**Command Line Arguments in C++**

**Accepting command line arguments in C++ using argc and argv**

**Source:** [Alex Allain](https://plus.google.com/113987539774523532573?rel=author) <http://www.cprogramming.com>

In C++ it is possible to accept command line arguments. Command-line arguments are given after the name of a program in command-line operating systems like DOS or Linux, and are passed in to the program from the operating system. To use command line arguments in your program, you must first understand the full declaration of the main function, which previously has accepted no arguments. In fact, main can actually accept two arguments: one argument is number of command line arguments, and the other argument is a full list of all of the command line arguments.

The full declaration of main looks like this:

int main ( int argc, char \*argv[] )

The integer, argc is the ARGument Count (hence argc). It is the number of arguments passed into the program from the command line, including the name of the program.   
  
The array of character pointers is the listing of all the arguments. argv[0] is the name of the program, or an empty string if the name is not available. After that, every element number less than argc is a command line argument. You can use each argv element just like a string, or use argv as a two dimensional array. argv[argc] is a null pointer.   
  
How could this be used? Almost any program that wants its parameters to be set when it is executed would use this. One common use is to write a function that takes the name of a file and outputs the entire text of it onto the screen.

#include <fstream>

#include <iostream>

using namespace std;

int main ( int argc, char \*argv[] )

{

if ( argc != 2 ) // argc should be 2 for correct execution

// We print argv[0] assuming it is the program name

cout<<"usage: "<< argv[0] <<" <filename>\n";

else {

// We assume argv[1] is a filename to open

ifstream the\_file ( argv[1] );

// Always check to see if file opening succeeded

if ( !the\_file.is\_open() )

cout<<"Could not open file\n";

else {

char x;

// the\_file.get ( x ) returns false if the end of the file

// is reached or an error occurs

while ( the\_file.get ( x ) )

cout<< x;

}

// the\_file is closed implicitly here

}

}

This program is fairly simple. It incorporates the full version of main. Then it first checks to ensure the user added the second argument, theoretically a file name. The program then checks to see if the file is valid by trying to open it. This is a standard operation that is effective and easy. If the file is valid, it gets opened in the process.