ADB03

COLLABORATORS

	TITLE : ADB03	

ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		July 7, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

ADB03

1.1 AmigaDOS For Beginners - Part 3 - Files Directories and Paths

```
                AMIGADOS FOR BEGINNERS

BY FRANK BUNTON

 COPYRIGHT © FRANK P. BUNTON 1993-1998

PART 3 - FILES DIRECTORIES AND PATHS

This article contains the topics listed below. The concepts discussed
are rather important. If you find them rather hard going at this stage
then please make a note to come back to them after you have read further
into the articles on this disk. You will, in all likelihood, understand
these topics much better after reading them again at a later time.


                 Files

                 Naming Files And Directories

                 Directories

                 The Current Directory

                 The PATH Concept
               === End of Text ===
```

1.2 Part 3 - Directories Files & Paths - Files

```
FILES

A disk file is any item that has been saved to the disk. Files include
items such as:-

- Programs
```

- AmigaDOS Commands, Libraries and other related items.
- Word processing documents
- Data bases
- Pictures
- Animations
- Icons
- etc. etc.
=== End of Text ===

1.3 Part 3 - Directories Files & Paths - Naming Files & Directories

NAMING FILES DIRECTORIES AND DISKS

The name of a file, directory or disk can consist of up to 30 characters. These characters can be any that can be entered from the keyboard except for the following characters which are either prohibited or inadvisable:-

: / # ? % | ( ) [ ] * " ; ` > <

These are all used in special ways in AmigaDOS.

The first two : / will not be accepted as part of a file, directory or disk name because they are used in the
path names
.

The rest will be accepted in some circumstances but not in others.
They can all cause problems if they are accepted. Don't use any of them!

In my opinion it is best to stick to the following sets of characters, then you will not have any problems:-

- alphabetic a to z and A to Z
- numeric 0 to 9
- dot .
- hyphen -
- underscore _
- space
- back slash \

The dot (.) character is often used to add a suffix to a file name. For example:-

Name.BAS indicates a program written in basic language
Name.LHA indicates a file compressed using the LHA program
Name.EXE indicates the file is an executable program
Name.TXT indicates a plain text file that can be read.
Name.DOC indicates a document saved from a word processor
or a document (text) file associated with a program.

Such suffixes, although they can be helpful, are by no means necessary unless a particular program requires that they be present.

AmigaDOS itself uses files with certain suffixes such as:-

```
   Name.info     to name an icon file
   Name.library  to identify a library file
   Name.device   to identify device drivers
   Name.font     to identify a font directory information file
```

In MS-DOS you are restricted to a file name of 8 characters plus an extension of a dot and three more characters. There are no such restrictions in AmigaDOS.

Many people do not like using spaces and use the underscore instead. This is because, if a file or directory name includes a space, it often has to be enclosed in inverted commas to make AmigaDOS recognise the name including the spaces as a single name. Without the spaces AmigaDOS would consider the name as different names separated by spaces. For example:-

```
  My First File
```

could be considered by AmigaDOS as three separate file names whereas these two alternatives would be taken as a single file name:-

```
  "My First File"

   My_First_File
```

When using inverted commas, they must include the full path to the file (see
PATH
) and not just the file name.

=== End of Text ===

1.4 Part 3 - Directories Files & Paths - Directories

DIRECTORIES

A directory is a means of combining a number of files, and maybe some sub-directories, into the one convenient group. A "sub-directory" is one directory contained within another directory (See
PATH
).

The rules for giving names to directories is the same as
that for files
.

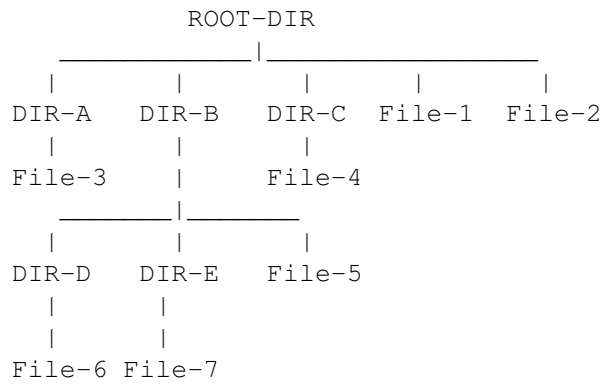
In Shell/CLI, directories are the equivalent of the Workbench "drawers" that appear when you open a disk icon and find a disk window full of drawers. You will probably have noticed that some disk windows contain both drawers and files. When you double click on a drawer icon to open it, you will probably find file icons in its window and you might find other drawer icons in the window as well which can, in turn, be opened.

New Amiga users, especially those who previously had a Commodore 64 or 128, will have to get used to the idea that each disk can be divided into a number of directories and sub-directories (i.e. directories within directories). The Commodore 64 & 128 stored all their files in one directory. If you looked at that one directory you saw everything on the disk.

With the Amiga you can, if you like, put everything into the one directory. However, this would be like throwing hundreds of (physical) documents into a big box. It would take too long to find anything. With letters and documents (paper ones, not computer ones) it is best to put related articles into the one folder then related folders into the same filing cabinet drawer. Filing on Amiga disks is very similar.

Briefly, each disk has a "root" or "main" directory which can hold other directories as well as individual files. The "root" directory is the equivalent of Workbench's "disk window".

Each directory can hold sub-directories (i.e. a directory within a directory) as well as files:-



This structure (above) can be likened to a filing cabinet with three drawers (A, B and C). The cabinet also has a couple of papers sitting on top of it (File-1 and File-2). If we open Drawer-A we find File-3. If we open Drawer-C we find File-4. If we open Drawer-B we find two other drawers (D and E) as well as File-5. These extra drawers inside drawers could be likened to having a couple of boxes inside a filing cabinet drawer to hold special papers.

The analogy to a filing cabinet breaks down when you consider that, when one filing cabinet drawer is full and all the others are empty, the overall filing cabinet capacity is still nowhere near full and the other drawers still have plenty of room in them.

With a disk, however, you can make, say, three drawers but keep putting all the files into only one of them until the whole disk is full. The two empty drawers cannot then have anything put into them because the disk is full.

The size of a disk directory or sub-directory is not fixed in any way except by the overall capacity of the disk. It expands and contracts according to need. Any spare space on the disk can be allocated to any directory that needs it. When the disk is full, all its directories and sub directories are also full.

=== End of Text ===

1.5 Part 3 - Directories Files & Paths - Current Directory

THE "CURRENT DIRECTORY" CONCEPT

The "Current Directory" concept is very important when working in CLI or SHELL. Each CLI window operates in such a way as to recognise a particular directory as the "current" one. It can either be the disk's root directory or one of the sub-directories. The "Current Directory" can be any directory or sub directory on any disk within any of the disk drives attached to your computer, or in the RAM: or RAD: devices.

When you first open the SHELL window, it should have a prompt showing one of these:-

```
1.SYS:>
1.Workbench1.3:>
1.Workbench2.0:>
```

"SYS:" means the "root" directory of the "SYStem" disk. The system disk is the one used to boot up and it is always called "SYS:" by the operating system. The other two indicate that the "current directory" is the root directory of the Workbench disk.

Using the CD command you can change the current directory to any directory or sub-directory in any drive. For example, if you have one of the above prompts showing and you entered this:-

```
> CD devs
```

(Don't type in the ">". Press the return key after typing in "CD devs")

your prompt would become, for example:-

```
1.Workbench2.0:Devs >
```

which would indicate that the current directory is now the "Devs" directory on the workbench disk.

Changing the current directory to that in which you wish to work means that you do not have to specify the full "path" name with each command that you enter. Another advantage of changing the current directory is that some programs, when started from CLI, will look in the current directory for other files that they need. If the current directory is not the correct one then the program may fail.

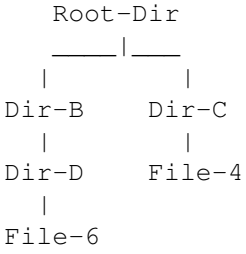
=== End of Text ===

1.6 Part 3 - Directories Files & Paths - The Path Concept

THE "PATH" CONCEPT

Another important concept that you need to master is that of "PATHS". Put simply, this means the "way" that the system must go to find a particular directory or file.

In the diagram given under the heading ‘Directories’, the "path" to "File-6" is from the root directory through "Dir-B" and "Dir-D" whereas the "path" to "File-4" is through "Dir-C". These paths are:-



In AmigaDOS terminology these would be shown as:-

```
DiskName:Dir-B/Dir-D/File-6
DiskName:Dir-C/File-4
```

The colon (:) and the slash (/) are "path separators".

The colon (:) is used to separate the name of a disk or the drive that it is in from the rest of the path. You must not use the colon for anything else.

The slash (/) is used to separate directory names from sub-directory names and from file names. It must not be used for anything else. For people who are used to MS-DOS please note that it is the REVERSE of the slash used in that system. MS-DOS uses \. (This gets VERY confusing if you are constantly changing back and forth from AmigaDOS to MS-DOS!!)

Note that "DiskName:" (e.g. "Workbenchx.x:") is the root directory of the disk.

For example, the "path" to find a particular printer driver in the workbench disk’s sub-directory "printers", which is in the directory "devs", is shown as follows:-

```
Workbenchx.x:devs/printers/printer_driver
```

or, if you are sure that the disk you want is in a particular drive then you can use a drive name instead of a disk name. For example, if a disk is in "df0:", you can use use:-

```
df0:devs/printers/printer_driver
```

If you are sure that the path you want is on the same disk as the current directory then the device name can be left out and you simply use:-

```
  :devs/printers/printer_driver

If the current directory is "df0:devs" then you can use:-

  printers/printer_driver

If the current directory is "df0:devs/printers" then you can use just
the file name, as in:-

  printer_driver

If the file name is in the root directory of the disk, then you simply
use:-

  df0:file_name or :file_name

Note that there cannot be any spaces in the path name unless the complete
path (including the device or disk name) is enclosed in quotation marks.
For example:-

  "df0:devs/printers/printer driver"

This is why you will often see file names with "_" instead of spaces.
Some people prefer to type the "_" instead of a space to save having to
use inverted commas, as in:-

   df0:devs/printers/printer_driver

=== End of Text ===
```