

Whitesmiths, Ltd.

PASCAL PROGRAMMERS' MANUAL

Release: 2.2

Date: March 1983

The C language was developed at Bell Laboratories by Dennis Ritchie; Whitesmiths, Ltd. has endeavored to remain as faithful as possible to his language specification. The external specifications of the Idris operating system, and of most of its utilities, are based heavily on those of UNIX, which was also developed at Bell Laboratories by Dennis Ritchie and Ken Thompson. Whitesmiths, Ltd. gratefully acknowledges the parentage of many of the concepts we have commercialized, and we thank Western Electric Co. for waiving patent licensing fees for use of the UNIX protection mechanism.

The successful implementation of Whitesmiths' compilers, operating systems, and utilities, however, is entirely the work of our programming staff and allied consultants.

For the record, UNIX is a trademark of Bell Laboratories; IAS, RSTS/E, VAX, VMS, P/OS, PDP-11, RT-11, RSX-11M, and nearly every other term with an 11 in it all are trademarks of Digital Equipment Corporation; CP/M is a trademark of Digital Research Co.; MC68000 and VERSAdos are trademarks of Motorola Inc.; ISIS and iRMX are trademarks of Intel Corporation; A-Natural and Idris are trademarks of Whitesmiths, Ltd. C is not.

Copyright (c) 1978, 1979, 1980, 1981, 1982, 1983

by Whitesmiths, Ltd.

All rights reserved.

PASCAL PROGRAMMERS' MANUAL

SECTIONS

- I. The Pascal Language
- II. Standard Pascal Subprograms
- III. Pascal/Portable C Interface Library
- IV. Pascal Internal Subroutines

SCOPE

This manual describes the Pascal programming language, as implemented by Whitesmiths, Ltd., and the three significant levels of the Pascal environment. Section I introduces the Pascal language, and Section II details the standard subprograms that the language defines. Section III describes a series of non-standard routines that serve to interface Pascal programs to the Whitesmiths portable C environment. Section IV, which is not of interest to the majority of programmers, describes the major internal routines on which this implementation depends at run time; given this information, the ambitious user should be able to displace any of them at need.

Further information on the machine independent C environment, which is fully available to the Pascal user, may be found in the C Programmers' Manual.

TABLE OF CONTENTS

I. The Pascal Language

I - 1	Introduction	the Pascal compiler
I - 2	Syntax	syntax rules for Pascal
I - 5	Identifiers	naming things in Pascal
I - 9	Declarations	declaring names in Pascal
I - 15	Statements	the executable code
I - 19	Expressions	computing values in Pascal
I - 23	Style	rules for writing good Pascal code
I - 26	Differences	comparative anatomy
I - 29	Diagnostics	compiler complaints

II. Standard Pascal Subprograms

II - 1	Builtins	the standard Pascal subprograms
II - 2	abs	compute absolute value
II - 3	arctan	arctangent
II - 4	chr	convert integer to char
II - 5	cos	cosine in radians
II - 6	dispose	deallocate a variable created with new
II - 7	eof	test for end of input file
II - 8	eoln	test for end of input line
II - 9	exp	exponential
II - 10	get	fill a file buffer from the file
II - 11	ln	natural logarithm
II - 12	new	allocate an instance of a variable
II - 13	odd	test low-order bit of an integer
II - 14	ord	find position in type of an ordinal expression
II - 15	page	output a page break to a textfile
II - 16	pred	return preceding value in ordinal type
II - 17	put	flush a file buffer to the file
II - 18	read	read a list of variables from a file
II - 19	readln	read a line of input from a textfile
II - 20	reset	initialize a file for input
II - 21	rewrite	initialize a file for output
II - 22	round	round real to integer
II - 23	sin	sine in radians
II - 24	sqr	square an argument
II - 25	sqrt	real square root
II - 26	succ	return succeeding value in ordinal type
II - 27	trunc	truncate real to integer

II - 28	write	write a list of variables to a file
II - 30	writeln	write a line of output to a textfile

III. Pascal/Portable C Interface Library

III - 1	Conventions	using the Pascal interface library
III - 3	argc	command line argument count
III - 4	argv	command line argument vector
III - 5	getarg	get command line argument
III - 6	nargs	get number of command line arguments
III - 7	pclose	close a Pascal file variable
III - 8	pcreate	create Pascal file by name
III - 9	pinit	connect Pascal file variable to C file
III - 10	popen	open Pascal file by name
III - 11	pseek	administer Pascal read/write file
III - 12	pstat	get status of Pascal file

IV. Pascal Internal Subroutines

IV - 1	Conventions	the Pascal runtime
IV - 3	Mapping	Pascal to C
IV - 6	_pargs	array of program arguments
IV - 7	chr	integer to char
IV - 8	eof	test for end of file
IV - 9	eoln	test for end of line
IV - 10	iabs	integer absolute value
IV - 11	input	standard input file
IV - 12	isqr	integer square
IV - 13	main	enter Pascal program
IV - 14	odd	test for odd integer
IV - 15	output	standard output file
IV - 16	p_abort	print message and abort
IV - 17	p_bget	get binary records
IV - 18	p_bput	put binary records
IV - 19	p_ckfd	check Pascal file variable
IV - 20	p_close	close Pascal files
IV - 21	p_cmp	compare two buffers
IV - 22	p_copy	copy a buffer
IV - 23	p_disp	free allocated datum
IV - 24	p_fget	get data from text file
IV - 25	p_files	the open Pascal files
IV - 26	p_fill	validate input buffer
IV - 27	p_fput	put data to text file
IV - 29	p_get	perform Pascal get
IV - 30	p_load	internal validate input buffer
IV - 31	p_new	allocate new datum

IV - 32	p_pnam	determine Pascal permanent filename
IV - 33	p_put	perform Pascal put
IV - 34	p_read	internal read input buffer
IV - 35	p_rset	open a Pascal file for reading
IV - 36	p_rwri	create a Pascal file for writing
IV - 37	p_sand	perform set and operation
IV - 38	p_scon	perform set construction operation
IV - 39	p_scpy	perform set copy operation
IV - 40	p_sdif	perform set difference operation
IV - 41	p_sequ	perform set equality comparison
IV - 42	p_sin	perform set membership test
IV - 43	p_sleq	perform set inclusion test
IV - 44	p_sor	perform set or operation
IV - 45	p_sub	check subscript bounds
IV - 46	p_tget	get nonfloating data from text file
IV - 47	p_tput	put nonfloating data to text file
IV - 48	p_unam	generate Pascal temporary file name
IV - 49	p_write	internal write output buffer
IV - 50	page	put page delimiter