

If all goes well and the compiler is duly pleased with your source code, the compiler creates an object code file. It's a middle step, one that isn't necessary for smaller programs but that becomes vital for larger programs.

Finally, the compiler links the object code file, which creates a real, live computer program.

If either the compiler or the linker doesn't understand something, an error message is displayed. At that point, you can gnash your teeth and sit and stew. Then go back and edit the source code file again, fixing whatever error the compiler found. (It isn't as tough as it sounds.) Then you attempt to compile the program again — you recompile and relink.



- ✓ The compiler translates the information in the source code file into instructions the computer can understand. The linker then converts that information into a runnable program.
- ✓ The GCC compiler recommended and used in this book combines the compiling and linking steps. An object file *is* created by GCC, but it is automatically deleted when the final program file is created.
- ✓ Object code files end in OBJ or sometimes just O. The first part of the object file name is the same as the source code filename.
- ✓ Feel free to cheerfully forget all this object code nonsense for now.
- ✓ Text editor ⇄ Compiler.
- ✓ Source code ⇄ Program.

## Compiling GOODBYE.C

The gritty details for compiling a program are in Appendix A. Assuming that you have thumbed through it already, use your powerful human memory to recall the proper command to compile and link the GOODBYE.C source code. Here's a hint:

```
gcc goodbye.c -o goodbye
```

Type that command at your command prompt and see what happens.

Well?

Nothing happens! If you have done everything properly, the GCC compiler merely creates the final program file for you. The only time you see a message is if you goof up and an error occurs in the program.