

Whitesmiths, Ltd.  
C Interface Manual  
for 8080

Edition 2.2  
March 1983





Whitesmiths, Ltd.

## C INTERFACE MANUAL FOR 8080

**Edition: 2.2**

**Date: March 1983**

**Revision: A**

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.**

## **C INTERFACE MANUAL FOR 8080**

### **SECTIONS**

- I.**           The A-Natural Language
- II.**          Programming Utilities
- III.a.**       Idris System Interface Library
- III.b.**       CP/M System Interface Library
- III.c.**       ISIS-II System Interface Library
- IV.**          Machine Support Library for 8080

### **SCOPE**

This manual describes the 8080/Z80 dependent aspects of the C programming environment provided by Whitesmiths, Ltd. In addition, it documents all of the utilities necessary for building new programs. Section I introduces the conventions and describes the format used by the Idris assembler. Section II succinctly describes the programming utilities of Idris, which also serve as cross support utilities for other host machines. Each subsection of Section III describes the library functions that interface the portable C library to Idris, CP/M (CDOS), or ISIS-II, and Section IV describes the runtime routines called upon by code produced by the 8080/Z80 C compiler.

Information on the C language and the portable library may be found in the C Programmers' Manual, while information peculiar to other machines supported by Whitesmiths, Ltd. is given in other C Interface Manuals.

THIS MANUAL IS PROVIDED WITH SEVERAL SOFTWARE PACKAGES, SOME OF WHICH USE ONLY A SUBSET OF THE FACILITIES DOCUMENTED. THE PRESENCE OF A MANUAL PAGE HERE DOES NOT IMPLY THAT THE CORRESPONDING SOFTWARE IS ALSO SUPPLIED.

## TABLE OF CONTENTS

### I. The A-Natural Language

I - 1	<b>As.80</b>	Introduction to A-Natural
I - 4	<b>Syntax</b>	rules for writing A-Natural
I - 7	<b>Defining</b>	defining symbols, data, and code
I - 10	<b>Moves</b>	how to move data about
I - 13	<b>Arithmetic</b>	manipulating data on the 8080
I - 17	<b>Control</b>	jumps and process control
I - 19	<b>Techniques</b>	8080 for grownups
I - 23	<b>Summary</b>	A-Natural in brief

### II. Programming Utilities

II - 1	<b>Introduction</b>	the programming utilities
II - 2	<b>Conventions</b>	using the utilities
II - 8	<b>ROM</b>	writing read-only code
II - 9a	<b>anat</b>	A-Natural translator
II - 10	<b>as.80</b>	A-Natural assembler for 8080
II - 12	<b>c</b>	multi-pass command driver
II - 15	<b>cpm</b>	maintain CP/M diskettes
II - 17	<b>db</b>	binary file editor/debugger
II - 22a	<b>dis80</b>	translate A-Natural relocatable files
II - 23	<b>hex</b>	translate object file to ASCII formats
II - 26	<b>lib</b>	maintain libraries
II - 29	<b>link</b>	combine object files
II - 33	<b>lord</b>	order libraries
II - 35	<b>p1</b>	parse C programs
II - 37	<b>p2.80</b>	generate code for 8080 C programs
II - 39	<b>pp</b>	preprocess defines and includes
II - 41	<b>prof</b>	produce execution profile
II - 43	<b>ptc</b>	Pascal to C translator
II - 45	<b>rel</b>	examine object files

### III.a. Idris System Interface Library

III.a - 1	<b>Interface</b>	to 8080 Idris system
III.a - 3	<b>Conventions</b>	Idris system subroutines

III.a - 5	c80	compile and link C programs
III.a - 6	pc80	compile and link Pascal programs
III.a - 7	Crt	C runtime entry
III.a - 8	Crtpr	set up profiling at runtime
III.a - 10	_pname	program name
III.a - 11	close	close a file
III.a - 12	create	open an empty instance of a file
III.a - 13	exit	terminate program execution
III.a - 14	lseek	set file read/write pointer
III.a - 15	onexit	call function on program exit
III.a - 16	onintr	capture interrupts
III.a - 17	open	open a file
III.a - 18	read	read from a file
III.a - 19	remove	remove a file
III.a - 20	sbreak	set system break
III.a - 21	uname	create a unique file name
III.a - 22	write	write to a file
III.a - 23	xecl	execute a file with argument list
III.a - 24	xecv	execute a file with argument vector

### III.b. CP/M System Interface Library

III.b - 1	Interface	to CP/M system
III.b - 3	Conventions	CP/M system subroutines
III.b - 5	c	compiling C programs
III.b - 6	pc	compiling Pascal programs
III.b - 7	ld	linking a C program
III.b - 8	chdr	C runtime entry
III.b - 9	_main	setup for main call
III.b - 10	_pname	program name
III.b - 11	close	close a file
III.b - 12	cpm	call CP/M or CDOS system
III.b - 13	create	open an empty instance of a file
III.b - 14	exit	terminate program execution
III.b - 15	lseek	set file read/write pointer
III.b - 15a	onexit	call function on program exit
III.b - 16	onintr	capture interrupts
III.b - 17	open	open an existing file
III.b - 18	read	read characters from a file
III.b - 19	remove	remove a file
III.b - 20	sbreak	set system break
III.b - 21	uname	create a unique file name
III.b - 22	write	write characters to a file

### III.c. ISIS-II System Interface Library

III.c - 1	Interface	to ISIS-II system
III.c - 3	Conventions	ISIS-II system subroutines
III.c - 4	_main	setup for main call
III.c - 5	_pname	program name
III.c - 6	close	close a file
III.c - 7	create	open an empty instance of a file
III.c - 8	exit	terminate program execution
III.c - 9	isis	call ISIS-II
III.c - 10	lseek	set file read/write pointer
III.c - 10a	onexit	call function on program exit
III.c - 11	onintr	capture interrupts
III.c - 12	open	open an existing file
III.c - 13	read	read characters from a file
III.c - 14	remove	remove a file
III.c - 15	sbreak	set system break
III.c - 16	uname	create a unique file name
III.c - 17	write	write characters to a file

### IV. Machine Support Library for 8080

IV - 1	Conventions	using the 8080 Machine Support Library
IV - 3	c.btou	unpack bits to unsigned
IV - 4	c.count	counter for profiler
IV - 5	c.dadd	add double into double
IV - 6	c.dcmp	compare two doubles
IV - 7	c.dcpy	copy double to double
IV - 8	c.ddiv	divide double into double
IV - 9	c.dmul	multiply double into double
IV - 10	c.dneg	negate double
IV - 11	c.dsub	subtract double from double
IV - 12	c.dtd	move double to double
IV - 13	c.dtf	convert double to float
IV - 14	c.dti	convert double to int
IV - 15	c.dtl	convert double to long
IV - 16	c.dtr	convert double to int on stack
IV - 17	c.ent	enter a C function
IV - 18	c.ents	save registers on entering a C function
IV - 19	c.entx	save registers and check stack on entering a C function
IV - 20	c.ftd	convert float to double
IV - 21	c.idiv	divide integer by integer
IV - 22	c.ihl	jump on hl
IV - 23	c.ilsh	integer left shift
IV - 24	c.imod	remainder of integer divided by integer
IV - 25	c.imul	multiply integer by integer
IV - 26	c.irsh	integer right shift
IV - 27	c.itd	convert integer to double
IV - 28	c.ladd	add long to long

IV - 29	c.land	and long into long
IV - 30	c.lclt	compare long to long, set NC
IV - 31	c.lcmp	compare long to long, set Z
IV - 32	c.lcom	complement long
IV - 33	c.lcopy	copy long to long
IV - 34	c.ldiv	divide long by long
IV - 35	c.llsh	long left shift
IV - 36	c.lmod	remainder of long divided by long
IV - 37	c.lmul	multiply long by long
IV - 38	c.lneg	negate long
IV - 39	c.lor	or long into long
IV - 40	c.lret	return from runtime function
IV - 41	c.lrsh	long right shift
IV - 42	c.lsub	subtract long from long
IV - 43	c.ltd	convert long to double
IV - 44	c.lxor	exclusive or long into long
IV - 45	c.r0	the double accumulator and other pseudo registers
IV - 46	c.repk	repack a double number
IV - 47	c.ret	return from a C function
IV - 48	c.rets	return from a C function
IV - 49	c.switch	perform C switch statement
IV - 50	c.udiv	divide unsigned by unsigned
IV - 51	c.uldiv	unsigned divide long by long
IV - 52	c.ulmod	remainder of unsigned long divided by long
IV - 53	c.ulrsh	unsigned long right shift
IV - 54	c.ultd	convert unsigned long to double
IV - 55	c.umod	remainder of unsigned divided by unsigned
IV - 56	c.unpk	unpack a double number
IV - 57	c.ursh	unsigned right shift
IV - 58	c.utd	convert unsigned to double
IV - 59	c.utob	pack unsigned into bits
IV - 60	c.zret	return from runtime compare function
IV - 61	in	input from port
IV - 62	out	output to port