

## **SEARCH PATH – How it's used:**

Before discussing the concept of a SEARCH PATH, there are a few things you must be familiar with.

(Information on the PATH command is provided at the end of this document.)

### **Command Processor**

The *cmd.exe* file is the command processor or command interpreter. It gets loaded into RAM when you start the computer. Its job is to process or interpret the information that you provide to it. For example, if you type a command at the command line; it reads the command and decides what to do with it. If you click on an icon on the desktop; it's the command processor that handles your request. Some commands are *built-in* to the *cmd.exe* file. (Note: In Windows98 – the command interpreter file is *command.com*.)

### **Internal Commands**

Commands that are loaded into memory when the computer is started are internal commands. They are always stored in memory and therefore easily accessible by the operating system. Internal commands are part of the *cmd.exe (xp)/command.com (98)* file. When executing an internal command, the operating system finds and executes the command in memory.

### **External Commands**

Commands that are not part of the command interpreter file and that are not loaded into memory when the computer is booted are referred to as External Commands. They are physical files that can be found on the hard drive. When executing an external command, the operating system must go to the hard drive, find the command, load it into memory and then execute it.

## **SEARCH PATH**

When executing any command from the command line the operating system must find the command before it can be executed. Here's how the search works:

- 1) It searches Memory (RAM) – if the command is an internal command, then it would be found in memory and executed.
- 2) If the command is not found in memory, then the operating system will look for the file in the directory from which you're executing the command. This directory is referred to as your *default directory* or your *current location*. (i.e., if the prompt from which you're executing the command reads: C:\DATA, the operating system would search the C:\DATA directory for a file that matches the name of the command that you typed.)
- 3) If the file is not found in the default directory, the operating system will begin sequentially searching each directory listed in the *search path*. (i.e., if the *search path* is set to C:\WINDOWS\SYSTEM32;C:\MYSTUFF, the operating system would search the first directory in the path (in this case, the *c:\windows\system32* directory); if the file is found there, it's

loaded from the hard drive into RAM and executed. If it were not found, the operating system would search the next directory in the path (in this case, the *c:\mystuff* directory). If the file is found there, it's loaded into memory and executed. In this example there's only two directories in the Search Path. If there were more directories, the search would continue through each of the directories.

**Note:** To see the current value of the search path, type PATH at a command prompt.

- 4) If the file is not found in any of the directories in the Search Path, the user would receive a message indicating "*the command is not recognized as an internal or external command*". (Note: Windows 98 users would receive a message that read: "Bad Command or File name".)

### Example 1:

Search Path = C:\WINDOWS\SYSTEM32;C:\NOVELL\CLIENT32;C:\PROGS

Default Directory: C:\MYFILES>

Assume the following command is executed

C:\MYFILES>CONFIG

Here's what would happen:

- 1) The operating system would check memory for this command. It's not an internal command so the search would continue.
- 2) The default directory (*C:\myfiles>*) would be searched for a file with the name of CONFIG\*. The file would not be found in this directory therefore the search would continue.
- 3) The *C:\windows\system32* directory would be searched for a file with the name of CONFIG. The file would not be found in this directory therefore the search would continue.
- 4) The *C:\novell\client32* directory would be searched for a file with the name of CONFIG. The file would not be found in this directory therefore the search would continue.
- 5) The *C:\progs* directory would be searched for a file with the name of CONFIG. The file would not be found in this directory therefore the search would continue.
- 6) Since there are no directories remaining in the search path and the file was not found, the message "*the command is not recognized as an internal or external command*" would be displayed (Windows 98 users would see the "*bad command or filename*" message.)

\*The operating system would search for any form of an executable that might exist for this file. Executables have extensions of .com, .exe, and .bat. This means that the operating system would actually be searching for any one of the following files: config.com; config.exe or config.bat. There are additional file types that will be searched for in XP, but typically an executable is identified as a file with an extension of .bat, .exe or .com. The additional file extensions that will be searched for are listed on page 168 in the textbook.

### Example 2:

Assume the same Search Path and default directory as in Example 1.

Assume the following command is executed

```
C:\MYFILES>EDIT
```

Here's what would happen:

- 1) The operating system would check memory for this command. It's not an internal command so the search would continue.
- 2) The default directory (*C:\myfiles>*) would be searched for a file with the name of EDIT\*. The file would not be found in this directory therefore the search would continue.
- 3) The *C:\windows\system32* directory would be searched for a file with the name of EDIT. The file *would* be found in this directory therefore the operating system would load the file from disk into memory and execute the command.

\*The operating system would search for any form of an executable that might exist for this file. Executables have extensions of .com, .exe, and .bat. This means that the operating system would actually be searching for any one of the following files: edit.com; edit.exe or edit.bat. The edit command is covered later in the semester. It does exist in the *C:\windows\system32* directory and the actual file's name is edit.com.

### Example 3:

Search Path = C:\;C:\NOVELL\CLIENT32;C:\PROGS

Default Directory: C:\WINDOWS\SYSTEM32>

Assume the following command is executed

```
C:\WINDOWS\COMMAND>CHKDSK
```

Here's what would happen:

- 1) The operating system would check memory for this command. It's not an internal command so the search would continue.
- 2) The default directory (*C:\windows\system32>*) would be searched for a file with the name of CHKDSK\*. The file *would* be found in this directory

therefore the operating system would load the file from disk into memory and execute the command.

\*The operating system would search for any form of an executable that might exist for this file. Executables have extensions of .com, .exe, and .bat. This means that the operating system would actually be searching for any one of the following files: chkdisk.com; chkdisk.exe or chkdisk.bat. The chkdisk command is covered later in the semester. It does exist in the C:\windows\system32 directory and the actual file's name is chkdisk.exe.

#### Example 4:

Search Path = C:;\;C:\NOVELL\CLIENT32

Default Directory: C:\MYFILES>

Assume the following command is executed

C:\MYFILES>CHKDSK

Here's what would happen:

- 1) The operating system would check memory for this command. It's not an internal command so the search would continue.
- 2) The default directory (*C:\myfiles>*) would be searched for a file with the name of CHKDSK\*. The file would not be found in this directory therefore the search would continue.
- 3) The *C:\* directory would be searched for a file with the name of CHKDSK. The file would not be found in this directory therefore the search would continue.
- 4) The *C:\novell\client32* directory would be searched for a file with the name of CHKDSK. The file would not be found in this directory therefore the search would continue.
- 5) Since there are no directories remaining in the search path and the file was not found, the message *'the command is not recognized as an internal or external command'* would be displayed (Windows 98 users would see the *"bad command or filename"* message.)

\*The operating system would search for any form of an executable that might exist for this file. Executables have extensions of .com, .exe, and .bat. This means that the operating system would actually be searching for any one of the following files: chkdisk.com; chkdisk.exe or chkdisk.bat. This *is* a valid command and is stored in the C:\windows\system32 directory. The actual file name is chkdisk.exe, however since the C:\windows\system32 directory is not included in the Search Path *and* it's not the default directory, the search is unsuccessful.

#### Example 5:

Search Path = C:;\;C:\NOVELL\CLIENT32

Default Directory: C:\MYFILES>

Assume the following command is executed

```
C:\MYFILES>DIR
```

Here's what would happen:

- 1) The operating system would check memory for this command. It is an internal command therefore the operating system would execute the command from RAM.

## PATH COMMAND

The below table provides examples of using the PATH command. For the syntax of the command, type `PATH /?` at the command prompt or `HELP PATH`

A few comments worth mentioning..... when changing the value of the search path from a command prompt, it changes it for only that window. If you close the window and reopen another command prompt window, the system's default value for the PATH will automatically be re-established. The default system value for path is provided through the graphical interface (properties of my computer / Advanced/ Environment Variables / System Variables).

PATH	Displays the existing search path setting
PATH ;	Clears the existing search path setting. This would direct the command interpreter to search only the current directory (after searching RAM) when searching for executables.
PATH C:\windows\system32;c:\windows;c:\progs	Sets the PATH to search the three directories listed (C:\windows\system32, c:\windows and C:\progs). When entering the directories to include in the path, they must be separated by a semi-colon (;).
PATH %PATH%;C:\progs	The %path% variable stores the current path. Therefore, if the current path were C:\windows\system32, the %path% variable's value would be the same. Using this variable in the PATH command provides a means of appending an additional directory to an existing path. In this example, the C:\progs directory would be added to the existing path.

