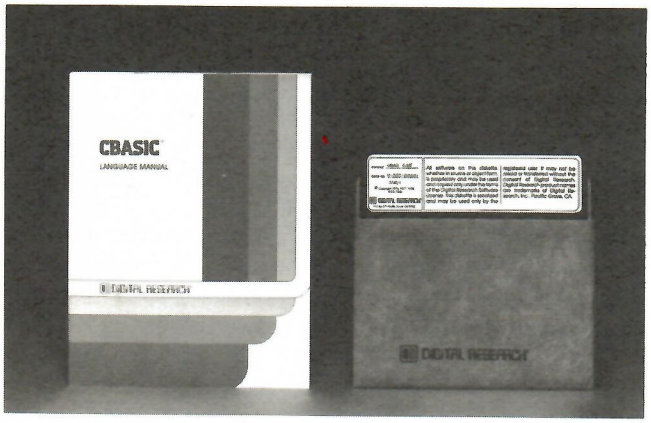
DIGITAL RESEARCH™ Product Brief

CBASIC®



CBASIC OVERVIEW

CBASIC is a commercial dialect of the popular BASIC language implemented as a compiler/interpreter. CBASIC is designed specifically to develop business applications. A source code file, created by a text editor or word processor, is compiled by CBASIC into an intermediate file composed of pseudo-code (P-code) instructions. During program execution, the CBASIC run-time monitor interprets each P-code instruction and performs the operation. This process enables CBASIC to use less random access memory (RAM) space, providing more memory area for the application program.

Due to its efficient use of RAM, precision arithmetic capability, high level file management and compatibility with 16-bit products, more software packages have been written in CBASIC for commercial business applications than in any other microcomputer language.

DECIMAL ARITHMETIC

CBASIC maintains real numbers in a binary coded decimal (BCD) floating point format, retaining 14 significant digits. Decimal arithmetic assures that fractional parts of dollar amounts will be exact and that ledgers will balance to the penny. Other binary floating point packages provide only seven significant digits, thus dollar amounts greater than $99,999.99 must be retained as a double precision number.

CBASIC supports integer arithmetic. Integer arithmetic is inherently faster on all microcomputers. By using integer variables whenever possible, program speed and throughput will be increased tremendously. Integer variables are defined in CBASIC by using a percent sign (%) as the last character of the variable name.

CBASIC FEATURES

- Extended Precision Decimal Arithmetic
- Expanded File Processing
- Comprehensive String Processing
- Efficient Use of Memory
- Reduced Software Development Time
- 16-Bit Compatibility
- Upward Compatibility to CB-80™
- Assembly Code Interface
- Debugging Capabilities
- Cross Reference Lister

EXPANDED FILE PROCESSING

CBASIC provides two types of files for maximum flexibility: fixed record length files and stream input/output files. Fixed record length files may have records of any length selected by the programmer. Files may be read sequentially or randomly, even allowing intermixing of the two methods of accessing. As records are read from the file, the values are automatically assigned to variables.

To minimize the impact on the available string space, buffers are dynamically allocated, and large records are automatically processed in 128-byte segments.

COMPREHENSIVE STRING PROCESSING

CBASIC offers a full complement of string functions. All strings are dynamically allocated and unused space is returned to the system automatically. String processing algorithms are specifically designed to minimize the use of memory and to reduce unnecessary movement of strings in memory. String variables are recognized by the compiler by appending a dollar sign ($) as the last character of the variable name.

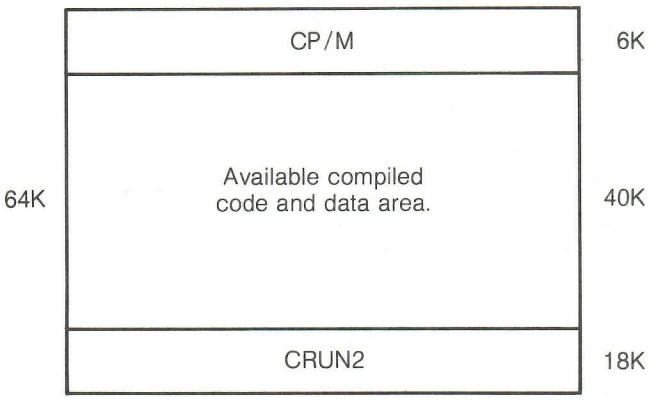
UPWARD COMPATIBILITY

CBASIC is upward compatible with CB-80, a native code compiler. The compatibility features of CBASIC allow programmers to develop and maintain only one copy of the source, this reduces maintenance, while supporting multiple hardware units.

CBASIC is also compatible with its 16-bit counterparts. Source code written under CBASIC is compatible with CBASIC-86™ for the Intel 8086/8088 microprocessor operating under CP/M-86™. CBASIC is compatible with CBASIC-16™, a version of CBASIC which operates under UNIX®.

EFFICIENT USE OF MEMORY

The run-time monitor CRUN2 is designed to provide the maximum amount of memory for user programs, while including all the features needed for commercial applications. The use of an intermediate representation of the program, often called P-code, gives the most compact coding for the program. This approach also provides significant speed improvements over interpreters.



All string and array data items are dynamically allocated to maximize the use of available memory. Memory is immediately reclaimed when a string or array is no longer required.

ASSEMBLY CODE INTERFACE

CBASIC provides a means to interface routines written in assembler with CBASIC. The PEEK and POKE functions can examine or change locations in memory. The SAVEMEM statement reserves space in memory for machine language subroutines. At execution time, a specified file that contains the assembler routine is loaded into memory. A CALL statement can be used to execute the assembler routine.

REDUCED SOFTWARE DEVELOPMENT TIME

CBASIC reduces software development time. The CBASIC compiler does not penalize programmers for using blanks and remarks in the source code, therefore CBASIC programs are easier to read and maintain. CBASIC supports variable names up to 31 characters in length to assist program documentation.

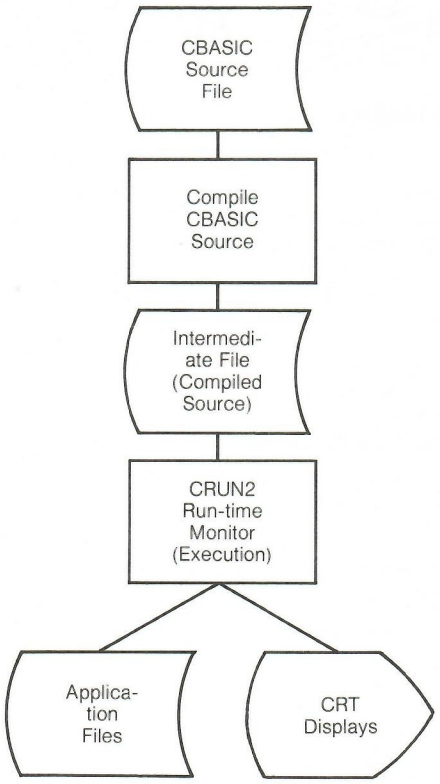
CBASIC programs can be created in modules. The multi-line function feature of CBASIC allows the programmer to define frequently used code as functions. These functions and modules can easily be combined to form a complete program.

The %INCLUDE directive tells the CBASIC compiler to include source code, located in a separate file, as part of the program. By using this directive, programmers can share frequently used source code. For example, file definitions, special routines or CRT definitions can be placed in a file which is accessible by many programmers, all using the same source code. For maximum flexibility, the %INCLUDE directives can be nested six deep.

EXECUTING CBASIC PROGRAMS

CBASIC executes application programs in two steps. First, the CBASIC compiler compiles the source code into a pseudo-code (P-code). This removes all comments, blanks, or blank lines from the executable portion of the code. The P-code generated by CBASIC makes the file small in comparison to the original source file.

The run-time monitor contains the operating system interface routines, standard function calls, arithmetic processing and string processing routines. Because the run-time monitor does not contain an editor, it is only 18K bytes long. Combining the small run-time monitor with the compact P-code interpreter increases the amount of application program area available for program data. As a result, the application programs are far more functional and performance oriented.



Software vendors marketing application programs are not required to provide source code to a customer. The intermediate file contains all the instructions translated to P-code. This guarantees that the application program delivered has not been modified and that no changes have been made to the source. This helps software vendors to protect their products.

TRACE OPTION

The trace option is available to help debug programs during execution. This allows the programmer to monitor the execution of a program by displaying each line number on the console as it is being executed. The range of line numbers to trace can be specified, allowing the programmer to isolate certain areas of a program and monitor the execution.

CROSS REFERENCE LISTER

The Cross Reference Lister is a utility program that creates a disk file containing an alphabetized list of all identifiers used in a CBASIC program. The way the identifier is used function, parameter, or global is provided, as well as a list of each line that the identifier appears in.

COMPILER DIRECTIVES

Compiler directives are commands which are interpreted by the compiler to perform certain functions. These directives are included as part of the source code and are executed at compile time.

%INCLUDE
The %INCLUDE directive is used to include source code from a separate file as part of a compiled program.

%CHAIN
The %CHAIN directive is used to set the size of a program's constant, code, data, and variable areas.

%EJECT
The %EJECT directive positions the listings directed to the printer to the top of the next page by outputting a formfeed character.

%LIST & %NOLIST
The %LIST and %NOLIST directives allow selected portions of a program to be listed during compilation.

%PAGE
The %PAGE directive sets the length of a page output to the printer.

COMMAND LINE DIRECTIVES

Command line directives are compiler options, which may be selected when compiling CBASIC source code. These directives are included as part of the command line when a file is compiled.

- **B** suppresses the listing of a program on the console during compilation.
- **C** suppresses the generation of an INT file, to increase compilation speed. This is useful to quickly check the programs syntax.
- **D** suppresses translation of lower case letters to upper case, in the source program.
- **E** directs the run-time program to accompany any error messages with the CBASIC line number that the error occurred in. This directive also enables the TRACE option during execution.
- **F** directs the compiled output lists to the printer, as well as to the console.
- **G** directs the compiled output listing to a disk file, as well as to the console.

CBASIC LANGUAGE STATEMENTS & FUNCTIONS

Control Statements
CHAIN
FOR-NEXT
GOSUB
GOTO
IF . . . THEN . . . ELSE
ON GOSUB
ON GOTO
RETURN
STOP
WHILE-WEND

String Functions
ASC(A$)
CHR$(I%)
LEFT$(A$,I%)
LEN (A$)
MATCH(A$,B$,I%)
MID(A$,I%,J%)
RIGHT$(A$,I%)
SADD(A$)
STR$(X)
UCASE$(A$)
VAL(A$)
VARPTR(variable)

Numeric Functions
ABS(X)
ATN(X)
COS(X)
EXP(X)
FLOAT(I%)
INT(X)
INT%(X)
LOG(X)
RND
SGN(X)
SIN(X)
SQR(X)
TAN(X)

Miscellaneous
COMMAND$
RANDOMIZE
FRE

File Statements and Functions
CLOSE
CREATE
DELETE
FILE
IF END
OPEN
PRINT
READ
RENAME (A$,B$)
SIZE (A$)

Input/Output Statements and Functions
CONCHAR%
CONSOLE
CONSTAT%
DATA
INP
INPUT
LPRINTER
OUT
POS
PRINT
PRINT USING
READ
RESTORE
TAB

Machine Language Interface Statements
CALL
PEEK
POKE
SAVEMEM

CBASIC PACKAGE

CBASIC is shipped on an 8-inch single-sided, single-density diskette. The diskette contains the following machine-readable files:

CBAS2.COM
CRUN2.COM
XREF.COM

SOFTWARE PERFORMANCE REPORT

CBASIC is supported by Digital Research's Software Performance Report (SPR) system. This service provides a prompt response to technical problems associated with CBASIC. Users are provided with SPR forms, which serve as a communications device to inform the Digital Research Product Support staff of user-identified problems.

LANGUAGE SOFTWARE DIRECTORY

The Language Software Directory serves as support for Vertical Market Specialists. Systems integrators use the directory as an up-to-date reference tool for accounting and vertical market application packages.

The directory contains a listing of companies actively marketing standard business application packages, followed by product name and description, and memory requirements.

HARDWARE REQUIREMENTS

- Intel 8080/8085 or ZILOG Z-80 microprocessor
- Operates with any CP/M®, MP/M II™ or CP/NET™ system
- Minimum of 24K bytes of memory; 48K bytes is recommended

DIGITAL RESEARCH

Digital Research, Pacific Grove, CA is the leading producer of microcomputer operating systems, languages and utilities, for 8- and 16-bit microcomputers. For 8 years, Digital Research has been involved with the design, development, and support of microcomputer software. Digital Research's operating systems are the industry standard. Digital Research's languages and programming tools are designed for the professional programmer writing commercial software packages. Together, they form a family of compatible software products. Digital Research users include over 300,000 systems, 400 OEMs and 500 independent software houses.

ORDER INFORMATION

Product	Order Description
CBASIC Language	One 8" single-density, single-sided diskette and CBASIC Documentation.
CBASIC Documentation	CBASIC: A Reference Manual

DIGITAL RESEARCH™
P.O. Box 579
Pacific Grove, CA 93950
408-649-3896
TWX 910 360 5001