

## *Equal-Sign Directive*

The equal-sign directive associates a symbol name with an integer expression.

```
name = expression
```

Ordinarily, expression is a 32-bit integer value. When a program is assembled, all occurrences of name are replaced by expression during the assembler's preprocessor step.

Suppose the following statement occurs near the beginning of a source code file:

```
COUNT = 500
```

Further, suppose the following statement should be found in the file 10 lines later:

```
mov eax, COUNT
```

When the file is assembled, MASM will scan the source file and produce the corresponding code lines:

```
mov eax, 500
```

## Why Use Symbols?

We might have skipped the COUNT symbol entirely and simply coded the MOV instruction with the literal 500, but experience has shown that programs are easier to read and maintain if symbols are used.

Suppose COUNT were used many times throughout a program. At a later time, we could easily redefine its value:

```
COUNT = 600
```

Assuming that the source file was assembled again, all instances of COUNT would be automatically replaced by the value 600.

