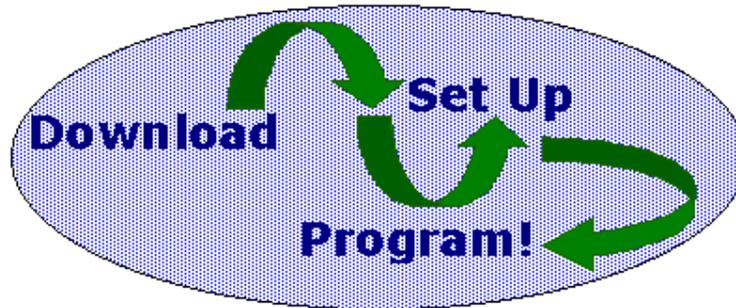


Welcome to

# Miracle

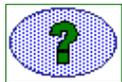
## C

A quick guide to  
*downloading,*  
*installing,* and  
*using* the Miracle C  
Compiler

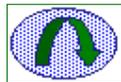


The **Miracle C** compiler runs under MS Windows (3.1, 95, 98, and NT) and compiles for MS DOS. It's easy to download and install, and the registered version is much less expensive than many other C compilers, making it an excellent choice for beginning as well as experienced programmers.

This guide shows you how to download, install, and use the basic functions of the **Miracle C** compiler. You don't need to have any prior experience with compilers, or with downloading or installing applications. As long as you know how to create, name, and move folders in Windows, you'll be able to start compiling your own C code within a few minutes.



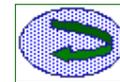
Compiler Basics



Downloading



Setting Up



Using

## *Compiler Basics*

If you're new to programming, you may not know what a compiler is and does. This section is for you.

### What is a compiler and what does it do?

A compiler is a program that turns C-language code into binary instructions that your computer can understand and execute.

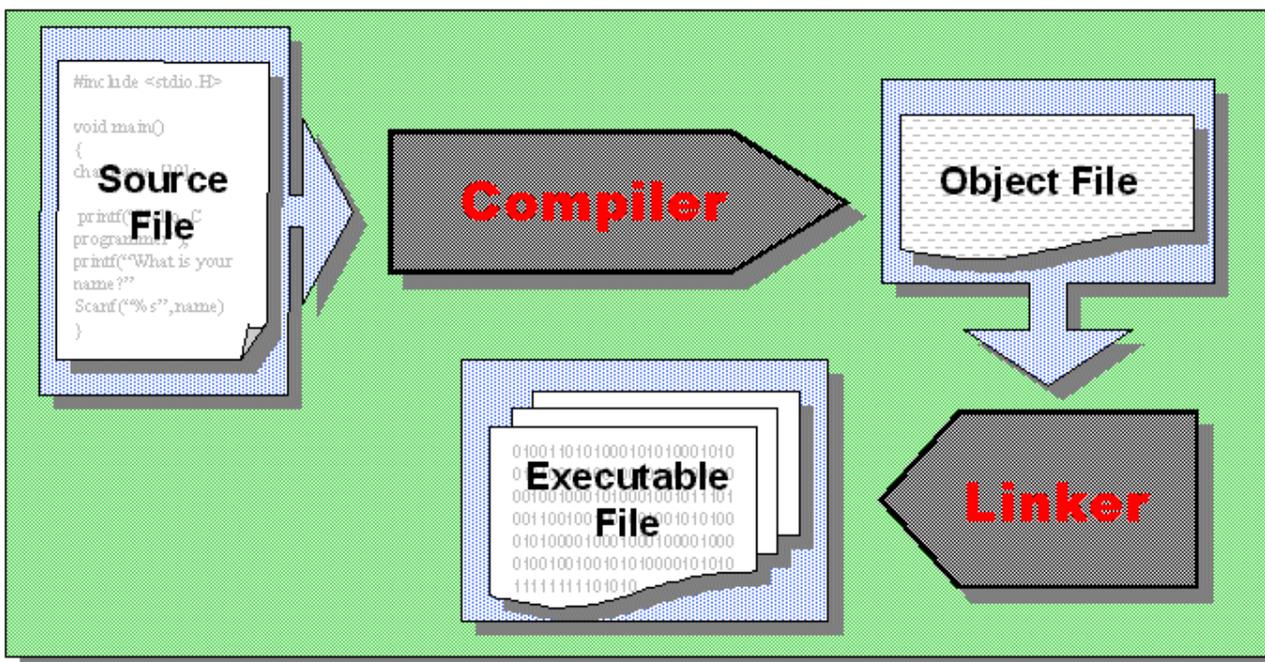
Many beginning C students start out by writing a program that looks like this:

```
#include<stdio.h>
void main()
{
    printf("Hello,
    world!");
}
```

This classic program (or old chestnut) simply displays the words "Hello,World!" on the computer screen. If you type this into any text editor and save it with a .c filename extension, you've created a source file.

Unfortunately, your computer cannot execute the instructions contained in a source file. It can't even understand them, because the C language in which source code is written is a high-level language--that is, it contains characters, symbols and words that represent complex, numbers-based instructions. The only instructions a computer can understand and execute are those

written in **machine language**, consisting entirely of numbers. So before your computer can carry out your C program, your compiler must convert your source file into an **object file**(an intermediate level); then a related program called a **linker** (which comes with the compiler) must convert the object file into an **executable file**. The drawing below illustrates the process.



## But isn't a compiler what you use to write source code?

You can write source code in any text-editing or word-processing program and then cut and paste it into a compiler, but most compilers come with their own text editors. Compiler text editors are geared specifically to the job of writing source code. They generally color the different program elements as in the "Hello,World!" example above). This enables programmers to catch mistakes early and to read their work more easily. Some compiler text editors also automatically indent or outdent lines according to standard C programming conventions.

## Do compilers help with debugging?

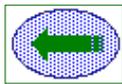
They do, in two important ways:.

- Before compiling, the font coloring and indentation features of the compiler's text editor help the programmer catch minor errors.
- During compiling, the compiler delivers error messages when it comes across instructions that it can't understand or that violate C rules.

See [Using the Miracle C Compiler](#) to find out how errors messages work in Miracle C.

## Are all compilers pretty much the same?

Compilers can vary widely in ease of use, and in the number and quality of debugging and other features to assist the programmer. And, of course, they can vary in price.



[To Top](#)

## *Downloading the Miracle C Compiler*

Downloading the Miracle C compiler is easy and should take only two or three minutes. Follow these steps:

1. Create a new folder on your C:\ drive. Name the folder "MCW".

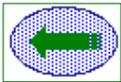


*Miracle C's path information is preset to C:\ MCW so it is important that you use this file name and location. You can change the location of your compiler later.*

2. Connect to the World Wide Web and go to the [Miracle C Download Page](http://www.ncf.carleton.ca/~bg283/) (<http://www.ncf.carleton.ca/~bg283/>).
3. Click Download Now.
4. Click Save File in the "Unknown File Type" Dialogue box.
5. Navigate through the "Save As" dialogue box to select the MCW folder you created.
6. Click Save.

The miracle.zip file is 246 kilobytes. With a 56k modem, it should take about 1 minute to download.

Now you're ready to set up your compiler.



[To Top](#)

---

## *Setting Up the Miracle C Compiler*

Now that you've downloaded the zip file, you have to extract (or "unzip") the compressed folders you downloaded and adjust your DOS window settings. If you don't have an extraction program on your computer, you can download one quickly from [Winzip](#).

### 1. Extract the compiler

- a. In the C:\MCW folder, right click on the folder Miracle.zip.
- b. Click Extract To. (Or Extract, depending on your extraction program).
- c. Type "C:\MCW" in the "Extract To" field. (Note: "C:\MCW" may appear automatically).
- d. Click Extract.
- e. Close the extraction program window if it is still open.

### 2. Adjust DOS Window settings

The programs you write, compile, and execute with **Miracle C** will run in a popup DOS window. Normally, such a window stays open only as long as the program is running. If your program is very short--as the programs you may write as a beginning C student are likely to be--the program might appear only as a brief flash on your screen.

In order to keep the DOS screen up for as long as you need to review your program's output, follow these steps:

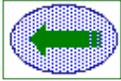
- a. Open the Windows folder (From the Desktop:My Computer>C>Windows)



- b. Right click on the `command` icon.

- c. Click Properties.
- d. Click the Program tab.
- e. Uncheck the Close on Exit checkbox and click OK.

Now you're ready to use **Miracle C**.



[To Top](#)



## *Using the Miracle C Compiler*

This section shows you how to use the basic functions of the Miracle c compiler. Click to go to the section you want to see.

- [Tutorial: Writing, saving and compiling the "Hello, World" program.](#)
- [How do I...?](#)
- [Understanding common error messages.](#)
- [Learning the C language with the Miracle C Language Library and Workbench Help.](#)

## **Tutorial: Writing, saving, and compiling the "Hello, World" program.**



1. Start **Miracle C** by opening the C:\MCW folder and clicking the MCW.exe icon.

2. Create a new source file. You can do this in one of three ways:



- Click the  button.
- Press Ctrl + N.
- Click File and New.

3. Type the following code into the text window:

```
/*Hellowor.c*/  
#include<stdio.h>  
  
void main()  
{  
printf("Hello,  
world!");  
}
```

Notice that the **Miracle C** text editor colors your code for you:

- **Red:** constants, including hard-coded text strings
- **Blue:** preprocessor directives, function types, and data types
- **Black:** function names and separators
- **Green:** comments

**4. Save your file.** You can do this in one of three ways:



- Click the  button.
- Press **Ctrl + S**.
- Click **File** and **Save**.

**5. Name your file.** Make sure you preserve the **.c** filename extension, for example "hellowor.c".



*Miracle C will only accept filenames of 8 characters or less, not including the .c extension.*

**6. Compile your program.** You can do this in one of three ways:

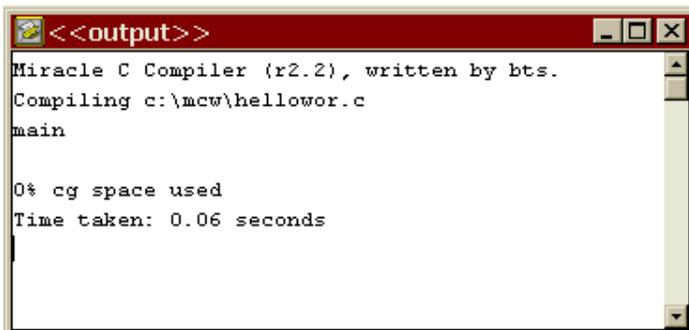


- Click the  button.
- Press **F6**.
- Click **Run** and **Compile**.



**If you have not registered your copy of **Miracle C**, you will see a reminder message every time you compile and link. Don't forget to register so you can obtain online support and compiler source code!!**

After compiling, Miracle C displays an output window showing the results of the compilation. If the compilation was successful, you see something like this:

A screenshot of a Windows-style output window titled '<<output>>'. The text inside the window reads: 'Miracle C Compiler (r2.2), written by bts.', 'Compiling c:\mcw\hellowor.c', 'main', '0% cg space used', and 'Time taken: 0.06 seconds'. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

If your output window looks like this, close it and proceed.

If your output window shows error messages, see [Understanding common error messages](#).

**7. Link your program.** You can do this in one of three ways:



- Click the  button.

- Press F7.
- Click Run and Build.

After linking, Miracle C displays an output window showing the results of the linking. If the linking was successful, you see something like this:



```
<<output>>
Miracle C Compiler (r2.2), written by bts.
Compiling c:\mcw\hellowor.c
main

0% cg space used
Time taken: 0 seconds

Linking object c:\mcw\hellowor.obj,,c:\mcw\hellowor.map,c:\mcw\ccl.lib ;

Build completed
```

If your output window looks like this, close it and proceed.

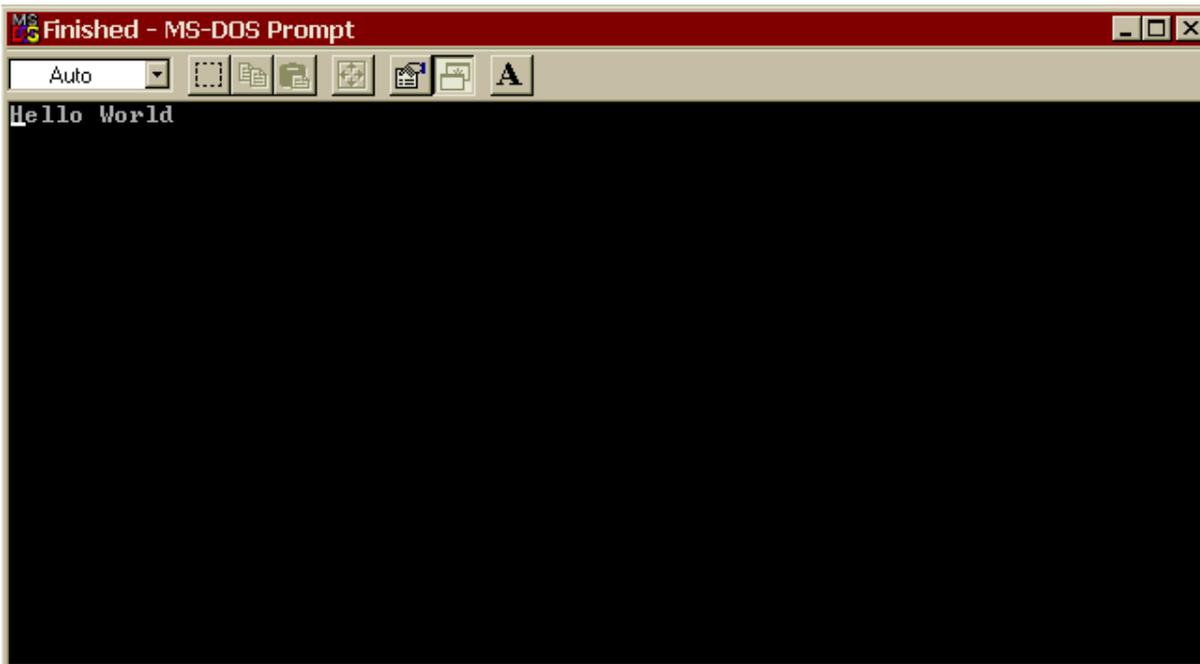
If your output window shows error messages, see [Understanding common error messages](#).

## 8. Run your program. You can do this in one of three ways:

- Click the  button.
- Press F8.
- Click Run and Execute.

Then click Run in the Execute Program dialogue box.

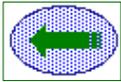
Here's what the output of your program should look like:



```
Finished - MS-DOS Prompt
Auto
Hello World
```

***Congratulations! You've just written, compiled, and run your first program with **Miracle C!*****

Back to [Using the Miracle C Compiler](#)



[To Top](#)

---

## How do I...?

Click on the topics below to find out how to perform specific tasks with **Miracle C**.

- [Starting](#) a new source file
- [Saving](#) a source file
- [Naming](#) a source file
- [Opening](#) a previously named source file
- [Compiling](#) a program
- [Linking](#) a program
- [Running](#) a program
- [Making](#) the compiler more accessible
- [Adjusting](#) the font your source code is displayed in
- [Checking](#) path information
- [Moving](#) the compiler to another directory
- [Renaming](#) the compiler directory

### *Starting a new source file*

To start a new source file, you can do one of three things:



- Click the  button.
- Click "File" and "New".
- Press Ctrl + N.

Back to [How do I...?](#)

---

### *Saving a source file*

To save your source file, you can do one of two things:



- Click the  button.
- Click "File" and "Save."
- Press Ctrl + S.

Back to [How do I...?](#)

---

## Naming a source file

To name your file, click **File** and **Save**, then type in the file name, being sure to preserve the .c filename extension (example: "hellowor.c").



*Miracle C will only accept filenames of 8 characters or less, not including the .c extension.*

Back to [How do I...?](#)

---

## Opening a previously saved source file

To open a source file you've already saved, you can do one of three things:



- Click the  button.
- Click "File" and "Open."
- Press Ctrl +O.

Back to [How do I...?](#)

---

## Compiling a program

To Compile your source file, you can do one of three things:



- Click the  button.
- Press F6.
- Select "Compile" in the Run menu.



**If you have not registered your copy of **Miracle C**, you will see a reminder message every time you compile and link. Don't forget to register to obtain online support and compiler source code!**

After compiling, Miracle C displays an output window showing the results of the compilation. If the compilation was successful, you see something like this:

```
<<output>>
Miracle C Compiler (r2.2), written by bts.
Compiling c:\mcw\hellowor.c
main

0% cg space used
Time taken: 0.06 seconds
```

If your output window looks like this, close it and proceed.

If your output window shows error messages, see [Understanding common error messages](#).

Back to [How do I...?](#)

---

## Linking a program

To link your object file, you can do one of three things:



- Click the  button.
- Press F7.
- Select "Build" in the Run menu.

After linking, Miracle C displays an output window showing the results of the linking. If the linking was successful, you see something like this:

```

Miracle C Compiler (r2.2), written by bts.
Compiling c:\mcw\hellowor.c
main

0% cg space used
Time taken: 0 seconds

Linking object c:\mcw\hellowor.obj,,c:\mcw\hellowor.map,c:\mcw\ccl.lib ;

Build completed

```

If your output window looks like this, close it and proceed.

If your output window shows error messages, see [Understanding common error messages](#).

Back to [How do I...?](#)

---

## Running a program

To run your program, you can do one of three things:



- Click the  button.
- Press F8.
- Select "Execute" in the Run menu.

Then click Run in the Execute Program dialogue box.

Back to [How do I...?](#)

---

## Making the compiler more accessible

You may not find it convenient to go into the C:\MCW folder every time you want to use Miracle C. You can make the program more accessible in two ways:

- Add a shortcut to your Windows desktop.
- Add Miracle C to your Programs menu.

## Here's how to add a shortcut to your Windows desktop:

1. Open the C:\MCW folder.



2. Right click the `Mcw.exe` icon.
3. Click Send to.
4. Click Desktop As Shortcut.

Now you can start Miracle C by clicking the icon on your desktop.

## Here's how to add Miracle C to your Programs menu:

1. Click Start, point to Settings, and then click Taskbar & Start Menu.
2. Click the Start Menu Programs tab.
3. Click Add, and then click Browse.



4. Locate `Mcw.exe`, then double click it.
5. Click Next, then double-click the menu on which you want the program to appear.
6. Type the name that you want to see on the Programs menu (for example, "Miracle C ", then click Finish.

Back to [How do I...?](#)

---

## *Adjusting the font your source code is displayed in*

Miracle C's default font is 8-point Courier New. To change the font, follow these steps:

1. After creating your source file, click File and then click Font.
2. Select font size and style, then click OK.

Back to [How do I...?](#)

---

## *Checking path information*

Path information tells the compiler where to look for the files it needs to compile your source code. You must be sure that the compiler's path information accurately reflects the location not only of the `Mcw.exe` program, but also of the other files which accompanied `Mcw.exe` in the download.

To check the path information, follow these steps:



1. In Windows Explorer, locate `Mcw.exe`. Make sure that the "Include" folder and the "ccl.lib" and "ledit.dll" files are in the same location as `Mcw.exe`. If they aren't, find them and move them.



2. Right click `Mcw.exe`, then click Properties.
3. Click the General tab.

4. Note the location information.
5. In the Miracle C menu bar, click Options, then click Compiler.
6. Look at the Base path information. It should be identical to the location information you saw in Windows Explorer. If it isn't, change it.
7. Look at the Include path information. It should be identical to the Base path information, but with `\include` added. If it isn't, change it.
8. Click OK.
9. In the Miracle C menu bar, click Options, then Linker.
10. Look at the library path information. It should be identical to the Base path information. If it isn't, change it.

Back to [How do I...?](#)

---

## ***Moving the compiler to another directory or renaming the compiler***

You can move or rename the MCW folder just as you would any other Windows folder. However, if you move or rename MCW, or if you remove the contents of MCW and place them in another folder, you must change Miracle C's path information to reflect the move. If you do not, the compiler will be unable to find the files it needs to process your source code.

After moving or renaming the MCW folder, adjust the compiler path information by following these steps:

1. On the Miracle C menu bar, click Options, then Compiler.
2. In the "Base path" field, type the current path information. For example, if you have moved the entire MCW file to the desktop, type `C:\windows\desktop\mcw`. If you have left the folder in the C:\ directory, but renamed it "Compiler", type `C:\Compiler`.
3. In the "Include path" field, type the same path information, then add `\include`. Example: `C:\windows\desktop\mcw\include`.
4. Click OK.
5. On the Miracle C menu bar, click Options, then Linker.
6. In the "Library path" field, type the same path information that you typed in the "Base path" field.
7. Click OK.

Back to [How do I...?](#)

---

## **Understanding Common Error Messages**

When a compiler translates your source code into object code, it expects to find code that conforms to C language rules. If it finds anything that doesn't conform to the rules, it displays an error message in the output window.

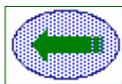
Error messages can have a variety of causes: faulty path information, missing header files, undeclared variables, and so forth. Most of the error messages you are likely to get when compiling simple programs, however, are the result of "syntax errors"--flaws not in the logic of your program but in the expression of that logic. Syntax errors are often merely typographical: missing semicolons, misplaced brackets, and so forth.

Unfortunately, no compiler can simply tell you "You forgot the semicolon after the statement," or "You forgot the header file." It can only tell you what it expected to find, but did not find, in your code. Because of this, C compiler error messages can be cryptic. With a little practice, though, you will be able to quickly locate and fix common errors in your code based on Miracle C's error messages.

The table below provides brief explanations of some of the most common error messages returned by Miracle C.

Error Message	Possible explanation	Error Example
#include stdio.h ignored--file not found.	Incorrect path information. Make sure the 'include path' and 'base path' in the Options>Compiler menu accurately reflect the location of your compiler . See <a href="#">Checking path information</a> .	(none--path-related error)
function not declared ('Hello World')	Missing header file (such as <stdio.h>).	<pre>void main() {     printf("Hello, world."); }</pre>
parse error (a), expecting `,' or `SEP' ('Hello World') }	Missing semicolon at the end of a statement.	<pre>#include void main() {     printf("Hello, world.") }</pre>
parse error (a), expecting `}' "	Missing bracket '}' at the end of a function.	<pre>#include void main() {     printf("Hello, world.");</pre>
parse error (a), expecting `)' " or `,' ('Hello World'"	Missing close parenthesis ')' at the end of a statement.	<pre>#include void main() {     printf("Hello, world."); {</pre>
parse error (a), expecting `SEP' '{ printf("Hello World")	Mistaken semicolon placed before the statements attached to a function.	<pre>#include void main(); {     printf("Hello, world."); {</pre>
variable 'Hello' not found '(Hello, world)'	Missing quotation marks around a string constant.	<pre>#include void main() {     printf(Hello, world); {</pre>
variable 'var' not found	Variable used without being declared.	<pre>#include void main() {     i=1;     printf("Hello, world. %i", i) }</pre>
parse error (a), expecting `)' " or `,' ('Hello World)'	Missing close quotes at the end of a string constant.	<pre>#include void main() {     printf("Hello, world); {</pre>

Back to [Using the Miracle C Compiler](#)



[To Top](#)

---

## Learning the C language with the Miracle C Language Library and Workbench Help.

Miracle C comes with an concise, easy-to-use C language library that you can access by clicking [Help](#) and [Language/Library](#) in the Miracle C menu bar.

Use the Library to:

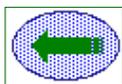
- Find definitions and examples of key C language features.
- Find lists of C lexical elements (operators, reserved words, and so forth).
- See what functions are located in the preset header files.
- Find the syntax and see examples of built-in C functions (*scanf()*, *strcat()*, etc.).

Miracle C also comes with a Workbench Help file that you can access by clicking [Help](#) and [Workbench Help](#) in the Miracle C menu bar.

Use Workbench Help to:

- Answer questions about the Miracle C user interface.
- View explanations and examples of Miracle C menu options.
- View a list of available sample programs.

Back to [Using the Miracle C Compiler](#)



[To Top](#)

---

## Basic Compiler Terminology

### Source File

File containing program instructions in their original form, written in a programming language such as C, C++, or Java. Source code uses common symbols (characters, words and phrases) to represent commands to the computer. C language source files carry `.c` filename extensions. See [Webopedia's definition](#) for more details and links to related information.

Back to [Compiler Basics](#)

---

### High-level language

A programming language that uses English-like characters, words, and phrases to represent program commands. A low-level language, by contrast, is one which employs symbols and syntax very different from English (or any other "natural" language), for example, 1's and 0's. See [Webopedia's definition](#) for more details and links to related information.

Back to [Compiler Basics](#)

---

## Machine language

The only language computers can understand. Machine language consists entirely of numbers, and is therefore too complex for most people to use directly. Programs are therefore written in a high-level language such as C or Pascal, and then translated into machine language by a compiler. See [Webopedia's definition](#) for more details and links to related information.

Back to [Compiler Basics](#)

---

## Object file

The file type produced by a compiler. Object code is an intermediary form between source code and executable code. Object files carry `.obj` filename extensions. See [Webopedia's definition](#) for more details and links to related information.

Back to [Compiler Basics](#)

---

## Linker

The program that converts an object file into an executable file. See [Webopedia's definition](#) for more details and links to related information.

Back to [Compiler Basics](#)

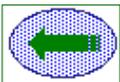
---

## Executable file

The final file type produced by the compiling and linking processes. An executable file consists entirely of machine language, and can be accessed and run by the computer for which it was compiled. Executable files carry `.exe` filename extensions. See [Webopedia's definition](#) for more details and links to related information.

Back to [Compiler Basics](#)

Back to [Using the Miracle C Compiler](#)



[To Top](#)

---

"Welcome to Miracle C" © 1999 Ted Barker