



Tech Info Library

GS BASIC: Features and Editing Commands

Revised: 10/19/87
Security: Everyone

GS BASIC: Features and Editing Commands

=====

This article last reviewed: 15 October 1987

ToolBox Primitives

GS BASIC includes primitives (built-in, low-level commands) that let the user interact with the ToolBox routines. Whereas most other ToolBox routines are implemented in external libraries that need to be explicitly called up, these commands -- Eventdef, Menudef, and Taskpoll -- are included in GS BASIC itself because they bind the ToolBox and BASIC line numbers.

Eventdef. The structure of Macintosh-like programs, as implemented by the Apple II GS and its ToolBox routines, requires that programs be event-driven. This means that the heart of a program is a main loop that constantly monitors what events -- mouse clicks, keyboard presses, window update events, network activity, and so on -- have taken place. Eventdef associates BASIC line numbers with these events: whenever one of these events occurs, GS BASIC starts running the instructions specified in a table set up by the programmer and maintained by Eventdef.

Menudef. One of the places a mouse click can occur is in the menu bar. When this happens, a menu is displayed, letting the user select a menu item. The user specifies that a certain function is to be performed. Like Eventdef, Menudef associates BASIC line numbers with menu items, so that when a menu item is selected, GS BASIC can easily determine what statements to run, and run them.

Taskpoll. Programs shouldn't start fielding events and responding to them until all the ToolBox libraries are loaded and the tables that Menudef and Eventdef maintain are initialized. The Taskpoll command lets the programmer turn off the interrupting feature of GS BASIC and turn them on when ready for them.

Procedures and Functions

The most convenient structure of of any program is a procedure, function, or

subroutine. In most BASICs, including Applesoft, subroutines are implemented in a very straightforward fashion. GS BASIC provides a more powerful way to specify and call procedures.

Applesoft lets programmers call subroutines with the GOSUB xxx command, where xxx is the line number to which control branches. Statements are then executed until a RETURN command is executed, at which time the program returns to the line that called the subroutines, and continues from there.

GS BASIC allows programmers to define specific procedures that are called with the PROC command. The PROC command performs the same functions as Applesoft's GOSUB, and lets the programmer pass local parameters as well.

Editing Commands

GS BASIC has a more powerful line editor than Applesoft's. Access it by typing EDIT linnum[-linnum]

CONTROL-D	delete character to the left of cursor; moves line to left
CONTROL-F	delete character under the cursor; moves line left
CONTROL-X	delete entire line
CONTROL-Y	delete line to left and under cursor
CONTROL-E	toggle insert mode
RETURN	accept entire line (even characters to right of cursor)
DELETE	same as CONTROL-D

GS BASIC Commands

_ (underscore)	ERRLIN	LOG(SCALB(
ABS(ERROR	LOG1(SCALE(
AND	ERRTOOL	LOG2(SECONDS@
ANU(ERRTXT\$(LOGB%(SET
APPEND	EVENTDEF	MENUDEF	SGN(
AS	EXCEPTION	MID\$(SHOWDIGITS
ASC(EXEC	MOD	SIN(
ASSIGN	EXEVENT@ (NEGATE(SPACE\$(
ATN(EXFN	NEW	SPC(
AUTO	EXP(NEXT	SQR(
AUXID@	EXP1(NORMAL	SRC
BASIC@ (EXP2(NOT	STEP
BDF	FILE(NOTRACE	STOP
BREAK	FILTYP(OFF	STR\$(
BTN(FILTYP=	ON	SUB\$(
CALL	FIX(OPEN	SWAP
CALL%	FN	OR	TAB(
CAT	FOR	OUTPUTPUT	TAN(
CATALOG	FRE	OUTREC	TASKPOLL
CHAIN	FREMEM(PDL(TASKREC%(
CHR\$(GET	PEEK(TASKREC@ (
CLEAR	GOSUB	PERFORM	TEN(
CLOSE	GOTO	PFX\$(TEXT
COMPI(GRAF	PI	TEXTPORT
CONT	HEX\$(POKE	THEN

CONV(HLIST	PREFIX	TIME\$
CONV@	HOME	PREFIX\$	TIME(
CONV#	HPOS	PRINT	TIMER
CONV\$(IF	PROC	TO
CONV%(IMAGE	PROGNAM\$	TRACE
CONV& (INDENT	PUT	TXT
COPY	INIT	QUIT	TYPE
COS(INPUT	R.STACK%(TYP(
CREATE	INSTR(R.STACK@ (UBOUND(
DATA	INT(R.STACK& (UCASE\$(
DATE\$	INVERSE	RANDOMIZE	UIR(
DATE(INVOKE	READ	UNLOCK
DEF	JOYX(REC(UNTIL
DEL	JOYY(RELATION(UPDATE
DELETE	KBD	REM	USING
DIM	LEFT\$(REMDR	VAL(
DIR	LEN(RENAME	VAR(
DIV	LET	REP\$(VAR\$(
DO	LIBFIND	RESTORE	VARPTR(
EDIT	LIBRARY	RESUME	VARPTR\$(
ELSE	LIST	RETURN	VOLUMES
END	LISTTAB	RIGHT\$(VPOS
EOF	LOAD	RND(WHILE
EOFMARK(LOCAL	ROUND(WRITE
ERASE	LOCATE	RUN	XOR
ERR	LOCK	SAVE	

Toolbox Definition Files (TDFs)

QuickDraw	ADB	Desk	Dialog
Event	Font	Intmath	LineEdit
List	Loader	Locater	Memory
Menu	MiscTool	NoteSyn	Print
QDAux	Scheduler	Scrap	StdFile
Text	Window	Control	

Built-in Constants and Reserved Variables

PI	to 20 digits
LISTTAB	number of spaces between line number and first statement
INDENT	controls indenting of FOR...NEXT and WHILE...UNTIL loops
OUTREC	maximum line length
VPOS	vertical cursor position
HPOS	horizontal cursor position
FRE	amount of memory available
SHOWDIGITS	specifies the number of digits to display for REAL numbers
KBD	holds value of last key pressed
ERR	contains error number
ERRLIN	contains line where error occurred

Variable Specifications

Real: (+ or - 1.7E38) (less than 1.5E-45 equals 0)

Double real: (+ or - 1.7E308) (less than 5.0E-324 equals 0)
Integer: (%) (2 bytes) (-32768 to 32767)
Double integer: (@) (4 bytes) (-2147483648 to 2147483647)
Long integer: (&) (8 bytes) (-9223372036854775808 to 9223372036854775807)
String: (\$) 255 characters
Arrays: (!)

Tech Info Library Article Number:795