

## Please Don't Leave Me Out!

What exactly does the following line mean?

```
#include <stdio.h>
```

It's an instruction for the compiler to do something, to *include* a special file on disk, one named `STDIO.H`, in with your source code.

Figure 23-1 illustrates the concept for the `#include <stdio.h>` instruction. The contents of the `STDIO.H` file are read from disk and included (inserted) into your source code file when it's compiled.

Figure 23-2 shows how several lines of `#includes` work. Each file is read from disk and inserted into your source code, one after the other, as the source code is compiled.

## Say! Aren't you the `#include` construction?

The `#include` construction is used to tell the compiler to copy lines from a *header file* into your source code. This instruction is required by the compiler for using many of the C language functions. The header file contains information about how the functions are used (*yes, prototypes*), as well as other information that helps the compiler understand your program.

Here's the format for using `#include`:

```
#include <filename>
```

The `#include` directive is followed by a *filename* held in angle brackets. The *filename* must be in lowercase and typically (though it's not a rule) ends with a period and a little *h*. Like all `#-`sign things at the start of your source code, don't end this line with a semicolon!

Sometimes, the filename is a partial path, in which case the partial path needs to be included, as in

```
#include <sys/socket.h>
```

The path is `sys/`, followed by the header filename, `socket.h`.