

ALAN FREEDMAN

# The Computer Desktop Encyclopedia

**amacom**

**American Management Association**

New York • Atlanta • Boston • Chicago • Kansas City • San Francisco • Washington, D.C.  
Brussels • Mexico City • Tokyo • Toronto

This book is available at a special discount when ordered in bulk quantities. For information, contact Special Sales Department, AMACOM, a division of American Management Association, 1601 Broadway, New York, NY 10019.

This publication is designed to provide accurate and authoritative information in regard to the subject matter covered. It is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional service. If legal advice or other expert assistance is required, the services of a competent professional person should be sought.

Library of Congress Cataloging-in-Publication Data

Freedman, Alan

Computer desktop encyclopedia / Alan Freedman.

p. cm.

ISBN 0-8144-0012-4 (CD-ROM). — ISBN 0-8144-0010-8 (pbk.)

1. Computers—Dictionaries. I. Title.

QA76.15.F732 1996

004'.03—dc20

96-724

CIP

© 1996 The Computer Language Company Inc., Point Pleasant, PA 18950.

All rights reserved.

Printed in the United States of America.

The publication may not be reproduced, stored in a retrieval system, or transmitted in whole or in part, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of AMACOM, a division of American Management Association, 1601 Broadway, New York, NY 10019.

Printing number

10 9 8 7 6 5 4 3 2 1

(2) (Stacker) A realtime compression program from Stac Electronics, Carlsbad, CA, that doubles the disk capacity of a PC. A coprocessor board is optionally available.

### **stack overflow**

An error condition that occurs when there is no room in the stack for a new item. Contrast with *stack underflow*.

In DOS, a Stack overflow error message means that there is not enough room on the stack to handle hardware interrupts. Increase the number of stacks in the STACKS= command in the CONFIG.SYS file.

You can also get this message when other things go haywire; for example, a bad expansion board or one that isn't seated properly in the slot can cause erratic signals eventually leading to this message.

### **stack pointer**

An address that identifies the location of the most recent item placed on the stack.

### **stack underflow**

An error condition that occurs when an item is called for from the stack, but the stack is empty. Contrast with *stack overflow*.

### **stackware**

A HyperCard application that is made up of a HyperCard stack (data) and HyperTalk programming.

### **STAIRS**

(STorage And Information Retrieval System) An IBM text document management system for mainframes. It allows users to search for documents based on key words or word combinations.

### **standard**

A specification for hardware or software that is either widely used and accepted (de facto) or is sanctioned by a standards organization (de jure). See *standards & compatibility*.

### **standard cell**

The finished design of an electronic function ready for chip fabrication. It can be as small as a clock circuit or as large as a microprocessor. It is used to make custom-designed chips.

### **standard deviation**

In statistics, the average amount a number varies from the average number in a series of numbers.

### **Standard Mode**

Windows operation mode. See *Windows*.

### **standards & compatibility**

Standards is the most important issue in the computer field. As an unregulated industry, we have wound up with thousands of data formats and languages, but very few standards that are universally used. This subject is as heated as politics and religion to vendors and industry planners.

No matter how much the industry talks about compatibility, new formats and languages appear routinely. The standards makers are always trying to cast a standard in concrete, while the innovators are trying to create a new one. Even when standards are created, they are violated as soon as a new feature is added by the vendor.

If a format or language is used extensively and others copy it, it becomes a de facto standard and may become as widely used as official standards from ANSI and IEEE. When de facto standards are sanctioned by these organizations, they become stable, at least, for a while.

In order to truly understand this industry, it is essential to understand the categories for which standards are created.

### Machine Languages

Machine language is the fundamental standard for hardware compatibility. It is the language the CPU understands. All programs that are presented to the computer for execution must be in the machine language of that particular CPU family.

Vendors may produce systems with different machine languages. IBM's mainframe family differs from its AS/400 midrange series, which is also different than its RS/6000 series. Digital's newer Alpha line of computers does not understand the machine language of the many earlier VAX computers that have been installed. For all the major hardware platforms, see *hardware platforms*.

After a program is written, it must be translated (assembled, compiled or interpreted) into the machine language the computer understands. In order to run in a different machine, the program must be reassembled or recompiled into a different machine language.

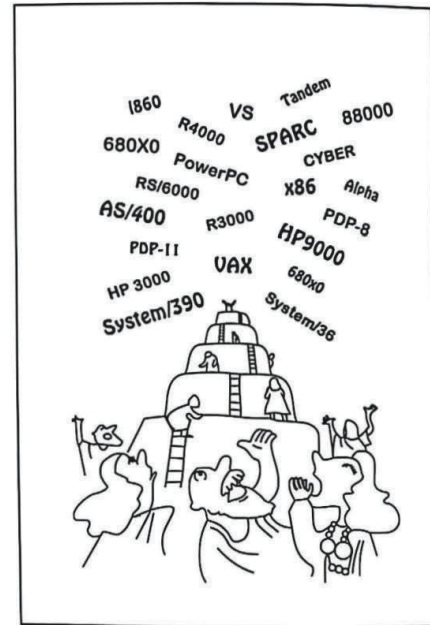
Since the late 1960s, companies seeking a chunk of the IBM market have designed computers that run the same machine language as the IBM mainframes. RCA's Spectra 70 was the first IBM-compatible mainframe, and companies, such as Amdahl, Intel, National Advanced Systems, Hitachi and Fujitsu have introduced IBM-compatible mainframes at one time or another. IBM PC machine language compatibility is achieved by using a processor from Intel's x86 family of microprocessors or one of the Intel clone chips.

Machine language compatibility can also be achieved by emulation. An emulator is software (or hardware or both) that executes the machine language of another computer directly. With DOS and Windows emulation in a UNIX workstation or a PowerMac, users can run non-native programs in their computer.

Emulation goes back to the 1950s and 1960s when IBM built a 1401 emulator in its System/360 to ease migration from the very popular 1401s to the new 360s. The terms simulator and emulator are used interchangeably.

### Data Codes/ characters Codes

The data code built into the computer determines how each letter, digit or special character (\$, %, #, etc.) is represented in binary code. Fortunately, there are only two methods in wide use: EBCDIC and ASCII. IBM minis and mainframes use EBCDIC. ASCII is used for most everything else, including all PCs and Macs. ASCII is a 7-bit code placed into an 8-bit storage unit. The seven bits provide the basic set of 128 ASCII characters. The 8th bit adds storage for another 128 symbols, and these symbols vary from font to font and system to system. For example, the DOS character set contains line drawing and foreign language characters. The ANSI character set places the foreign language characters in different locations. In the Mac, the upper 128 characters can be custom drawn.



When systems are moved from one computer to another, converting between ASCII and EBCDIC is just a small part of the data conversion process. It is done in conjunction with converting file formats from the old to the new systems. The following is a sample of ASCII and EBCDIC code. See *ASCII chart*, *hex chart* and *EBCDIC chart*.

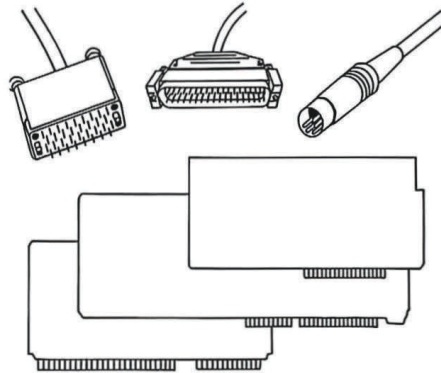
| Character | ASCII    | EBCDIC   |
|-----------|----------|----------|
| space     | 01000000 | 00100000 |
| period    | 01001011 | 00101110 |
| < sign    | 01001100 | 00111100 |
| + sign    | 01001110 | 00101011 |
| \$ sign   | 01011011 | 00100100 |
| A         | 11000001 | 01000001 |
| B         | 11000010 | 01000010 |

## Hardware Interfaces

The hardware interface specifies the plugs, sockets, cables and electrical signals that pass through each line between the CPU and a peripheral device or communications network.

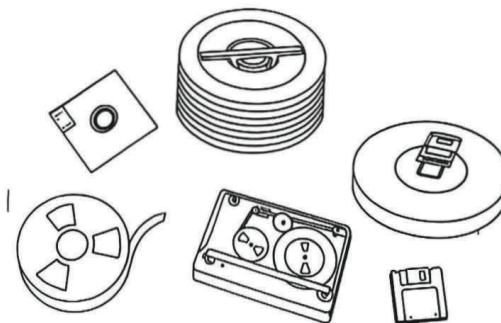
Common hardware interfaces for personal computers are the Centronics parallel interface used for printers and the RS-232 interface, typically used for modems, graphics tablets, mice and printers. In addition, the IDE and SCSI interfaces are commonly used for disks and tapes, and the GPIB IEEE 488 standard is used for process control instruments.

The bus in a computer's motherboard, into which additional printed circuit boards are inserted, is a hardware interface. For example, the Micro Channel in IBM's PS/2 series accepts a physically different board than the original PC bus. The PCI bus and the VL-bus also have different pin configurations and are not interchangeable. LANs, such as Ethernet and Token Ring, dictate the hardware interface as part of their specifications. An Ethernet cable cannot connect to a Token Ring board, and vice versa.



## Storage Media

There are many varieties of disk cartridges, floppy disks, reel-to-reel tapes, tape cartridges and tape cassettes. Each one has its own unique shape and size and can be used only in drives designed to accommodate them. With removable media, the physical standard is half the compatibility issue. The other half is the recording pattern, which is invisible to the human eye. Magnetic tapes and disks fresh out of the box are blank. The



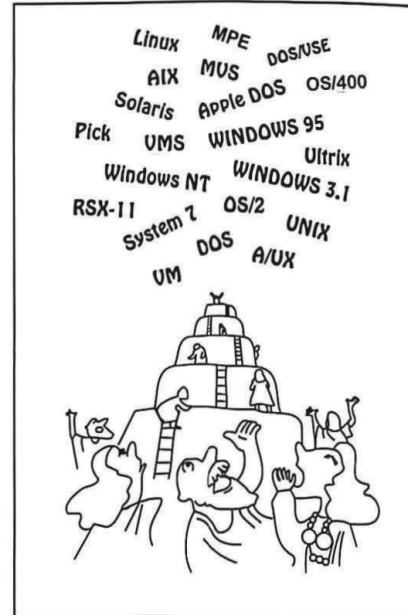
recording tracks are placed onto the surface by the read/write head of the storage drive. Thus, the same floppy disk that stores 720K bytes in one disk drive, can hold 800K if formatted for another. If the computer reads an incompatible tape or reads and writes an incompatible disk, it will signal a read/write error.

For minicomputers and mainframes, half-inch magnetic tape reels and cartridges are the common interchangeable medium. For personal computers, the 3.5" floppy disk is the standard.

### Operating Systems

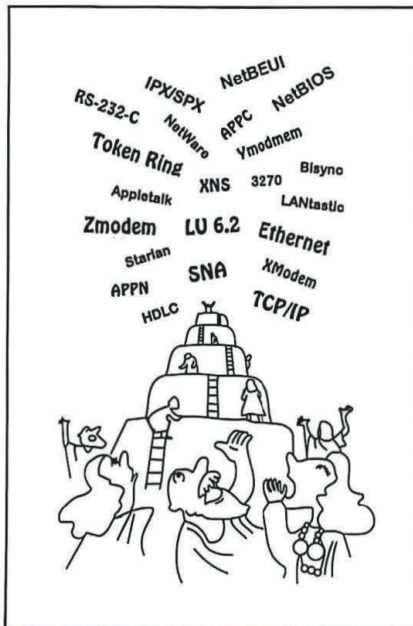
An operating system is a master control program that manages the running of the computer system. In all environments, except for specialized scientific and process control applications, the operating system interacts with the application programs. The application programs must "talk" to the operating system.

If application programs are moved to a different computing environment, they have to be converted to interface with a different operating system. If a new operating system is installed that is not compatible with the old one, the application programs have to be converted to the new operating system.



### Communications & Networking

Transmitting between two personal computers is relatively simple. All that's required is a modem and communications program in each computer that uses the same file transfer protocol. Most comm programs support several protocols and are widely used to upload and download files and gain access to a BBS or online service. Connecting to a major online service, such as America Online or CompuServe, is even more straightforward when you use the service's front-end software.



Communications within the enterprise is far more complex. Over the years, organizations have developed islands of computer systems, each with their own networking protocols and access methods. The enterprise is faced with tying mainframes, minis and PCs together for file sharing, electronic mail and routine data processing.

It is a daunting task connecting client machines running DOS, Windows, Mac and Motif with servers that run DOS, Windows, Mac, VMS, MVS and UNIX via protocols such as TCP/IP, IPX and NetBIOS over topologies such as Ethernet, Fast Ethernet, Switched Ethernet, Token Ring, FDDI and ATM using devices such as bridges, routers, hubs, switches and gateways, and there you have it. Oh, add interfacing between multiple mail and messaging systems and managing the entire process from one management console.

To understand the layers of protocols required to transmit a message from one machine to another, review the layers of the OSI model. OSI is a seven-layer reference model for worldwide communications defined by the International Standards Organization (ISO). Originally thought to be the future standard for communications, it was never widely supported. However, it does serve as a teaching model to line up other protocols against. See *enterprise networking*, *client/server* and *OSI model*.

## Programming Languages

Every software program is written in a programming language, and there is at least one programming language for every major CPU series. There is typically an assembly language and a number of high-level languages for each family. Assembly languages are machine specific, and the machine language they generate runs on only one CPU family. Unless the machine languages are very similar, it is difficult to translate an assembly language program from one CPU series into another.

The high-level programming language was created to eliminate this machine dependency. Programming languages, such as COBOL, FORTRAN and BASIC are designed to run on many different computers. However, due to dialects of each language, compatibility is always an issue. Each compiler vendor keeps adding new features to its language thereby making it incompatible with previous or other versions.

By the time a new feature becomes a standard, a dozen new features have been already implemented. For example, dBASE became a de facto standard business programming language. Since 1981, dBASE spawned competitive products, such as Clipper, QuickSilver, Force III, dbXL and Foxbase, all of which are incomplete versions of dBASE. None of them provides every command in dBASE, and they all provide features not found in dBASE.

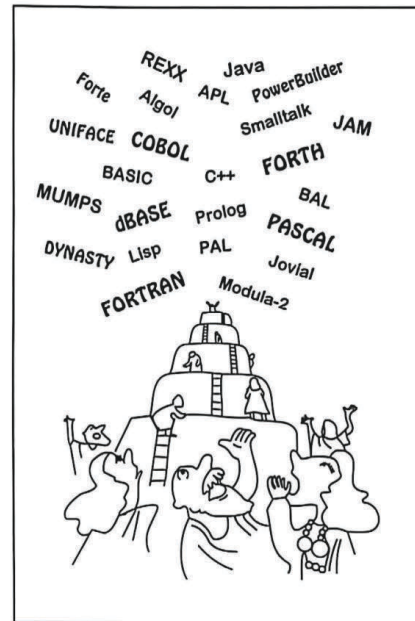
There's no rule of thumb for translating one dialect of a programming language into another. The job may be difficult or easy. At times, software is written to translate one dialect into another, as well as one programming language into another. If the translation program cannot translate the program entirely, then manual tailoring is necessary. In these cases, it is often easier to rewrite the program from scratch.

Compatibility can be achieved when a programming language conforms to the ANSI (American National Standards Institute) standard for that language. If the same version of an ANSI COBOL compiler is available for two different CPUs, a program written in ANSI COBOL will run on both machines.

Beyond traditional programming languages such as COBOL and C, known as third-generation languages, or 3GLs, there are more than a hundred

software environments used to develop client/server applications on LANs (see *client/server development system*). Each of them attempts to provide less programming-like and more English-like syntax, which is considered a *fourth-generation language*, or 4GL.

They also provide visual development tools to build the graphical user interface by "drawing" the screen and dragging and dropping symbols on it. They may provide the ability to point and click and drag and drop symbols to link objects together rather than by writing programming code. These software-building tools are proliferating and producing even more standards to be dealt with. The programming pools are becoming increasingly fragmented. It's no longer just COBOL or dBASE or C. Now it's PowerBuilder, SQLWindows, ObjectView, DYNASTY, OMNIS and JAM, etc. etc.



### File Management Systems

In its simplest form, a data file uses fields of the same length for each item of data, for example, a plain EBCDIC or ASCII file would look like:

```
Chris Smith  34 Main St.    Bangor    ME18567
Pat Jones    10 W. 45 St.   New York  NY10002
```

A common format created by BASIC programming languages is an ASCII comma delimited file; for example, the data above would look as follows:

```
"Chris Smith","34 Main St.,""Bangor","ME","18567"
"Pat Jones","10 W. 45 St.,""New York","NY","10002"
```

Both file formats above are simple, contain only data (except for quotes and commas) and can be easily manipulated by a word processor. However, data files may also contain special codes that identify the way the data is structured within the file. For example, variable length records require a code in each field indicating the size of the field.

Whether fixed or variable length fields, the data in non-DBMS systems is linked directly to the processing. The program must know the layout of the fields in each record that it processes, and it cannot accept records in a different format. In order to process a different file format, the program must be changed.

Incompatible file formats often exist within the same organization because they were developed separately. The following fixed-length records are incompatible even though they contain the same data. As long as a file management system is used rather than a database management system (DBMS), the program that processes the first file structure would have to be changed to process the second.

|      |         |      |    |     |
|------|---------|------|----|-----|
| Name | Address | City | St | Zip |
| Name | Address | City | St | Zip |

### Record Formats

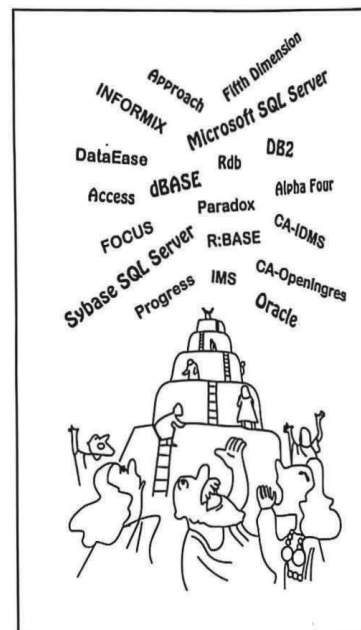
These two fixed-length record layouts are incompatible even though they contain the same kinds of data. The same program cannot process them unless it's designed to input both record formats. In order to process a different file, the program has to be changed or the file structure (record layout) has to be changed.

### DBMSs (Database Management Systems)

DBMSs have their own proprietary formats for storing data. For example, a header record with a unique format that contains identification data is typically placed at the beginning of each file. Codes may also be embedded in each record.

Most DBMSs have an import and export capability that converts popular database formats into their proprietary format. If not, the program usually can import and export a plain EBCDIC or ASCII file, which is stripped of all proprietary codes and can be used as a common denominator between both systems. If conversion facilities cannot be found, a custom program can be written to convert one database format into another if documentation describing the old format is available.

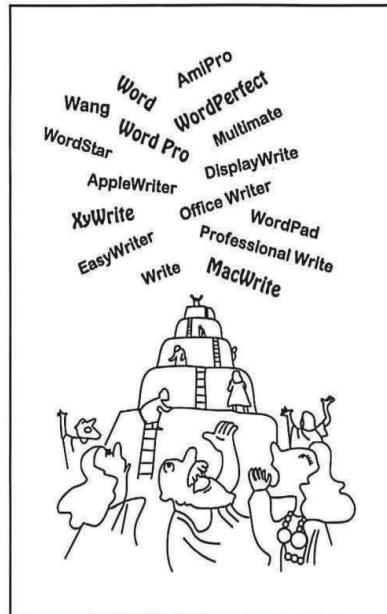
The application program interface (API), or language used by the application program to



"talk" to the database, is typically a proprietary language in every DBMS. However, SQL has become the de facto standard database language for client/server environments. In theory, that means any application program requesting data in the SQL language would work with any DBMS that supports SQL. Like everything else however, there are dialects of SQL, and only fundamental expressions can be guaranteed compatible. Whatever special features exist within the DBMS, proprietary syntax is needed to activate them, thus automatically making one DBMS incompatible with another.

## Text Systems

Although the basic structure of an English-language text file is standard throughout the world: word, sentence, paragraph, page; every word processing, desktop publishing and typesetting program uses its own codes to set up the layout within a document. For example, the code that turns on boldface in WordPerfect Version 5.0 is [BOLD]; in WordStar, it's ^PB.

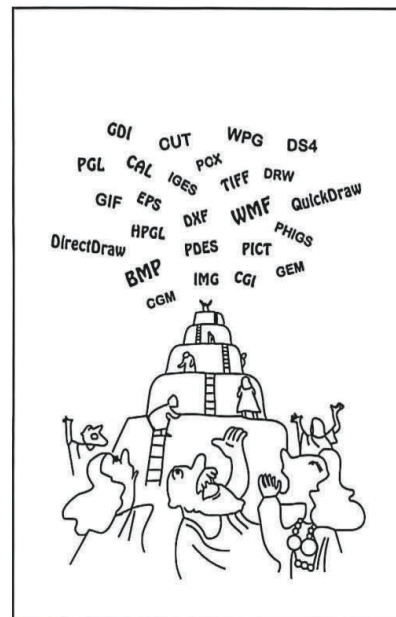


The codes that define a header, footer, footnote, page number, margin, tab setting, indent and font change are unique to the word processing program in which the document was created or the desktop publishing program into which the text file is converted. Even the codes to end a line and paragraph are not the same.

Document conversion is accomplished with special conversion programs or black boxes. Although every word processing program has a search and replace capability, it may not be effective for converting embedded layout codes from one format to another. In some programs, the search & replace simply does not handle layout codes. In addition, while some systems use one code to turn a function on and another code to turn it off, other systems use the same code for on and off, requiring manual verification and tailoring when using the search & replace function.

## Graphic Systems

There are many formats for storing a picture in a computer; but, unlike text and data files, which are primarily made up of alphanumeric characters, graphics formats are more complex. To begin with, there are the two major categories of graphics: vector graphics (objects made up of lines) and raster graphics (TV-like dots). Images stored in vector format can be moved to another vector system typically without loss of resolution. There are 2-D vector formats as well as 3-D vector formats. In transferring raster images among different devices, resolution is a major concern. Such transfers can occur without loss of resolution as long as the new format has the same or higher resolution as the older one. Standard graphics formats allow graphics data to be moved from



machine to machine, while standard graphics languages let graphics programs be moved from machine to machine. For example, GKS and PHIGS are major graphics languages that have been adopted by high-performance workstation and CAD vendors. Apple's consistent use of its QuickDraw language helped the Macintosh become popular in graphics-oriented applications. High-resolution graphics has typically been expensive to implement due to its large storage and fast processing requirements. However, as personal computers become more powerful, graphics are becoming widely used in business applications. The ability to see a person's face or a product's appearance on screen is now as commonplace as text and data.

## The Future

After 30 some years of computing, we've managed to create more than 3,000 languages, formats and programming interfaces in the computer business. While some of them become bona fide standards endorsed by recognized standards organizations, some of the most widely used are de facto standards. The PC is the perfect example.

The problem of standards is a never-ending dilemma. How do you forge ahead with new standards if they have to cling to the old designs for compatibility? At some point, the new has to be a break from the past, because the previous infrastructure is not usable. It seems to be the way of things.

## Star

The Xerox workstation that formally introduced the graphical user interface and desktop metaphor in 1981. Although unsuccessful, it was inspiration for Xerox's subsequent computers and for Apple's Lisa and Macintosh. See *Alto*.

**The First Commercial GUI**  
The Xerox Star was the first commercial implementation of a graphical user interface. This was the inspiration for the Macintosh and Windows. (Photo courtesy of Xerox Corporation.)

