**8.1.2 Using the bin Format To Generate .EXE Files**

The .EXE file format is simple enough that it's possible to build a .EXE file by writing a pure-binary program and sticking a 32-byte header on the front. This header is simple enough that it can be generated using DB and DW commands by NASM itself, so that you can use the bin output format to directly generate .EXE files.

Included in the NASM archives, in the misc subdirectory, is a file exebin.mac of macros. It defines three macros: EXE\_begin, EXE\_stack and EXE\_end.

To produce a .EXE file using this method, you should start by using %include to load the exebin.mac macro package into your source file. You should then issue the EXE\_begin macro call (which takes no arguments) to generate the file header data. Then write code as normal for the bin format – you can use all three standard sections .text, .data and .bss. At the end of the file you should call the EXE\_end macro (again, no arguments), which defines some symbols to mark section sizes, and these symbols are referred to in the header code generated by EXE\_begin.

In this model, the code you end up writing starts at 0x100, just like a .COM file – in fact, if you strip off the 32-byte header from the resulting .EXE file, you will have a valid .COM program. All the segment bases are the same, so you are limited to a 64K program, again just like a .COM file. Note that an ORG directive is issued by the EXE\_begin macro, so you should not explicitly issue one of your own.

You can't directly refer to your segment base value, unfortunately, since this would require a relocation in the header, and things would get a lot more complicated. So you should get your segment base by copying it out of CS instead.

On entry to your .EXE file, SS:SP are already set up to point to the top of a 2Kb stack. You can adjust the default stack size of 2Kb by calling the EXE\_stack macro. For example, to change the stack size of your program to 64 bytes, you would call EXE\_stack 64.

A sample program which generates a .EXE file in this way is given in the test subdirectory of the NASM archive, as binexe.asm.