Executable Extensions

The syntax for creating an executable extension is:

```
set .ext=command [options]
```

This tells **CMD.EXE** to run the specified command whenever you name a file with the extension **.ext** at the prompt.

.EXT is the executable file extension; *command* is the name of the internal command, external program, alias, or batch file to run; and *[options]* are any command-line startup options you want to specify for the program, batch file, or alias.

Normally, when you type a filename (as opposed to an alias or internal command name) as the first word on the command line, **CMD.EXE** looks for a file with that name to execute. The file's extension may be *.EXE* or *.COM* to indicate that it contains a program, it may have a batch file extension like *.BTM*, or the file's contents may indicate that it is executable. (To change the default list of extensions see the PATHEXT environment variable and the PathExt setting in *CMD.INI*.)

You can add to the default list of extensions, and have **CMD.EXE** take the action you want with files that are not executable programs or batch files. The action taken is always based on the file's extension. For example, you could start your text editor whenever you type the name of a *.DOC* file, or start your database manager whenever you type the name of a *.DAT* file.

You can use environment variables to define the internal command, external program, batch file, or alias to run for each defined file extension. To create an executable extension, use the SET command to create a new environment variable. An environment variable is recognized as an executable extension if its name begins with a period.

For example, if you want to run a word processor called *EDITOR* whenever you type the name of a file that has an extension of *.EDT*, you could use this command:

```
[c:\] set .edt=c:\edit\editor.exe
```

If the command specified in an executable extension is a batch file or external program, **CMD.EXE** will search the PATH for it if necessary. However, you can make sure that the correct program or batch file is used, and speed up the executable extension, by specifying the full name including drive, path, filename, and extension.

Once an executable extension is defined, any time you name a file with that extension the corresponding program, batch file, or alias is started, with the name of your file passed to it as a parameter.

The following example defines *B.EXE* (the Boxer text editor) as the processor for *.MAK* files:

```
[c:\] set .mak=c:\boxer\b.exe -s
```

With this definition, if you have a file named *INIT.MAK* in the current directory and enter the command:

```
        [c:\source] init
```

CMD.EXE will execute the command:

```
        c:\boxer\b.exe -s c:\source\init.mak
```

Notice that the full pathname of *INIT.MAK* is automatically included. If you enter parameters on the command line, they are appended to the end of the command. For example, if you changed the above entry to:

```
        [c:\source] init -w
```

the command processor would execute the command:

```
        c:\boxer\b.exe -s c:\source\init.mak -w
```

In order for executable extensions to work, the command, program, batch file, or alias must be able to interpret the command line properly. For example, if a program you want to run doesn't accept a file name on its command line as shown in these examples, then executable extensions won't work with that program.

Executable extensions may include wildcards, so you could, for example, run your text editor for any file with an extension beginning with **T** by defining an executable extension called *.T**. Extended wildcards (*e.g.*, *DO[CT]* for *.DOC* and *.DOT* files) may also be used.

The search for executable files starts in the current directory, then proceeds to each subdirectory specified by the PATH environment variable (if a "." is used in the PATH the current directory is not searched first; see the PATH command for details).

You may need to take this search order into account when using executable extensions. Using the *.MAK* example above, if you had a file named *FORMAT.MAK* in the current directory and entered the command FORMAT A:, your command would run the Boxer editor specified by the executable extension, instead of finding the standard **OS/2** FORMAT command as you perhaps intended. You can get around this by remembering that the OS/2 FORMAT command is in the file *FORMAT.COM*. If you entered the command FORMAT.COM A: then the *.MAK* executable extension would not be checked, and the search would continue until it found the *FORMAT.COM* file.

To remove an executable extension, use the UNSET command to remove the corresponding variable.

From:
http://www.osfree.org/doku/ - **osFree wiki**

Permanent link:
http://www.osfree.org/doku/doku.php?id=en:docs:cmd:file:ext&rev=1400924197

Last update: **2014/05/24 09:36**