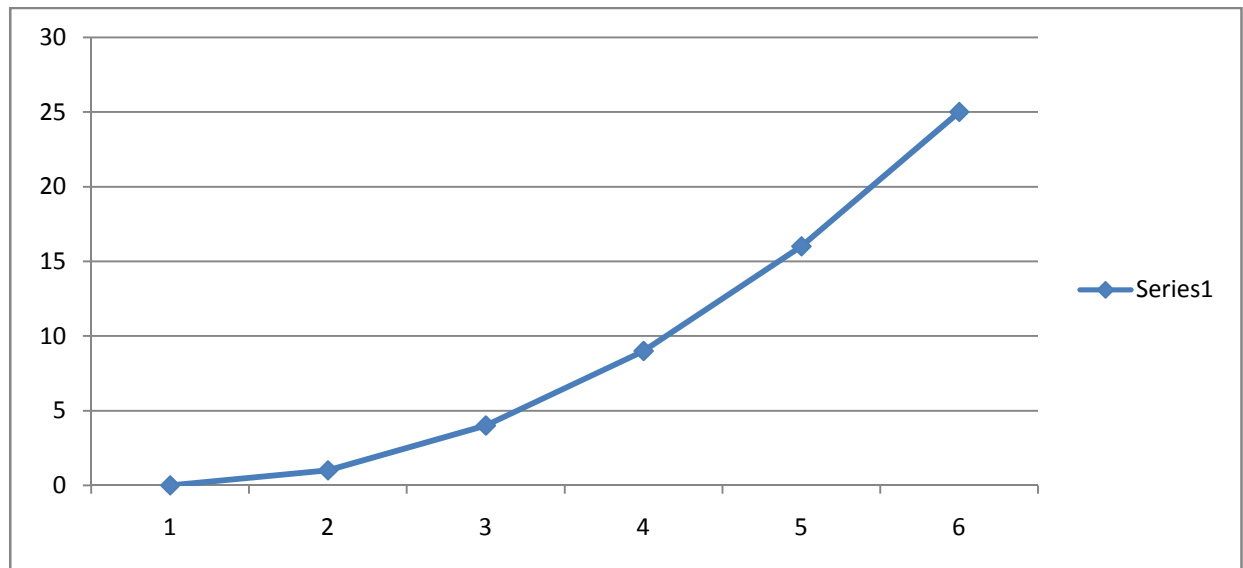
☐ Fill down formulas in columns adjacent to data

Buttons: OK, Cancel

Now that the data values are in the Excel sheet, they can be referenced for use by graphs. To do this, press “Insert >> 2-D >> Line Chart,” right click the empty chart, and select the imported data as your range.



After hitting “Ok,” the graph should be generated.



Ending Notes

- Multiple pieces of data can be output and stored in the data file at one time, so long as there is a delimiter of some sort.
- Remember, all file names can be changed, as long as there is consistency.
- Make sure the data file is not open when being written to.
- **Make sure Data Connectivity is always enabled in Excel. To do this, click the Office button in the upper left hand corner of Excel, click Excel Options at the very bottom. Go to Trust Center, then Trust Center Settings. Click External Content, and check off “Enable all data connections.”**