

Appendix A: Commands and Functions

This Appendix presents all the filePro commands and functions. The first pages list them in functional groups, by name only. The remaining pages list them alphabetically, presenting their name, syntax, examples, and other useful information.

NOTE

THIS APPENDIX IS PRECEDED BY TWO INDEXES. THIS FIRST INDEX (2 PAGES) IS IN ORDER OF THE COMMANDS IN THE APPENDIX. THE SECOND INDEX (1 PAGE) LISTS THE COMMANDS AND FUNCTIONS IN ALPHABETIC ORDER, WITH PARAGRAPH REFERENCES.

01	<u>Programming Control Commands</u>				563
	CALL	CALL NOAUTO	CHAIN	END	
	EXIT	GOSUB	GOSUB OF	GOTO	
	GOTO OF	RESTART	RETURN	SCREEN	
	SKIP	SLEEP	SWITCHTO	UPDATE	
02	<u>Output Control Commands</u>				573
	FORM	FORMM	HARDCOPY	LOGTEXT	
	OUTS	PAGE	PRINT	PRINTER	
	READOUTPUT()	RESET	SELECT	SORT	
	SORTINFO()				
03	<u>Directory and File Management Commands</u>				579
	ACCESS()	CHDIR	EXISTS()	FILENAME()	
	GETCWD()	OPENDIR()	NEXTDIR()	CLOSEDIR()	
	FREESPACE()	NUMFIELD()	NUMRECS()	SYSTEM	
	USER				
04	<u>HTML and GUI Commands</u>				587
	CURSOR	CURSOR PATH	GUI	HTML	
	HTMLERRNO()	JSFILE	MESSAGE	MOUSE PATH	
	NOT HTML	TOHTML()			
05	<u>File I/O Commands</u>				589
	CLOSE	CLOSE()	CREATE()	EXPORT	
	FILESIZE()	IMPORT	OPEN()	READ()	
	READLINE()	RECLLEN()	REMOVE()	SEEK()	
	TELL()	WRITE	WRITE()	WRITELINE()	
06	<u>Lookup Commands</u>				595
	CLEARB	COPY	COPY TO	COPYIN	
	DELETE	DROP	DROPALL	DROPALL BEFORE	
	DROPALL AFTER	FIELDEDIT()	FIELDLEN()	FIELDNAME()	
	FIELDVAL()	GETNEXT	GETPREV	LOCKED()	
	LOOKUP	SYNC			

07	<u>Other Processing Commands</u>				599
	BACKGROUND	BEEP	BREAK	CLEAR	
	CLS	DEBUG	DECLARE	DIM	
	ERRNAME()	ESCAPE	GETENV()	HELP	
	IXCOMMENT()	IXSORT()	MENU	PUTENV()	
	READSCREEN()	SAVE	SET	STATUS()	
	VIDEO				
08	<u>Message and Display Commands</u>				604
	CLEARP	CLEAR	DISPLAY	ERRORBOX	
	INPUT	INPUT POPUP	INPUTPW	INPUTPW POPUP	
	LISTBOX()	MSGBOX	POPUP	POPUP UPDATE	
	SHOW	SHOW POPUP	SHOW RAW	SHOWCTR	
	SHOWTOCOL				
09	<u>Keyboard Commands</u>				610
	FLUSHKEY	INKEY	PUSHKEY	WAITKEY	
10	<u>Field Manipulation Commands - Dates</u>				611
	ADDMONTH()	BOM()	BOQ()	BOY()	
	DOM()	DOQ()	DOY()	DOW()	
	EOM()	EOQ()	EOY()	ISLEAP()	
	MDAY()	WOM()	WOQ()	WOY()	
11	<u>Field Manipulation Commands - Math</u>				614
	ABS()	AVG()	BASE()	CEIL()	
	FLOOR()	FRAC()	INT()	MAX()	
	MIN()	MOD()	PI()	RAND()	
	SIGN()	SQRT()	TOT()		
12	<u>Field Manipulation Commands - Financial</u>				618
	TVM_FV()	TVM_I()	TVM_N()	TVM_PMT()	
	TVM_PV()				
13	<u>Field Manipulation Commands - Trigonometry</u>				619
	SIN()	DSIN()	SINH()		
	ASIN()	DASIN()	ASINH()		
	COS()	DCOS()	COSH()		
	ACOS()	DACOS()	ACOSH()		
	TAN()	DTAN()	TANH()		
	ATAN()	DATAN()	ATANH()		
	DTOR()	RTOD()			
14	<u>Field Manipulation Commands - Logarithms</u>				621
	EXP()	EXP10()	LOG()	LOG10()	
15	<u>Field Manipulation Commands - Other Field Data</u>				622
	ASC()	BLOB	CHR()	COMPARE()	
	DLEN()	DOEDIT()	EDIT()	INSTR()	
	LEN()	MEMO	MID()	REPEAT()	
	SOUNDEX()	STRTOK()	XLATE()		

ALPHABETIC LIST OF filePro's 203 COMMANDS AND FUNCTIONS

COMMAND	VERSION	PARA	COMMAND	VERSION	PARA
ABS()		145	DTAN()	(v4.8)	178
ACCESS()	(v4.8)	30	DTOR()	(v4.8)	183
ACOS()	(v4.8)	174	EDIT()		195
ACOSH()	(v5.0)	176	END		4
ADDMONTH()	(v4.5)	129	EOM()	(v4.5)	137
ASC()		189	EOQ()	(v4.5)	138
ASIN()	(v4.8)	168	EOY()	(v4.5)	139
ASINH()	(v5.0)	170	ERRNAME()	(v4.5)	95
ATAN()	(v4.8)	180	ERRORBOX		111
ATANH()	(v5.0)	182	ESCAPE		96
AVG()		146	EXISTS()		32
BACKGROUND	(v5.0)	87	EXIT		5
BASE()	(v4.5)	147	EXP()	(v4.8)	185
BEEP		88	EXP10()	(v4.8)	186
BLOB	(v5.0)	190	EXPORT		56
BOM()	(v4.5)	130	FIELDEDIT()	(v4.5)	78
BOQ()	(v4.5)	131	FIELDLEN()	(v4.5)	79
BOY()	(v4.5)	132	FIELDNAME()	(v4.5)	80
BREAK		89	FIELDVAL()	(v4.5)	81
CALL		1	FILENAME()	(v5.0)	33
CALL NOAUTO	(v5.0)	2	FILESIZE()	(v4.5)	57
CEIL()	(v4.5)	148	FLOOR()	(v4.5)	149
CHAIN		3	FLUSHKEY	(v4.8)	125
CHDIR	(v4.5)	31	FORM		17
CHR()		191	FORMM		18
CLEAR		90	FRAC()		150
CLEARB		69	FREESPACE()	(v4.8)	38
CLEARP		108	GETCWD()	(v4.5)	34
CLEARs	(v4.5)	109	GETENV()		97
CLOSE	(v4.5)	53	GETNEXT		82
CLOSE()		54	GETPREV		83
CLOSEDIR()	(v4.8)	37	GOSUB		6
CLS		91	GOSUB OF	(v4.5)	7
COMPARE()	(v4.5)	192	GOTO		8
COPY		70	GOTO OF	(v4.5)	9
COPY TO	(v4.8)	71	GUI	(v5.0)	45
COPYIN		72	HARDCOPY		19
COS()	(v4.8)	171	HELP		98
COSH()	(v5.0)	173	HTML	(v4.8)	46
CREATE()	(v4.5)	55	HTMLERRNO()	(v4.8)	47
CURSOR	(v5.0)	43	IMPORT		58
CURSOR PATH	(v5.0)	44	INKEY		126
DACOS()	(v4.8)	75	INPUT		112
DASIN()	(v4.8)	169	INPUT POPUP		113
DATAN()	(v4.8)	181	INPUTPW		114
DCOS()	(v4.8)	172	INPUTPW POPUP		115
DEBUG		92	INSTR()		196
DECLARE	(v4.8)	93	INT()	151	
DELETE		73	ISLEAP()	(v4.5)	140
DIM		94	IXCOMMENT()	(v4.8)	99
DISPLAY		110	IXSORT()	(v4.8)	100
DLEN()		193	JSFILE	(v4.8)	48
DOEDIT()	(v4.5)	194	LEN()		197
DOM()	(v4.5)	133	LISTBOX()	(v4.5)	116
DOQ()	(v4.5)	134	LOCKED()	(v4.8)	84
DOW()	(v4.5)	136	LOG()	(v4.8)	187
DOY()	(v4.5)	135	LOG10()	(v4.8)	188
DROP		74	LOGTEXT (v4.5)	20	
DROPALL	(v4.5)	75	LOOKUP		85
DROPALL BEFORE		76	MAX()		152
DROPALL AFTER		77	MDAY()	(v4.5)	141
DSIN()	(v4.8)	166			

COMMAND	VERSION	PARA	COMMAND	VERSION	PARA
MEMO	(v5.0)	198	TAN()	(v4.8)	177
MENU		101	TANH()	(v5.0)	179
MESSAGE		(v5.0) 49	TELL()	(v4.5)	65
MID()		199	TOHTML()	(v4.8)	52
MIN()		153	TOT()		159
MOD()		154	TVM_FV()	(v5.0)	160
MOUSE PATH	(v5.0)	50	TVM_I()	(v5.0)	161
MSGBOX		117	TVM_N()	(v5.0)	162
NEXTDIR()	(v4.8)	36	TVM_PMT()	(v5.0)	163
NOT HTML	(v4.8)	51	TVM_PV()	(v5.0)	164
NUMFIELD()	(v4.5)	39	UPDATE		12
NUMRECS()	(v4.5)	40	USER		42
OPEN()	(v4.5)	59	VIDEO	(v4.5)	107
OPENDIR()	(v4.8)	35	WAITKEY		128
OUTS		21	WOM()	(v4.5)	142
PAGE		22	WOQ()	(v4.5)	143
PI()	(v4.8)	155	WOY()	(v4.5)	144
POPUP		118	WRITE	(v4.5)	66
POPUP UPDATE		119	WRITE()	(v4.5)	67
PRINT		23	WRITELINE()	(v4.5)	68
PRINTER		24	XLATE()	(v4.5)	203
PUSHKEY		127			
PUTENV	(v4.5)	102			
RAND()		156			
READ()	(v4.5)	60			
READLINE()	(v4.5)	61			
READOUTPUT()	(v5.0)	25			
READSCREEN()	(v4.8)	103			
RECLN()	(v4.8)	62			
REMOVE()	(v4.5)	63			
REPEAT()	(v4.5)	200			
RESET		26			
RESTART		10			
RETURN		11			
RTOD()	(v4.8)	184			
SAVE	(v5.0)	104			
SCREEN		12			
SEEK()	(v4.5)	64			
SELECT		27			
SET		105			
SHOW		120			
SHOW POPUP	(v4.5)	121			
SHOW RAW	(v5.0)	122			
SHOWCTR	(v4.5)	123			
SHOWTOCOL	(v4.5)	124			
SIGN()	(v4.5)	157			
SIN()	(v4.8)	165			
SINH()	(v5.0)	167			
SKIP		13			
SLEEP	(v4.5)	14			
SORT		28			
SORTINFO()	(v4.8)	29			
SOUNDEX()		201			
SQRT()		158			
STATUS()	(v5.0)	106			
STRTOK()	(v4.8)	202			
SWITCHTO	(v4.5)	15			
SYNC	(v4.5)	86			
SYSTEM		41			

Appendix A: Commands and Functions

Programming Control Commands

A.01 CALL

Calls another table as a subroutine.

Syntax - CALL "filename"
CALL "full path name to prc table"

- 01 If the CALLED table is in the same filePro file as the parent table only the filename is required.
- 02 If the CALLED table is in another filePro file it MUST be named with FULL path name.
call /appl/filepro/Library/popcal
- 03 When the CALLED table is ended the program returns to the original table.
- The CALL table ends with an END command or a SCREEN command.
 - The program pointer then goes to the next line of code in the original table, after the CALL command.
- 04 The CALL table has its own set of dummy fields.
- It only recognizes other dummies that have been declared global in the automatic table. When the CALL table ends, all of its other dummy fields are cleared and restored to the original table's values.

```

18 ----- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
@wuk36 If:          '***** trigger pop up of the calendar
      Then: call "popcal"
19 ----- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      If: za eq 36          'Exited from popcal w/no change
      Then: screen ,36
20 ----- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      If:                  'Got new value from dummy in popcal
      Then: 36=za; display; pushkey "[ENTR]"; end
21 ----- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

```
 - To pass a value from the CALLED table to the original table:
 - You must make the dummy **global**,
 - You must declare it in the automatic table. To be safe, also declare it in the calling table and in the called table (just to be sure it sticks!).
- 05 Long name (DECLARE) variables can be used in CALLED tables.
- If they are only used in the CALLED table, they need only be DECLARED (or DECLAR LOCAL).
 - If they are going to pass data from the CALL table to the original table, they must be DECLARED GLOBAL either in the automatic table or the original table. Then they must be DELCARED EXTERN in the CALL table.
- 06 The CALL table is loaded into memory when the CALL command is executed. The table remains in memory until another CALL to another table is executed, or until processing ends.
- 07 You cannot CALL another table from within a CALL table. You cannot CHAIN CALLED tables together.
- HINT** - CALL allows you to build a library of program modules. These can be stored and used in multiple applications simply by CALLing them with full path name.
- 08 Arrays and labels used in a CALLED table expire when processing returns to the main table. They are local to the CALL table.

Appendix A: Commands and Functions

Programming Control Commands

A.01 CALL (Continued)

09 @wef, @wlf and other When-Field labels can be used in a CALL table.

1. They work just like in an input table. They can end with an END or a SCREEN command (which will only end the processing under the label).
2. You CANNOT use the SCREEN command to load a screen, however, because the CALL table will see it as the end of processing. You CAN, however, use POPUP UPDATE -.

EXAMPLE:

```

  If: ' COPYRIGHT |   ©) 1997 STN, Inc.   All Rights Reserved
 2 Then: 'WRITTEN BY |   B. Easton   03/23/97
    If: '  filePro |   4.1
 3 Then: '  Usage |   clerk           'CALLED INPUT TABLE
    If:
 7 Then: "'custpr" in called from "ncewohdrinpnew", the main input table.
    If: '***** put up customer screen ****
11 Then: popup update -, "cust"      "'cust" is a regular filePro screen!
    If:
12 Then: clearp
    If: @sk="BRKY"                '** user keyed break-key
13 Then: end
    If: 95 ne ""
14 Then: gosub savit
    If:
15 Then: end                      ← NOTE - This ENDS the CALL table
    If: '*****
16 Then: '***** SUBROUTINES *****
savit If: 'save data from popup to file
55 Then:
    If:
56 Then: lookup fon = wmmphon@ k=95  i=A -nx
newph If: not fon
57 Then: lookup fon = wmmphon@ r=free -ep
    If: newph
58 Then: fon(1)=95
    If: '** phone fields and nearest intersection
59 Then: fon(2)=uh; fon(3)=uf; fon(4)=ur
    If:
60 Then: fon(5)=ud; fon(6)=uo; fon(7)=ui; write fon
    If:
61 Then: return '*****
@wef15 If:
62 Then: show "** Enter Name of Caller \kB To Lookup Account **"; end
@wef95 If:
63 Then: uk=95; show "** Enter Account Number \kB To Lookup Account **"; end
@wefuh If: 95 eq ""
66 Then: skip
    If:
67 Then: show "** Enter Phone Number Or \kJ To Use Caller's Home Phone No. Above **"; end
@whpuh If:
68 Then: uh=58; display; pushkey "[ENTR]"; end
@wbl15 If:
77 Then: qm(12,*)=95; gosub brsit
    If: cst
78 Then: 95=cst(1); gosub fillus; display
    If: cst and qm ne 95
79 Then: goto ioio
    If:
80 Then: screen ,15                ← NOTE - This only ENDS the @WBL15 label

```

Appendix A: Commands and Functions

Programming Control Commands

A.02 CALL NOAUTO

Calls another table as a subroutine.

Syntax - CALL NOAUTO "filename"
CALL NOAUTO "full path name to prc table"

This command is identical to CALL, **except** all dummy fields defined in the automatic table are ignored.

A.03 CHAIN

Chain links a table to the current table.

Syntax CHAIN "filename"
CHAIN "full path name to prc table"

- 01 Unlike CALL, CHAIN tables retain all the values of the original table. This includes dummy fields and DECLARE GLOBAL longname fields. CHAIN tables are linked, not independent.
- 02 You can CHAIN multiple CHAIN tables.
- 03 Unless the CHAINED table is in the current filePro directory, you must use the full path name.
- 04 CHAINED tables are significantly slower in operation than CALL tables.

A.04 END

Stops processing immediately.

- 01 End is not needed at bottom of table.

- 02 End is used to prevent waste of processing time.

```

1  -----
   If: 1 eq "F"          'Got my answer - why go on?
   Then: action1; end
2  -----
   If: 1 eq "G"
   Then: action2; end
3  -----
   If: 1 eq "H"
   Then: action3
4  -----
   If:
   Then: end            'not necessary but - just in case!
5  -----

```

- 03 End separates one section of the processing table from another.

```

11 -----
   If:                  'no input processing goes past here
   Then: end
12 -----
@wlf1 If:                'start of when-field processing
      Then: . . .
13 -----

```