

## **AmigaDOS For Beginners - Glossary**

**COLLABORATORS**

	<i>TITLE :</i> AmigaDOS For Beginners - Glossary		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		August 27, 2022	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>AmigaDOS For Beginners - Glossary</b>	<b>1</b>
1.1	AmigaDOS For Beginners - Glossary	1
1.2	AmigaDOS For Beginners - Glossary - DOS	3
1.3	AmigaDOS For Beginners - Glossary - ARexx	5
1.4	AmigaDOS For Beginners - Glossary - Arguments	5
1.5	AmigaDOS For Beginners - Glossary - ASCII	6
1.6	AmigaDOS For Beginners - Glossary - Booting	6
1.7	AmigaDOS For Beginners - Glossary - KickStart	8
1.8	AmigaDOS For Beginners - Glossary - Boot Block	8
1.9	AmigaDOS For Beginners - Glossary - Startup-Sequence	9
1.10	AmigaDOS For Beginners - Glossary - Bytes	9
1.11	AmigaDOS For Beginners - Glossary - C.H.U.G.	10
1.12	AmigaDOS For Beginners - Glossary - CLI And Shell	11
1.13	AmigaDOS For Beginners - Glossary - Commands	11
1.14	AmigaDOS For Beginners - Glossary - Keywords	12
1.15	AmigaDOS For Beginners - Glossary - Options	12
1.16	AmigaDOS For Beginners - Glossary - Commands Lines	13
1.17	AmigaDOS For Beginners - Glossary - CPU	14
1.18	AmigaDOS For Beginners - Glossary - Custom Chips	14
1.19	AmigaDOS For Beginners - Glossary - Devices	15
1.20	AmigaDOS For Beginners - Glossary - Physical Devices	15
1.21	AmigaDOS For Beginners - Glossary - Logical Devices	16
1.22	AmigaDOS For Beginners - Glossary - Directory Utility	17
1.23	AmigaDOS For Beginners - Glossary - Disks & Drives	18
1.24	AmigaDOS For Beginners - Glossary - Disk Structure & File Systems	19
1.25	AmigaDOS For Beginners - Glossary - Header Block	21
1.26	AmigaDOS For Beginners - Glossary - Comments & FileNotes	21
1.27	AmigaDOS For Beginners - Glossary - Format & Template	22
1.28	AmigaDOS For Beginners - Glossary - Information Window	22
1.29	AmigaDOS For Beginners - Glossary - Mounting	23

---

1.30 AmigaDOS For Beginners - Glossary - Number Systems . . . . .	25
1.31 AmigaDOS For Beginners - Glossary - Comparative Table Of Number Systems . . . . .	25
1.32 AmigaDOS For Beginners - Glossary - Decimal Number System . . . . .	26
1.33 AmigaDOS For Beginners - Glossary - Binary Number System . . . . .	27
1.34 AmigaDOS For Beginners - Glossary - Octal Number System . . . . .	28
1.35 AmigaDOS For Beginners - Glossary - Hexadecimal Number System . . . . .	28
1.36 AmigaDOS For Beginners - Glossary - Operating System . . . . .	29
1.37 AmigaDOS For Beginners - Glossary - Public Domain . . . . .	31
1.38 AmigaDOS For Beginners - Glossary - Protection Bits . . . . .	32
1.39 AmigaDOS For Beginners - Glossary - Memory RAM & ROM . . . . .	33
1.40 AmigaDOS For Beginners - Glossary - ROM . . . . .	34
1.41 AmigaDOS For Beginners - Glossary - RAM . . . . .	34
1.42 AmigaDOS For Beginners - Glossary - CHIP . . . . .	35
1.43 AmigaDOS For Beginners - Glossary - FAST . . . . .	35
1.44 AmigaDOS For Beginners - Glossary - SLOW . . . . .	36
1.45 AmigaDOS For Beginners - Glossary - RAM & RAD Disks . . . . .	36
1.46 AmigaDOS For Beginners - Glossary - Scripts . . . . .	37
1.47 AmigaDOS For Beginners - Glossary - Stack . . . . .	37
1.48 AmigaDOS For Beginners - Glossary - Strings . . . . .	38
1.49 AmigaDOS For Beginners - Glossary - Viruses . . . . .	39
1.50 AmigaDOS For Beginners - Glossary - WildCards . . . . .	40
1.51 AmigaDOS For Beginners - Table Of ASCII Codes . . . . .	40
1.52 AmigaDOS For Beginners - Table Of ASCII Codes - Printable . . . . .	41
1.53 AmigaDOS For Beginners - Table Of ASCII Codes - Non Printable . . . . .	42
1.54 AmigaDOS For Beginners - Table Of ASCII Codes - Non Standard . . . . .	42

---

## Chapter 1

# AmigaDOS For Beginners - Glossary

### 1.1 AmigaDOS For Beginners - Glossary

AMIGADOS FOR BEGINNERS

GLOSSARY

COPYRIGHT F.P. BUNTON 1993-1998

AmigaDOS

ARexx

Arguments

ASCII

ASCII Codes

Binary Numbers

Boot Block

Booting

Bytes

C.H.U.G.

CHIP Memory

CLI And Shell

Commands

Command Keywords

Command Lines

Command Options

Comments & FileNotes

CPU

Custom Chip Set

Decimal Numbers

Devices

Directory Utilities

Disks & Drives

Disk Structure

DOS

FAST Memory

FileNotes & Comments

File System

Format & Template

Header Blocks

Hexadecimal Numbers

Information Window

Keywords

Kickstart

Logical Devices

Memory

Memory - CHIP

Memory - FAST

Memory - SLOW

Memory - RAM

Memory - ROM

MSDOS

Mounting

Number Systems

Octal Numbers  
Operating System  
Options in Commands  
Peripherals  
Physical Devices  
Protection Bits  
Public Domain  
RAM: & RAD: Disks  
RAM Memory  
ROM Memory  
Scripts  
SLOW Memory  
Stack  
Startup-Sequence  
Strings  
Viruses  
Wildcards  
=== End of Glossary ===

## 1.2 AmigaDOS For Beginners - Glossary - DOS

AMIGADOS FOR BEGINNERS - GLOSSARY

DOS

DOS was originally an acronym for Disk  
Operating System

. Some say it should

be "Disk Based Operating System" as it does a lot more than operate a  
disk and that it was really an operating system based (stored) on a disks.  
Some say it should be

Device

Operating System. These days a lot of the  
operating system is hard coded into the Amiga's

ROM

chips so that it is  
no longer disk based. It is best to forget that DOS ever stood for

---

anything.

AmigaDOS is the term applied to Amiga's equivalent of MS-DOS (see below) that is used on IBM compatible computers.

AmigaDOS is not just a set of commands nor is it just a  
Shell or CLI  
window.

It is the software side of the Amiga's complete operating system. It controls all the functions of the computer such as multitasking, peripheral control and disk filing systems.

IBM Compatible computers use MS-DOS or, as IBM users often call it, DOS for short. (They don't seem to realise that there are others types of computers!). The MS stands for MicroSoft, the company that produces the MS-DOS system.

The two types of DOS (AmigaDOS and MS-DOS) are not compatible but have many similarities in command names and usage. The comments made above about AmigaDOS could be equally well applied to MS-DOS.

=== End of Text ===

---



### 1.3 AmigaDOS For Beginners - Glossary - ARexx

AMIGADOS FOR BEGINNERS - GLOSSARY

AREXX

ARexx is a programming language.

It is the Amiga version of the IBM language "Rexx".

It is a programming language somewhat similar to BASIC in the way programs are written and executed through an interpreter but quite a bit different to BASIC in its capabilities.

The ARexx manual says:-

"... it can act as a hub through which applications - even those created by different companies - can exchange data and commands".

Thus it can be used to tell one program to do things like pass commands or data to another program, or to receive commands or data from another program. This is known as "Inter Process Communication"

A lot of people only use AREXX for writing small programs to communicate with other programs. However, AREXX can be used for much more than this. It can be used for writing small or large programs which are fully functional in their own right.

ARexx has been supplied as a standard part of the Amiga package from Workbench 2.0 onwards, However, ARexx manuals were only supplied with Workbench 2.0 and 2.1. Workbench 3.0 and 3.1 did not supply ARexx manuals!!

ARexx For Beginners, which has been written along the same lines as "AmigaDOS For Beginners", is available from the same source as "AmigaDOS For Beginners".

=== End of Text ===

### 1.4 AmigaDOS For Beginners - Glossary - Arguments

AMIGADOS FOR BEGINNERS - GLOSSARY

ARGUMENTS

An ARGUMENT is a piece of information entered by the user into a

command line  
that causes the AmigaDOS command to act in a certain way.

---

The contents of the argument can be varied at the discretion of the user, within the guidelines of the syntax of the command, to suit the way the user wants the command to work. Compare this to a

```
COMMAND
  and its
KEYWORDS
  which cannot be varied by the programmer.
```

Examples of arguments are contained in these command lines:-

```
DIR df0:devs
```

"DIR" is the command and "df0:devs" is the argument inserted by the user.

```
FORMAT DRIVE df0: NAME MyDisk
```

"FORMAT" is the command and "DRIVE" and "NAME" are its keywords.  
"df0:" and "MyDisk" are arguments inserted by the user.

=== End of Text ===

## 1.5 AmigaDOS For Beginners - Glossary - ASCII

AMIGADOS FOR BEGINNERS - GLOSSARY

ASCII

ASCII, pronounced "Askey", stands for American Standard Code for Information Interchange. Basically, it is a system of allocating a unique

```
CODE number
  for each character such as 65 for "A", 66 for "B", etc. Most ↔
computers
```

store text files in standard ASCII codes thus making them easily transferrable.

An ASCII file (in word processing or data base terminology) is one saved without all the special codes such as those used by the word processor to indicate bold, underlining and other special features or such as those used by the data base program to indicate column and row information, formulas, etc. Thus the ASCII file contains only the ASCII codes for the text itself.

As well as text, ASCII covers covers codes that are not visible on screen or paper such as "Control Codes".

=== End of Text ===

## 1.6 AmigaDOS For Beginners - Glossary - Booting

## AMIGADOS FOR BEGINNERS - GLOSSARY

## BOOTING

"Booting" is the process of getting the computer up and running.

To achieve this, the system uses:-

Kickstart  
and it must be able to find a bootable disk. To be bootable a floppy disk ↔

must have:-

Boot Block

Startup-Sequence File

A hard drive partition must also have a "Startup-Sequence" file ↔  
but it

does not have a boot block as such. It is made bootable from within the "HDToolBox" program that should be in the "Tools" directory of your workbench disk.

A cold boot is one where the power is turned on after having been turned off.

A warm boot is one where the power is not turned off. The boot is effected by a reset with the CTRL-LeftAmiga-RightAmiga keys.

=== End of Text ===

---

## 1.7 AmigaDOS For Beginners - Glossary - KickStart

AMIGADOS FOR BEGINNERS - GLOSSARY

KICKSTART

"Kickstart" is a small program that enables the  
operating system  
to read

the

boot block

of a disk and get the system up and running. Without  
Kickstart the computer could not read the boot block and therefore could not  
get under way.

In very early Amigas, Kickstart was on a "Kickstart Disk" which you had  
to insert before your Workbench disk. Once the Kickstart software was  
read into

RAM

you could put in you Workbench disk and boot up the system.  
Very awkward system!!

These days Amigas have the Kickstart software hard coded into the  
ROM  
chips.

=== End of Text ===

## 1.8 AmigaDOS For Beginners - Glossary - Boot Block

AMIGADOS FOR BEGINNERS - GLOSSARY

BOOT BLOCK

A "boot block" is a special area on a floppy disk that contains the  
instructions necessary for the computer's

---

operating system  
to be able to  
start up the system from that disk.

A floppy disk without a boot block cannot be used to boot the system.  
However, it also needs a  
Startup-Sequence  
file to properly get the system  
up and running.

A boot block can be put on a disk with the AmigaDOS  
command  
INSTALL.

=== End of Text ===

## 1.9 AmigaDOS For Beginners - Glossary - Startup-Sequence

AMIGADOS FOR BEGINNERS - GLOSSARY

STARTUP-SEQUENCE

The presence of a  
boot block  
(and nothing else) on a disk will do very  
little except put a CLI window onto the screen.

To properly complete the booting process, the bootable disk needs to have  
a file called "startup-sequence" in a directory called "s". This file  
can be customised to make the system boot up in different ways for different  
purposes.

The startup-sequence file contains a series of AmigaDOS commands. The  
one on your boot disk can be read now by [clicking here](#)

=== End of Text ===

## 1.10 AmigaDOS For Beginners - Glossary - Bytes

AMIGADOS FOR BEGINNERS - GLOSSARY

BYTES

A byte is the standard unit of computer memory. Larger units are in multiples  
of bytes, e.g. Kilobytes, Megabytes, etc. (see table below) and smaller  
units are Bits and Nybbles (see table below).

A byte can be a single unit of data. One character of the letter that  
you write will be stored in one byte, or one machine language instruction

---

in a program is stored in one byte.

Other units of memory as related to a byte are:-

BIT	= smallest memory unit
NYBBLE	= 4 bits (2 nybbles = 1 byte)
BYTE	= 8 bits or 2 nybbles
WORD	= 2 bytes or 16 bits
LONGWORD	= 4 bytes or 32 bits
KILOBYTE (kb or k)	= 1024 bytes
MEGABYTE (mb or meg)	= 1024 kilobytes = 1,048,576 bytes
GIGABYTE (gb or gig)	= 1024 megabytes = 1,048,576 kilobytes

The multiplier "1024" is used rather the "1000" because computers think in the

binary  
number system in which:-

binary 100	= decimal 4	(2 power 2)
binary 1000	= decimal 8	(2 power 3)
binary 10000	= decimal 16	(2 power 4)
etc. etc.		
binary 10000000000	= decimal 1024	(2 power 10)

=== End of Text ===

## 1.11 AmigaDOS For Beginners - Glossary - C.H.U.G.

AMIGADOS FOR BEGINNERS - GLOSSARY

COMMODORE HORNSBY USER GROUP INC. (C.H.U.G.)

P.O. BOX 1578 HORNSBY NORTHGATE N.S.W. 2077 AUSTRALIA

C.H.U.G. is a club where owners of Commodore and Amiga computers can get together to help each other to better understanding their computers and to get the most possible out of them.

There are two subgroups - The Amiga Group (all models)  
- The Commodore Group (64 and 128 machines)

Commodore PC (IBM compatible) computers are not supported

FACILITIES

- Club meetings twice a month
- SIG (Special Interest Group) meetings on demand
- Public Domain Software Library (free catalogue disk for members)
- Club Magazine 6 times per year
- Club Newsletter every month
- Magazine Library of commercial publications
- Help for Beginners and others with problems
- Club "shop" (at meetings only) for purchase of disks and some

other  
goods

For more information about C.H.U.G., please write to:-

Commodore Hornsby User Group Inc.  
P.O. Box 1578  
Hornsby Northgate  
N.S.W. 2077  
AUSTRALIA

=== End of Text ===

## 1.12 AmigaDOS For Beginners - Glossary - CLI And Shell

AMIGADOS FOR BEGINNERS - GLOSSARY

CLI AND SHELL

CLI is an acronym for  
Command Line  
Interface. It is a program that displays  
a window into which the user can "Interface" with the Amiga by typing  
"Lines" of "Commands".

Shell is the more recent version of CLI (Command Line Interface). Workbench 1.2 and earlier had only CLI. Workbench 1.3 had the old CLI plus the new CLI enhancements which they called "Shell". In wb1.3 CLI and Shell could both be used individually. From Workbench 2 onwards, the Shell enhancements were incorporated into the standard CLI so that they became one and the same thing.

=== End of Text ===

## 1.13 AmigaDOS For Beginners - Glossary - Commands

AMIGADOS FOR BEGINNERS - GLOSSARY

COMMANDS

A "Command" is one of the AmigaDOS "c" directory programs such as, for example:-

DIR - a program that reads a disk directory  
COPY - a program that copies a file

The command can have  
Keywords  
and  
Options

.  
Commands, keywords and options are more fully discussed in the article  
05.COMMANDS.

=== End of Text ===

## 1.14 AmigaDOS For Beginners - Glossary - Keywords

AMIGADOS FOR BEGINNERS - GLOSSARY

KEYWORDS

A "Keyword" is a special word used in conjunction with  
commands  
to tell

the command to act in a certain way. For example:-

```
COPY FROM name1 TO name2
```

The "FROM" and "TO" are "KEYWORDS" that tell AmigaDOS command COPY that  
name1 is the file to copy FROM and name2 is the file to copy TO.

See also

Options  
.

Commands, keywords and options are more fully discussed in the article  
05.COMMANDS.

=== End of Text ===

## 1.15 AmigaDOS For Beginners - Glossary - Options

AMIGADOS FOR BEGINNERS - GLOSSARY

OPTIONS

An "option" for a  
command  
is a  
keyword  
which can be used or left out at  
the discretion of the user and which can affect the way in which the command  
works.

As an example, the DIR command used without options, as in:-

---



```
> DIR DirectoryName
```

displays the names of the directories and files contained by the directory but not the contents of the sub-directories.

It can have options added as in:-

```
> DIR DirectoryName OPTION
```

For example, the use of the ALL keyword option:-

```
> DIR DirectoryName ALL
```

will display all the contents of the sub-directory. Other options do other things.

Thus an option, while being optional, can change the way the command works.

Commands, keywords and options are more fully discussed in the article 05.COMMANDS.

=== End of Text ===

## 1.16 AmigaDOS For Beginners - Glossary - Commands Lines

### AMIGADOS FOR BEGINNERS - GLOSSARY

#### COMMAND LINES

A "Command Line" is a line entered at the Shell/CLI prompt or contained within a

```
script  
file that contains:-
```

```
Commands  
- essential
```

```
Keywords  
- sometimes essential, sometimes  
optional
```

```
Arguments  
- sometimes essential, sometimes optional
```

A typical command line would be:-

```
COPY FROM df0:s/startup-sequence TO df1:s/startup-sequence
```

with the command and its keywords in upper case and the arguments in lower case.

---

Commands, keywords and options are more fully discussed in the article 05.COMMANDS.

=== End of Text ===

## 1.17 AmigaDOS For Beginners - Glossary - CPU

AMIGADOS FOR BEGINNERS - GLOSSARY

CENTRAL PROCESSING UNIT - CPU

The Central Processing Unit (CPU) is the real brains behind the computer. Among many other things, it is responsible for the execution of all the instructions in a program and for maintaining order between all the various programs that are multitasking at the same time.

The Amiga uses Motorola processors in the 680xx range. For example:-

Amiga 1000/500/600	-	68000
Amiga 1200	-	68020
Amiga 4000/030	-	680EC030
Amiga 4000/040	-	68040

=== End of Text ===

## 1.18 AmigaDOS For Beginners - Glossary - Custom Chips

AMIGADOS FOR BEGINNERS - GLOSSARY

CUSTOM CHIPS

Whereas the

CPU

and other memory chips are not specific to the Amiga, the custom chip set are Amiga specific.

The Amiga's custom chips are the feature that makes the Amiga really stand out as an excellent computer. The CPU passes over the handling of all the graphics and sound to the custom chips leaving the CPU free for other tasks.

These chips have names such as:-

Agnus Alice Amber Buster Denise Gary Lisa

There have been three main versions of the custom chip set:-

---

OCS - Old Chip Set (the original)  
ECS - Enhanced Chip Set  
AGA - Advanced Graphics Architecture

There was to have been an AAA chip set but, with the demise of Commodore, and all the uncertainty of Amiga ownership, it never seemed to have got off the drawing boards.

=== End of Text ===

## 1.19 AmigaDOS For Beginners - Glossary - Devices

AMIGADOS FOR BEGINNERS - GLOSSARY

DEVICES

The Amiga Operating System recognises two types of devices:-

Physical Devices

Logical Devices

Devices are more fully discussed in Part 19 - Devices.

=== End of Text ===

## 1.20 AmigaDOS For Beginners - Glossary - Physical Devices

AMIGADOS FOR BEGINNERS - GLOSSARY

PHYSICAL DEVICES

Physical Devices (as opposed to  
Logical Devices  
) are the physical items

such as:-

Disk Drives  
Printers  
Modems  
Parallel Port (Normally used for printers)  
Serial Port (Normally used for modems but also, rarely, some printers)  
etc. etc.

The term Physical Device also includes some not so obviously physical devices such as

RAM: and RAD:  
drives.

The term "Peripheral" is used to describe physical devices that are external to the computer. For example floppy disk drives, monitors, printers, modems, etc.

=== End of Text ===

## **1.21 AmigaDOS For Beginners - Glossary - Logical Devices**

AMIGADOS FOR BEGINNERS - GLOSSARY

LOGICAL DEVICES AND MAKING ASSIGNMENTS

---

A "logical device" (as opposed to a Physical Device) is one which does not exist in reality but is really a "reference" to a directory on a disk. The system sets up the logical devices either automatically or in the startup-sequence using the ASSIGN command. Users can also use ASSIGN in a Shell/CLI window or a script file.

We talk about this process as "making an assignment" to a directory.

A typical logical device is FONTS: which is a reference to the "Fonts" directory on the disk from which the system was booted. Any program needing fonts knows to look in the logical device "FONTS:". However, the assignment can be changed to any other directory with, for example, :-

```
> ASSIGN Fonts: dfl:MyFonts
```

Programs still look for fonts in the logical device "FONTS:" but now the system directs them to "dfl:MyFonts".

It is much easier to use logical devices such as this as you can change the reference point without having to tell the programs where to find the new directory. They still just look for the assigned logical device.

=== End of Text ===

## 1.22 AmigaDOS For Beginners - Glossary - Directory Utility

### AMIGADOS FOR BEGINNERS - GLOSSARY

#### DIRECTORY UTILITY

A Directory Utility is a program that lets you examine the contents of disks and directories by directory and file name rather than by workbench icons. A good Directory Utility will have two windows side by side in which are shown the contents of two different directories, either on the same or different disks.

The program allows you to see files not normally visible by icons.

It allows you to do a lot of AmigaDOS functions such as copying, deleting, renaming, viewing, etc. without having to learn all the CLI commands. You can, for instance, click on a file name in one window and then click on a button called COPY and the file will be copied from the directory represented in that window to the one represented in the other window.

Modern directory utilities such as "Directory Opus" © Jonathan Potter and GPSoftWare and "DirWork" © Chris Hames (both available through retail outlets) have an extremely large range of capabilities and can be configured to suit you own purposes.

There are also quite a few in the  
Public Domain  
=== End of Text ===

## 1.23 AmigaDOS For Beginners - Glossary - Disks & Drives

### AMIGADOS FOR BEGINNERS - GLOSSARY

#### DISKS & DRIVES

DISK - A storage medium for data. Most disks use a magnetic storage system similar to that for audio and video tapes. Magnetic "read" and "write" heads can write data magnetically to the disk surface and read it back again. Optical reading systems using lasers are also available for "Compact Disks" but at time of writing these are "Read Only".

FLOPPY DISKS for the Amiga are normally 3-1/2" Double Sided Double Density (DSDD) floppy disks with a capacity of up to 880

kilobytes  
of data. Double

sided means that both sides of the disk are used (as opposed to some of the very early disks that only used on side). Double Density refers to the density of the magnetic material on the surface of the disk. It is unnecessary and inadvisable to use 3-1/2" DSHD (Double Sided High Density) disks in the standard Amiga floppy drives. To the best of my knowledge 3-1/2" disks were never made in a density less than "Double".

The Amiga uses the same type of floppy disks as any other computer. However, if you buy pre-formatted disks you will no doubt find that they are in IBM compatible format and you will have to re format them unless you are going to use them with an IBM emulator or with a program that reads & writes IBM formatted disks such as "CrossDOS".

FLOPPY DRIVE is one that, unlike a Hard Drive (see below) allows floppy disks to be inserted and removed and interchanged at the user's will. The standard Amiga Floppy Drives are 3-1/2" drives that take DSDD (Double Sided Double Density) floppy disks (see above).

HARD DRIVE is similar in many ways to a floppy drive (see above) but it is able to store much larger quantities of data and to access the data much faster than a floppy. The advantages are that most, if not all, of your programs and data can be stored on the one disk (depending on the size of the drive) and programs and data will load in a fraction of the time compared to floppies.

Hard drives have capacities ranging from 20  
megabytes  
(mb) upwards.

(The old Amiga A590 hard drive was 20mb.)

---

These days drives of 1  
gigabyte  
and higher are quite common for the home  
user.

Aminet's Australian mirror internet site has a 13 gigabyte drive!!

Most hard drives have the disadvantage that the disk in them is fixed into the drive casing and can not be taken out and replaced as can be done with floppies. However, there are also what is called "removable" hard disks with capacities of, say, 40 mb or 80 mb. These allow one disk or cartridge to be removed and another inserted.

=== End of Text ===

## 1.24 AmigaDOS For Beginners - Glossary - Disk Structure & File Systems

### AMIGADOS FOR BEGINNERS - GLOSSARY

#### DISK STRUCTURE & FILE SYSTEM

The "file system" is the system used by the Amiga's  
operating system  
to

organise files on a disk. The original file system used by Workbench 1.3 and earlier is now called the "Old File System" (OFS). The "Fast File System" (FFS) was introduced with some Hard Disk Drives used by wb1.3 to allow faster access to disk data. With workbench 2 onwards, FFS is the default on Hard Drives and OFS the default on floppies. However, FFS floppy disks can be formatted and read by Workbench 2+. Workbench 1.3 and earlier cannot read FFS floppies. FFS is not only faster but can store more data than OFS (see below)

---

In order to store information on a disk, its magnetic surface is divided into sectors or blocks, tracks and cylinders. These divisions are created when the disk is formatted.

A "Sector" or "Block" is the smallest unit that can be used for data storage. Files are stored in one or more blocks and if the last block is not full then the empty space cannot be used for anything else. A block has a data storage capacity of 488

bytes

in Old File System and 512 bytes in Fast

File System. A disk block is sometimes called a "sector".

(See also

Header Blocks

).

A "Cylinder" consists of a number of blocks. On a standard 3-1/2 inch disk this is 22 blocks, 11 on one side of the disk and 11 on the other side. The 11 blocks on each side are contained in "Tracks"

"Track" is a term that originated from single sided disks. Each side of the disk is divided into a number of concentric circles or tracks each containing a number of blocks.

"Side" or "Head" are also used to describe the two parts of a cylinder, one on each physical side of the disk.

The numbering of sectors, blocks, tracks, cylinders on a 3-1/2 inch disk follows these guidelines:-

Cylinders	- 0 to 79 (80 in total)
Tracks or sides or heads	- 0 and 1 in each cylinder (2 in total)
Blocks	- 0 to 1759 on a disk (1760 in total)
Sectors	- 0 to 10 on a track (11 in total)

Thus block number 1334 is on Sector 3 of Side (or track) 1 of Cylinder 60.

=== End of Text ===

---



## 1.25 AmigaDOS For Beginners - Glossary - Header Block

AMIGADOS FOR BEGINNERS - GLOSSARY

HEADER BLOCKS

Each file stored on a disk has a Header Block created for it. The information in this block is not part of the file but contains important housekeeping information about the file such as:-

- The file name
- Protection bit status
- File length
- The first block number of the file's data
- The number of blocks in the file
- The block number of the file's parent directory
- Any comments written to the file with the FILENOTE command
- The DATE and TIME stamps of the file

=== End of Text ===

## 1.26 AmigaDOS For Beginners - Glossary - Comments & FileNotes

AMIGADOS FOR BEGINNERS - GLOSSARY

COMMENTS & FILENOTES

A "Comment" or "Filenote" is a bit of information that can be attached to a file or directory by the user for his own information or for the information of other users. It does not in any way change the contents of the file.

The FILENOTE command can be used to attach the comment or amend an existing comment.

The LIST command can be used to view the comment.

If the file has an icon, then the comment or filenote can also be both viewed and changed from workbench by using the icon's Information Window. Most good Directory Utility programs also allow these things to be done.

=== End of Text ===

## 1.27 AmigaDOS For Beginners - Glossary - Format & Template

AMIGADOS FOR BEGINNERS - GLOSSARY

FORMAT & TEMPLATE

The

AmigaDOS

manuals all tend to use the FORMAT and TEMPLATE at the start of the

command

description to tell you how to use it. They are two different ways of saying the same thing.

A typical "Format" is:-

```
TYPE {<file|pattern>} [TO <name>] [OPT H|N] [HEX] [NUMBER]
```

and its equivalent "Template" is:-

```
FROM/A/M, TO/K, OPT/K, HEX/S, NUMBER/S
```

These are rather complex for the beginner to understand so I do not normally use them in these articles. You can read Part 63 if you like. However, I would suggest that the beginner leave this until much later.

=== End of Text ===

## 1.28 AmigaDOS For Beginners - Glossary - Information Window

AMIGADOS FOR BEGINNERS - GLOSSARY

INFORMATION WINDOW

The "Information Window" for an icon is obtained by these means:-

1. Select the icon by clicking once on it with the left mouse button.
2. Select one of these Workbench menu items (depending on your version):-

- v1.x - "Info" item in the "Workbench" menu

- v2+ - "Information" item in the "Icons" menu

This window will give you information about the object (disk, directory or file) that the icon represents. The items shown will depend on the type of icon but they will include some or all of:-

- Type of icon (disk, drawer, tool, project or trashcan).

---

Note that "files" are sub-divided into "tools" (programs) and "projects" which can be likened to files created by the tools. Thus a "project" could be a word processor document, a picture, a data base, or anything else that is not, in itself, a program.

-

Protection bit  
status. \*\*

- Date & Time written to disk.

-

Comment or FileNote  
\*\*

- Default Tool \*\* which is the program to be run if the icon represents a project. For example, if the project is a word processor document then the default tool could be the word processor itself so that double clicking the icon would first load the word processor then load the document into the word processor

- Tool Types \*\* which give the tool or the default tool some information about the way the program should operate.

\*\* These items can be altered by the user while the information window is being displayed.

=== End of Text ===

## 1.29 AmigaDOS For Beginners - Glossary - Mounting

AMIGADOS FOR BEGINNERS - GLOSSARY

MOUNTING

MOUNT is an AmigaDOS  
command  
that makes the Amiga recognise a  
device  
that  
is not automatically recognise during  
boot up  
. In other words, it makes  
sure that the device is "connected" to the system.

This process is called Mounting a device

The regular devices such as disk drives, keyboard, monitor are all automatically mounted during boot up.

The Mount List is a text file that has a list of all the devices that can be mounted and has technical information about the device that is used during the mounting process. A "Mount List Entry" is the details

for one individual devices contained within the mount list.

In v2.0 and earlier all the mount list entries were contained within a single mount list file which was stored in the "Devs" directory. Each item to be mounted needed an entry in this list and a corresponding MOUNT command in the

Startup-Sequence  
file.

From v2.1 onwards there can still be a mount list in "Devs" but there are also individual mount files called DOS Drivers held within the "DOSDrivers" directory. Items in this drawer are mounted automatically during boot up with the use of a single MOUNT command in the startup sequence file.

In some contexts, a floppy disk is said to be MOUNTED if it is physically in a disk drive and UNMOUNTED if it is being recognised by the system for some reason but is not currently in any drive.

=== End of Text ===

---

## 1.30 AmigaDOS For Beginners - Glossary - Number Systems

AMIGADOS FOR BEGINNERS - GLOSSARY

### NUMBER SYSTEMS

There are at least four number systems used in the computer world. These are:-

- Binary System  
- Base 2
- Octal System  
- Base 8
- Decimal System  
- Base 10
- Hexadecimal System  
- Base 16

Examples of numbers in each system are contained in this Comparative Table

.

=== End of Text ===

## 1.31 AmigaDOS For Beginners - Glossary - Comparative Table Of Number Systems

AMIGADOS FOR BEGINNERS - GLOSSARY

### COMPARATIVE NUMBERS TABLE

A short table of numbers in all four systems is as follows:-

Binary	Octal	Decimal	Hexadec.
00000001	01	1	01
00000010	02	2	02
00000011	03	3	03
00000100	04	4	04
00000101	05	5	05
00000110	06	6	06
00000111	07	7	07
00001000	10	8	08
00001001	11	9	09
00001010	12	10	0A

00001011	13	11	0B
00001100	14	12	0C
00001101	15	13	0D
00001110	16	14	0E
00001111	17	15	0F
00010000	20	16	10
00010001	21	17	11
.....	..	..	..
00011111	37	31	1F
00100000	40	32	20
.....	..	..	..
00111111	77	63	3F
01000000	100	64	40
.....	..	..	..
.....	..	..	..
11111111	377	255	FF
100000000	400	256	100

=== End of Text ===

## 1.32 AmigaDOS For Beginners - Glossary - Decimal Number System

### AMIGADOS FOR BEGINNERS - GLOSSARY

#### DECIMAL NUMBER SYSTEMS

"Decimal" is one of at four  
number systems  
that are used in computing.

To say a number system has a base ten means that there are 10 digits (0-9)  
that can be used before we start to use double digit numbers.

This system has developed because humans have 10 fingers and primitive  
humans counted on their fingers. Maybe some of us still do! If the powers  
that be had created us with 6 fingers on each hand we would have a number  
system based on 12 digits which would have made arithmetic a lot easier!

=== End of Text ===

## 1.33 AmigaDOS For Beginners - Glossary - Binary Number System

### AMIGADOS FOR BEGINNERS - GLOSSARY

#### BINARY NUMBER SYSTEMS

"Binary" is one of at four  
number systems  
that are used in computing.

Computers, to create an analogy with humans, have only  
one finger  
on each

hand, or a total of two digits. So computers use a Binary System with only  
the two digits of 0 and 1. This is really not a good analogy. A better one  
is to consider that a computer can be compared to millions of little  
switches which can only have two states - on and off:-

- When a switch is off it holds a value of 0
- When a switch is on it holds a value of 1

The switches are the  
bits  
of the computer memory. Eight bits make up one

byte.

Thus a byte can have each of its switches on or off, i.e. holding ↔  
a value

of 0 or 1. The byte can therefore hold numbers of decimal value from 0  
to 255. In the following example the bits are numbered from 0 (rightmost)  
to 7 (leftmost) and the binary numbers are represented by the 0's and  
1's below the bit numbers. For example, decimal 5 is binary 00000101.

Bit number values	Decimal Number
7 6 5 4 3 2 1 0	
0 0 0 0 0 0 0 0	0
0 0 0 0 0 0 0 1	1
0 0 0 0 0 0 1 0	2
0 0 0 0 0 0 1 1	3
0 0 0 0 0 1 0 0	4
0 0 0 0 0 1 0 1	5
0 0 0 0 0 1 1 0	6
0 0 0 0 0 1 1 1	7
0 0 0 0 1 0 0 0	8
.....	..
0 0 0 0 1 1 1 1	15
0 0 0 1 0 0 0 0	16
.....	..
.....	...
1 1 1 1 1 1 1 1	255

Thus the binary number, which is represented by the 0's and 1's in the  
above table, is directly related to the on and off condition of each bit  
of a byte. See

Comparative Table  
for more examples.

=== End of Text ===

## 1.34 AmigaDOS For Beginners - Glossary - Octal Number System

AMIGADOS FOR BEGINNERS - GLOSSARY

OCTAL NUMBER SYSTEMS

"Octal" is one of at four  
number systems  
that are used in computing.

The Octal system is not nearly as commonly referred to as the other three.

It uses only 8 digits, i.e. 0 to 7.

See

Comparative Table  
for examples.

=== End of Text ===

## 1.35 AmigaDOS For Beginners - Glossary - Hexadecimal Number System

AMIGADOS FOR BEGINNERS - GLOSSARY

HEXADECIMAL NUMBER SYSTEMS

"Hexadecimal" is one of at four  
number systems  
that are used in computing.

Hexadecimal is extensively used because the  
binary system  
(the computer's  
"native language") is too hard for humans to work with.

Hexadecimal relates more closely to binary than does decimal but is easier to understand than is binary.

Hexadecimal uses 16 digits for the decimal numbers 0 to 15. It does not start to get into double digit numbers until the decimal number 16. Because we only have 10 digits at our disposal, the letter A - F are used for the extra 6 digits and represent the decimal number 10 to 15 respectively.



See

Comparative Table  
for examples.

=== End of Text ===

## 1.36 AmigaDOS For Beginners - Glossary - Operating System

### AMIGADOS FOR BEGINNERS - GLOSSARY

#### OPERATING SYSTEM

##### WHAT IS AN OPERATING SYSTEM?

Every computer needs some sort of operating system. Without one, the computer would not know how to make a program work. In fact, it would not even know how to load a program from disk into memory. It would not know how to put graphics onto the screen. It would not know how to interpret keyboard presses or mouse movements. It would know NOTHING at all!

An operating system is really a whole set of instructions, or programs, and other types of information, to tell the computer how to do all the things that let you load programs and communicate with them via the keyboard and mouse, and to let you see the output on the screen.

There are two parts of the operating systems - disk based and ROM based.

##### The ROM Based Part Of The Operating System

First of all, there is the  
ROM  
based part of the Operating System. Let's  
call this the ROM-OS for short.

The ROM-OS is "hard coded" into the computer's ROM chips. By "hard coded" I mean that it is a set of instructions that are permanently written onto the various ROM chips. Being in ROM it cannot be altered unless a new ROM chip is installed. For example, if you want to update your Amiga from Version 1.3 to the Version 2 then you will need to have a new ROM chip put into your computer.

Because ROM cannot loose its memory when power is turned off, this part of operating system is always there waiting for you to use it.

The ROM-OS allows the computer to get started. It can read information from disk, load it into  
RAM,  
make a program operate properly, display  
information on the screen, and lots more.

##### The Disk Based Part Of The Operating System

Secondly, there is the Disk Based part of the Operating System.

---

This is what used to be called DOS for "Disk (based) Operating System". (In IBM compatible computers, it is called MS-DOS and in the Amiga it is called AmigaDOS although

AmigaDOS

is a lot more than just the disk based

commands.)

Some people call it "Disk Operating System". This is sometimes taken to mean a system for operating the disk drive and talking to the disk. While this is almost true for some computers such as the Commodore 64, it is far from being the whole truth for the Amiga. AmigaDOS includes more than just systems for operating the disk drive.

Originally, AmigaDOS referred to a whole lot of programs and information that is held on a disk. However, with Versions 2 and 3, a lot of things that used to be on disk are now held in ROM. Thus some AmigaDOS

commands

can no longer be found on disk.

These days "Disk Based" is no longer accurate but DOS is still used to mean the whole set of commands and files that the user can access to make the system operate.

For those commands still on disk, individual items can be loaded from disk into RAM as and when they are required. When they are finished with, the system can take them out of memory thus freeing up RAM for other activities.

As an example, when you copy a disk, a program called "DiskCopy" is loaded into RAM from the Workbench disk, the disk copying is carried out by that program, and then "DiskCopy" is removed from RAM. Next time you copy a disk, the "DiskCopy" program must be reloaded from the workbench disk.

Why Two Parts Of The Operating Systems?

Although the distinction between ROM based and DISK based has become somewhat blurred since Version 2, the following notes are still relevant.

The main difference between the two parts of the operating systems is that one is ROM based and one is DISK based. ROM-OS is in ROM memory and DOS is loaded into RAM memory. However, once it is loaded into memory, the Amiga "sees" the DOS command in the same way as it "sees" ROM-OS, i.e. it is just another set of instructions and/or information for the Amiga to use.

There are two main reasons for having two parts to the overall operating system.

Firstly, operating systems have become so large that to have it all hard coded into ROM would mean a lot of expensive memory tied up with a lot items that, normally, are used only one at a time. (For those who had or know the Commodore 64 computer, it DID have the complete operating system hard coded in ROM chips and discs were only used for programs and user data.)

In some cases, parts of the DOS are rarely, if ever, used by many people. It is much more effective to have a lot of the system on a disk ready to be

---

called up if and when required.

A disk is a much cheaper storage medium than RAM or ROM chips. A 3-1/2 inch disk that costs about \$1 can hold 880 kilobytes of information. This is more than the 512 kilobytes that comes with an unexpanded Amiga 500! Extra ROM chips equivalent to the capacity of a single disk would certainly cost far more than \$1.00!

Secondly, computer programs, including operating systems, have a habit of being continually upgraded. It is much easier and cheaper to upgrade your computer to the latest operating system by buying it on disk rather than having to have the ROM chips changed. (However, the upgrade to Version 2 involves both ROM-OS and DOS. Thus to get it you need a new ROM chip as well as new disks.)

For both these reasons, computer manufacturers put a lot of the operating system onto disk and call it the Disk BASED Operating System, i.e. DOS.

To put it VERY simply, DOS will hold those things not needed for basic operations and that may only be infrequently used, whereas the ROM-OS will contain those things needed to get the computer up and running, and to handle a lot of the facilities that are used all the time. Keyboard and mouse input and screen output would be two of the most common that are in ROM-OS. These sorts of things will operate far more efficiently from ROM-OS than from DOS.

=== End of Text ===

## 1.37 AmigaDOS For Beginners - Glossary - Public Domain

AMIGADOS FOR BEGINNERS - GLOSSARY

PUBLIC DOMAIN

Public Domain (P.D.) is software that can be legally copied and distributed freely by anybody. Most early P.D. had no copyright on it but nowadays a lot of it is copyrighted but has a notice indicating that it may be freely copied and distributed under certain conditions.

Freeware is another name for Public Domain.

Shareware, while often included in the P.D. category, is not strictly P.D. It works under an honour system. Normally, it can be freely copied and used for a short trial period. If you intend to continue using it then, legally, you must pay the author the requested shareware fee. If you do not pay the shareware fee then you are supposed to discontinue using it. Sometimes some key part of the program is missing which is provided when the fee is paid. Other means of encouraging payment are also employed.

=== End of Text ===

---

## 1.38 AmigaDOS For Beginners - Glossary - Protection Bits

AMIGADOS FOR BEGINNERS - GLOSSARY

### PROTECTION BITS

Every file on a disk has "Protection Bits" or, in MS-DOS terminology,

"Attributes" which tell the system certain things about the file. (This is one case where I think MS-DOS terminology is better - "Protection Bit" and PROTECT are a bit misleading as a lot of the bits have nothing to do with protecting the file!

These bits can be turned "on" or "off" thus changing the attributes of the file.

The protection bits that are available in all versions of AmigaDOS are:-

- r To indicate if the file is readable or not
  - w To indicate if the file can be written to or not
  - e To indicate if the file can be executed or not
-

d To indicate if the file can be deleted or not

These extra protection bits became available from v1.3 onwards:-

h To indicate that the file should be hidden

s To indicate if the file is a script or not

p To indicate if file is pure or not (explained in article RESIDENT)

a To indicate if the file has been archived or not

By changing the protection bit you can change how the system treats the file. For example, the "d" bit, if "turned off" changes the file from a deletable one to a non deletable one.

Protection bits can be displayed by the LIST command.

Protection bits can be changed with the PROTECT command.

They can also be viewed and changed from Workbench by using the icon's

Information Window  
or with a good  
Directory Utility  
program.

=== End of Text ===

## 1.39 AmigaDOS For Beginners - Glossary - Memory RAM & ROM

AMIGADOS FOR BEGINNERS - GLOSSARY

MEMORY - ROM AND RAM

Computer memory is the internal chips that are used to hold programming code and data.

Computers have two main types of memory. These are:-

ROM - READ ONLY MEMORY

RAM - RANDOM ACCESS MEMORY

In the Amiga, RAM is divided into "CHIP" and "FAST" memory.

CHIP RAM (Graphics Memory)

FAST RAM

SLOW RAM or SLOW-FAST RAM

=== End of Text ===

---

## 1.40 AmigaDOS For Beginners - Glossary - ROM

AMIGADOS FOR BEGINNERS - GLOSSARY

ROM - READ ONLY MEMORY

"Read Only Memory" (ROM) is  
     memory  
     that the computer can read information  
 from but, as the name implies, it cannot write information to it.

Its information is permanent, i.e. it is not lost when the power is turned off. Compare this with

    RAM  
     which can be written to as well as read from,  
 but which loses its memory when the power is turned off. A lot of the  
 Amiga's  
     operating system  
     is held in the ROM chips.

=== End of Text ===

## 1.41 AmigaDOS For Beginners - Glossary - RAM

AMIGADOS FOR BEGINNERS - GLOSSARY

RAM - RANDOM ACCESS MEMORY

"Random Access Memory" is  
     memory  
     that the computer can both read from and  
 write to. (Compare this with  
     ROM  
     which can be read from but not written to).

RAM loses its memory when the power is turned off.

In the Amiga, RAM can be one of:-

    CHIP

    FAST

    SLOW-FAST

    The term "Random Access" derives from the fact that the computer ↔  
     can gain  
 direct access to any part of the memory (start, middle or end) at will,  
 or randomly. Compare this to tape storage where the tape has to be  
 physically wound to the required spot before access can be gained.

The capacity of a computer is usually measured in the number of

---

kilobytes or megabytes  
of RAM memory.

It is in RAM that all the programs that you load from disk are placed. All your word processor documents and you paint program pictures, etc. are temporarily stored in RAM while you are using them.

=== End of Text ===

## 1.42 AmigaDOS For Beginners - Glossary - CHIP

AMIGADOS FOR BEGINNERS - GLOSSARY

CHIP RAM or CHIP MEMORY - Also called "Graphics Memory"

This is called "CHIP RAM" because it is  
memory  
accessible by the Amiga's

Custom Chips  
(the chips specially designed for the Amiga) as well as  
the  
CPU

.

These custom chips cannot access any other type of ram but chip ram.

Chip RAM is used to store data for graphics and sound and is sometimes called Graphics Memory but can also be used for normal program or data storage if there is no  
fast ram  
available, i.e. if fast ram is already full  
or if the computer is not equipped with fast ram.

The original Amiga 1000 and 500 computers could only access 1/2 megabyte (512 kilobytes) of chip ram. Later Amigas allowed access to 1 megabyte, then 2 megabyte of chip ram thus allowing bigger and better graphics and sound.

=== End of Text ===

## 1.43 AmigaDOS For Beginners - Glossary - FAST

AMIGADOS FOR BEGINNERS - GLOSSARY

FAST RAM

---

Fast RAM is

memory  
 that is only accessible by the  
 CPU  
 and not by the  
 custom  
 chips. It is called "FAST RAM" as programs stored in fast RAM run  
 with  
 more speed than those in stored  
 CHIP RAM  
 .

Many Amigas do not have fast ram in the standard model, but only chip ram.  
 Fast ram is an optional extra with such Amigas.

Memory added as expansion RAM is usually fast ram, not chip ram (but see  
 also

SLOW RAM  
 which discusses "Slow-Fast RAM").

=== End of Text ===

## 1.44 AmigaDOS For Beginners - Glossary - SLOW

AMIGADOS FOR BEGINNERS - GLOSSARY

SLOW RAM or SLOW-FAST RAM

Slow RAM is the expansion  
 memory  
 used by the Amiga 500 expansions that  
 fit into the trap door connector. It is recognised as  
 fast ram  
 by the  
 system (i.e. memory used by the processor but not by the custom chips)  
 but it does not operate as fast as true "fast ram". Hence the term  
 "slow-fast ram".

=== End of Text ===

## 1.45 AmigaDOS For Beginners - Glossary - RAM & RAD Disks

AMIGADOS FOR BEGINNERS - GLOSSARY

RAM DISK & RAD DISK

A "RAM DISK" is an area of RAM memory that is set up to act just like a  
 physical disk. It has a disk icon that can be opened. It can contain

---



drawers and files. You can copy files to it. The Startup-Sequence file usually creates a "T" directory in the Ram Disk for programs to use as temporary storage. All contents of a RAM Disk are lost on power loss or on reset.

A "RAD DISK" is another type of Ram Disk. RAD is a "Recoverable RAM Disk". It still loses its contents on loss of power but it retains them on reset. It can also be set up as a bootable disk from which you can boot during a

warm boot  
, or reset.

=== End of Text ===

## 1.46 AmigaDOS For Beginners - Glossary - Scripts

AMIGADOS FOR BEGINNERS - GLOSSARY

SCRIPTS

A script file, in the AmigaDOS context, is a file containing one or more command lines.

Scripts are written in plain text which makes them quite readable (once you understand the command syntax!) and they can be created with any text editor. They are normally stored in the "S" directory.

The startup-sequence file is an example of a script file.

=== End of Text ===

## 1.47 AmigaDOS For Beginners - Glossary - Stack

AMIGADOS FOR BEGINNERS - GLOSSARY

STACK

The "STACK" is an area of memory set aside for the use of a program for temporary storage and for the program's own internal "housekeeping". Each program creates its own stack in memory.

Different programs need different stack sizes. Although the vast majority

---

of users will never have to worry about it, the stack size for a program can be altered through the icon's

Information Window.

The stack size for a CLI process can be altered with the STACK ← command.

=== End of Text ===

## 1.48 AmigaDOS For Beginners - Glossary - Strings

AMIGADOS FOR BEGINNERS - GLOSSARY

STRINGS

To put it rather simply, a "string" is a single unit of data that contains one or more characters. Just as a whole lot of beads can be put into a "string of beads" so can a whole lot of characters be put together into a "string of characters". The thing about strings, as the computer sees them, is that the computer considers the string of characters as a single unit of data.

There are times when a string must be contained within "string delimiters" which are simply characters that define the limits, i.e. the start and finish, of the string. In AmigaDOS the string delimiter character is the inverted comma (").

If there are no spaces in the string then you often do not need to use quotation marks. For example:-

```
TYPE MyFile
```

will consider "MyFile" as the string representing the file name to be typed.

However, :-

```
TYPE My File
```

will consider the "My" and "File" as two separate strings representing two different file names. You would have to use:-

```
TYPE "My File"
```

to make AmigaDOS look at "My File" as a single string representing one file name only.

A string can contain any of the characters that can be entered via the keyboard (as well as some that can't!!). For example, all these are strings:-

```
"A String is a lot of characters put together"
```

```
"!@#%$^&* ()"
```

```
" A1 b2 C3 "
```

```
" 12 + 3 is Fifteen"
" : ; / ? . < "
```

Note that strings can contain the space character so that a single string can be made up of a number of words separated by spaces.

Note that the string does not include the string delimiter characters so that, if the string is represented by:-

```
"A String is a lot of characters put together"
```

The actual string is:-

```
A String is a lot of characters put together
```

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

## 1.49 AmigaDOS For Beginners - Glossary - Viruses

### AMIGADOS FOR BEGINNERS - GLOSSARY

#### VIRUSES

A virus is a small program that has been written to cause problems to computer users.

Viruses can be:-

```
Boot block viruses - they are in a disk's
    Boot Block
    File viruses      - they are attached to disk files such as
    AmigaDOS
    commands
    .
```

Viruses can be "benign" and only cause minor problems or silly messages, or they can be "malignant" and cause considerable damage to the data on your disks. You could find that ALL data has been completely wiped from your hard drive. They cannot do physical damage to the computer.

A decent Virus Checking and Killing program to detect and remove viruses is essential!. I use one or two

```
shareware
programs which are:-
```

- "Virus\_CheckerII" by Alex van Niel
- "VirusZ" by Georg Hörmann

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

## 1.50 AmigaDOS For Beginners - Glossary - WildCards

AMIGADOS FOR BEGINNERS - GLOSSARY

WILDCARDS

A wildcard is a character that can be inserted into a file name to replace other characters when searching for a file name on a disk.

The term wildcard is taken from card games where the joker, or other cards, can often be used in lieu of other cards.

In AmigaDOS files, the ? character can replace any other character and the # character is used to say "any number of the characters that follow me". Thus #? means any number of any characters.

The \* character is used by some programs as a substitute for #?

The use of Wildcards is fully explained in the article PATTERN MATCHING.

=== End of Text ===

## 1.51 AmigaDOS For Beginners - Table Of ASCII Codes

AMIGADOS FOR BEGINNERS - GLOSSARY

TABLE OF ASCII CODES

The standard set of ASCII code numbers range from 0 to 127.

They are standard as they are used by most computers including Amiga and IBM compatibles. (For those of you who were once Commodore Pet, 64 or 128 users, they are not the same as the so called "Pet ASCII" used by those machines.)

Non standard codes range from 128 to 255.

These codes are dealt with below under three headings:-

Non-Printable ASCII Code Numbers

- 0 to 31

Printable ASCII Code Numbers

- 32 to 127

Non-Standard Code Numbers

- 128 to 255

For the most part, beginners in AmigaDOS can forget about what all the codes are. The only reason that I am including these lists here is in case anyone is interested or in case they do have a need to know one or two of them.

---

=== End of Text ===

## 1.52 AmigaDOS For Beginners - Table Of ASCII Codes - Printable

Printable ASCII Code Numbers - 32 to 127

These are the code numbers that (mostly) produce readable characters when sent to the printer or screen.

Two of the codes are not really readable but are included as they have an effect on what you can read. They are:-

- the "space" (32) character which could be considered readable if you can see the space between two other characters.
- the "delete" code (127) that is definitely not readable but can affect the readable characters.

Code Number			Code Number			Code Number		
Dec	Hex	Character	Dec	Hex	Character	Dec	Hex	Character
32	20	Space	64	40	@	96	60	`
33	21	!	65	41	A	97	61	a
34	22	"	66	42	B	98	62	b
35	23	#	67	43	C	99	63	c
36	24	\$	68	44	D	100	64	d
37	25	%	69	45	E	101	65	e
38	26	&	70	46	F	102	66	f
39	27	'	71	47	G	103	67	g
40	28	(	72	48	H	104	68	h
41	29	)	73	49	I	105	69	i
42	2A	*	74	4A	J	106	6A	j
43	2B	+	75	4B	K	107	6B	k
44	2C	,	76	4C	L	108	6C	l
45	2D	-	77	4D	M	109	6D	m
46	2E	.	78	4E	N	110	6E	n
47	2F	/	79	4F	O	111	6F	o
48	30	0	80	50	P	112	70	p
49	31	1	81	51	Q	113	71	q
50	32	2	82	52	R	114	72	r
51	33	3	83	53	S	115	73	s
52	34	4	84	54	T	116	74	t
53	35	5	85	55	U	117	75	u
54	36	6	86	56	V	118	76	v
55	37	7	87	57	W	119	77	w
56	38	8	88	58	X	120	78	x
57	39	9	89	59	Y	121	79	y
58	3A	:	90	5A	Z	122	7A	z
59	3B	;	91	5B	[	123	7B	{
60	3C	<	92	5C	\	124	7C	
61	3D	=	93	5D	]	125	7D	}
62	3E	>	94	5E	^	126	7E	~
63	3F	?	95	5F	_	127	7F	DELETE

=== End of Text ===

## 1.53 AmigaDOS For Beginners - Table Of ASCII Codes - Non Printable

Non-Printable ASCII Code Numbers - 0 to 31

These are the code numbers that do not produce characters when sent to printer or screen but that carry out various functions.

Code Number				Code Number			
Dec	Hex	Char	Meaning	Dec	Hex	Char	Meaning
0	00	NUL	Null	16	10	DLE	Data Link Escape
1	01	SOH	Start of Heading	17	11	DC1	Direct Control 1
2	02	STX	Start of Text	18	12	DC2	Direct Control 2
3	03	ETX	End of Text	19	13	DC3	Direct Control 3
4	04	EOT	End Transmission	20	14	DC4	Direct Control 4
5	05	ENQ	Enquiry	21	15	NAK	Negative Acknowledge
6	06	ACK	Acknowledge	22	16	SYN	Synchronous Idle
7	07	BEL	Bell	23	17	ETB	End Transmission Block
8	08	BS	Back space	24	18	CAN	Cancel
9	09	HT	Horizontal Tab	25	19	EM	End of Medium
10	0A	LF	Line Feed	26	1A	SUB	Substitute
11	0B	VT	Vertical Tab	27	1B	ESC	Escape
12	0C	FF	Form Feed	28	1C	FS	Form Separator
13	0D	CR	Carriage Return	29	1D	GS	Group Separator
14	0E	SO	Shift Out	30	1E	RS	Record Separator
15	0F	SI	Shift In	31	1F	US	Unit Separator

This lot of seemingly gobbledegook is a list of codes that are mainly used for the control of the transmission of data between the computer and its peripherals, e.g. printer, modem, disk drive, etc.

=== End of Text ===

## 1.54 AmigaDOS For Beginners - Table Of ASCII Codes - Non Standard

Non-Standard Code Numbers - 128 to 255

The code numbers from 128 to 255 are not normally included in the "true" set of ASCII numbers. However, many fonts use these numbers for special characters that are generally produced when you press keys in conjunction with the ALT and the SHIFT/ALT control keys.

These code numbers are described as Non-ASCII as they have not been standardised. The characters (if any) that they represent may vary from computer to computer (e.g. from Amiga to IBM) and even from font to font for the same computer

The list shown below covers code numbers 160 to 255. My setup does not give any characters for codes 128 to 159.

If you do not see any characters under the "Character" headings below, or if some are missing, then the font you are using is different to mine in this range of code numbers.

Do not take this list as definitive. I have included it here only to show you the type of characters produced by the codes in the range 128-255. As indicated above, different fonts and different computers may produce a completely different set of characters.

Code Number			Code Number			Code Number		
Dec	Hex	Character	Dec	Hex	Character	Dec	Hex	Character
160	A0	Space	192	C0	À	224	E0	à
161	A1	ı	193	C1	Á	225	E1	á
162	A2	ç	194	C2	Â	226	E2	â
163	A3	£	195	C3	Ã	227	E3	ã
164	A4	¤	196	C4	Ä	228	E4	ä
165	A5	\$\yen\$	197	C5	Å	229	E5	å
166	A6		198	C6	Æ	230	E6	æ
167	A7	\$	199	C7	Ç	231	E7	ç
168	A8	¨	200	C8	È	232	E8	è
169	A9	©	201	C9	É	233	E9	é
170	AA	ª	202	CA	Ê	234	EA	ê
171	AB	«	203	CB	Ë	235	EB	ë
172	AC	\ensuremath{\lnot}	204	CC	Ì	236	EC	ì
173	AD		205	CD	Í	237	ED	í
174	AE	®	206	CE	Î	238	EE	î
175	AF	-	207	CF	Ï	239	EF	ï
176	B0	\textdegree{}	208	D0	Ð	240	F0	ð
177	B1	\ensuremath{\pm}	209	D1	Ñ	241	F1	ñ
178	B2	\$^2\$	210	D2	Ò	242	F2	ò
179	B3	\$^3\$	211	D3	Ó	243	F3	ó
180	B4	'	212	D4	Ô	244	F4	ô
181	B5	\$\mathrm{\mu}\$	213	D5	Õ	245	F5	õ
182	B6	¶	214	D6	Ö	246	F6	ö
183	B7	·	215	D7	\$\times\$	247	F7	\$\div\$
184	B8	,	216	D8	Ø	248	F8	ø
185	B9	\$^1\$	217	D9	Ù	249	F9	ù
186	BA	°	218	DA	Ú	250	FA	ú
187	BB	»	219	DB	Û	251	FB	û
188	BC	¼	220	DC	Ü	252	FC	ü
189	BD	½	221	DD	Ý	253	FD	ý
190	BE	¾	222	DE	Þ	254	FE	þ
191	BF	¿	223	DF	ß	255	FF	ÿ

=== End of Text ===