C CROSS COMPILER

XENIX TO HD64180/Z80

INSTALLATION GUIDE

VERSION 3.32

OCTOBER 1989

NOTICE

LIMITED WARRANTY

The liability of Cosmic, Sarl and Whitesmiths, Ltd. for the programs in this package shall be limited to the replacement of media which is defective in manufacture or recording. If such media is returned directly to Whitesmiths, Ltd. within ninety (90) days of purchase it will be replaced without cost.

SUCH WARRANTY IS IN LIEU OF ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL WHITESMITHS, LTD., BE LIABLE FOR ANY INCIDENTAL, CONSEQUENTIAL OR SPECIAL DAMAGES, INCLUDING BUT NOT LIMITED TO, ANY LOSS OF BUSINESS, INCOME OR PROFITS, EXPENSES INCURRED FOR TIME WHEN THE SYSTEM IS NOT OPERATIONAL, AND ANY LABOR COSTS RELATING TO OR ARISING OUT OF THE PERFORMANCE, FUNCTIONING OR USE OF THE SYSTEM.

THE PROGRAMS ARE FURNISHED ON A STRICTLY "AS IS" BASIS, and such limitation of warranty has been taken into account in determining the price of this package.

TABLE OF CONTENTS

I. INTRODUCTION

This package (3.32 cxidr86 64180/z80) is the Cosmic C Cross Compiler hosted on PC running XENIX, targeting the HD64180/Z80 processor. It consists of: compiler components, an assembler, a linker, libraries, utilities, and scripts for the compiler driver.

In order to install and operate this C cross package, you must provide the following:

1) a XENIX computer

2) a floppy drive

Your Cosmic C Cross Compiler is contained on one floppy disk in tar format. The following manuals are also included.

1) cxidr86 64180/z80 Installation Guide (the manual you are reading now)

2) C Language Specification

3) Cross Compiler Users' Guide for HD64180/Z80

Check your package to ensure that the floppy and all the manuals have been included. Before you proceed with the installation, read the entire installation guide, as well as the scripts that are part of the installation.

Note: If you already have a 3.x Whitesmiths compiler installed on your system, you will already have a **progenv** installed in your system directories. You should be aware that this package also includes a file called **progenv.** You must integrate this file yourself and edit the **Instal** script so that it will copy the modified **progenv** to the system directories.

Cosmic has also packaged an optional Pascal compiler for the cxidr86 hd64180/z80 compiler. The Pascal compiler is available as an add-on product to the basic C cross compiler package and should be installed after the cxidr86 hd64180/z80 package.

II. PREPARING FOR INSTALLATION

Installing the cross compiler package consists of two basic steps: extracting the floppy, and running a script to install the package.

The command lines that you type during the installation will be highlighted in boldface. All typed lines are terminated with a C/R or ENTER, although this will not be stated explicitly.

The installation script, called Instal will automatically move the compiler passes, libraries, and text files from a scratch directory to their appropriate system directories.

To avoid name conflicts with XENIX standard Utilities, the pr utility is installed as pr.33.

III. INSTALLATION

This section describes the steps you should follow to install the C Cross Compiler on your XENIX system. The exact commands you will type differ depending on the programming environment you choose to install as the default runtime environment. Descriptions of the three available environments and a detailed discussion of the considerations that will determine your choice of which to install as the default are in section 3.4 of the C Language Specification entitled "Runtime Environments". Whitesmiths, Ltd. strongly recommends that you read this section all the way through before proceeding with the installation.

To install the cxidr86 64180/z80 C Cross Compiler, perform the following steps:

1) Create a temporary directory and make it the current directory by typing:

```
# mkdir temp
# cd temp
```

2) Insert the floppy in the floppy drive, and then type:

```
# tar -xvf /dev/fd0
```

3) Start the installation script by typing:

```
# Instal
```

4) The installation script will automatically compile the test program acia.c, which is included with your package.

5) The temporary directory **temp** now contains all the original contents of the floppy, as well as all the sources of the libraries. You should move the source to some non-volatile place and remove **temp**.

The **.pro** file corresponding to the runtime libraries you choose to install must be copied to c.pro, which is the name under which the compiler will look for it. This step is performed automatically by the script **progenv** which copies the appropriate prototype file to **c.pro** in the current directory. The most convenient place to keep the copy of c.pro that you will be using on a regular basis is in the **/usr/bin** directory, where it can be accessed for anywhere on the system.

6) To copy the runtime environment and corresponding prototype file, type:

```
# progenv
```

IV. USING THE DOCUMENTATION

In the preceding sections of this document, you were given the minimum amount of information and explicit instructions required to install the cxidr86 64180/z80 compiler. From this point on, you will be deciding how to best use the compiler to meet your specific needs. To assist you in that effort, this section describes the manuals which you received in your package and provides some pointers on how to use them.

The manual you are reading now, the cxidr86 64180/z80 Installation Guide, leads you through the installation process and contains information specific to type cxidr86 64180/z80 implementation of the Cosmic C compiler. In addition to the previous sections, there is also a Section V, following this one, describing files you receive with the cxidr86 64180/z80 C compiler.

The C Language Specification describes the C programming language and the various library routines that make up the machine independant C environment.

The Cross Compiler User's Guide for HD64180/z80 describes the cross assembling utility and the programming utilities that make up the compiler and features tutorials showing how to use them.

V. FILES INCLUDED WITH THIS PACKAGE

The following is a list of the files that are included in this package:

COMPILER AND UTILITIES

c	the C compiler driver
clist	the utility which print the C source lines with the addresses of the C sources lines.
cp180	the C parser
cp280	the HD64180/Z80 code generator
cp380	the HD64180/Z80 code optimizer
cpp80	the C preprocessor
hex80	the utility to convert binary files to hex records
lby	the librarian
lines	the utility to extract line number
lm	the listing merge utility
lnk80	the linker
lord80	the library ordering utility
pr	the print utility
prdbg	the tabulate debug utility
rel80	the relocatable object module inspector
toprom	the utility to copy data to initialize in PROM
unhex	the utility to convert hex records to binary
x80	the HD64180/Z80 cross-assembler
Instal	script to install the compiler
instal.doc	the English Installation Guide version
install.doc	the French Installation Guide version
progenv	script to set up a c.pro programming environment
Test	script to test the compiler
cx80. pro	proto file for cxidr86 compiler for HD64180/Z80

STARTUP AND LIBRARIES

Crts.80	the C runtime startup module
Crtsrom.80	the C runtime startup module with initialization
libi.180	the Standard library for HD64180
libm.180	the Machine library for HD64180
libf.180	the floating point library for HD64180
libd.180	the double library for HD64180
libi.z80	the Standard library for Z80
libm.z80	the Machine library for Z80
libf.z80	the floating point library for Z80
libd.z80	the double library for Z80

HEADERS

math.h	mathematical functions header
setjmp.h	setjmp/longjmp header

stdarg.h	variable argument list header
stdio.h	standard I/O header
stdlib.h	general library functions header
string.h	string types header
wslxa.h	the Extended ANSI header

OTHER FILES

Crts.s	the C runtime startup module source
Crtsrom.s	the C runtime startup module source with initialization
acia.c	example program with interrupt handling
cright	Cosmic copyright notice
limit	notice of limited warranty

SOURCES LIBRARIES

LIBD and LIBF

acos.c	cnorm.c	exp.c	poly.s	sqrt.c
addexp.s	cosh.c	fabs.c	pow.c	sscanf.c
asin.c	cprint.c	fabs.s	printf.c	tan.c
atan.c	cscan.c	floor.c	round.c	tanh.c
atof.c	dtento.c	fmod.c	scanf.c	unpack.s
btod.c	dtoe.c	ftrunc.c	sin.c	
ceil.c	dtof.c	huge.s	sinh.c	
cfcon.c	dtog.c	log.c	sprnt.c	

LIBI

abs.s	iscntr.s	jmp.s	rand.c	strlen.s
atoi.c	isdig.s	malloc.c	sbreak.c	strncm.s
atol.c	isgraf.s	memchr.s	scanf.c	strncp.s
calloc.c	islow.s	memcmp.s	sprnt.c	strnct.s
cprint.c	isprnt.s	memcpy.s	sscanf.c	strpbr.s
cscan.c	ispunc.s	memset.s	strcat.s	strrch.s
getchr.c	isspac.s	printf.c	strchr.s	strspn.s
gets.c	isup.s	putchr.c	strcmp.s	tolower.s
isalnm.s	isxdig.s	putchr.s	strcpy.s	toupper.s
isalph.s	itob.c	puts.c	strcsp.s	

LIBM

bbtou.s	dtf.s	idiv.s	libc.s	oftl.s
btou.s	dtl.s	ilsh.s	llsh.s	pustr.s
butob.s	fadd.s	imul.s	lmul.s	repk.s
bysw.s	fcmp.s	imulz80.s	lmulz80.s	sav.s
cregs.s	fdiv.s	jltab.s	lneg.s	sav0.s
dadd.s	fmul.s	jtab.s	lor.s	savs.s
dcmp.s	fmvl.s	ladd.s	lsub.s	savs0.s
ddiv.s	frepk.s	land.s	ltd.s	toram.s
dmul.s	fsub.s	lclt.s	ltf.s	unpk.s
dmvl.s	ftd.s	lcmp.s	lxor.s	utob.s
dneg.s	ftl.s	lcom.s	movstr.s	zret.s
dsub.s	funpk.s	lcpy.s	mvl.s	

dtd.s ibc.s ldiv.s nftl.s

VI. USING THE CROSS COMPILER

This section explains briefly how to use your compiler package. Manual pages describing the relevant utilities are found in your Cross Compiler Users' Guide for HD64180/Z80.

Compiling and linking on the host machine is done as follows:

```
# c -o prog1 -proto cx80 prog1.c
```

or if you have rename the cx80.pro to c.pro

```
# c -o prog1 prog1.c
```

Note: The path for searching for prototype files is the one specified by your shell. If your **.pro** files are not in one of these directories, the c driver will not be able to find them. They will reside in /usr/bin at the conclusion of the installation procedure. However, the driver is easiest to invoke from within the current directory.

It is also possible to specify the fully qualified name of the **.pro** file via the **-proto*** flag (i.e. **-proto /user/privat/mydir/c.pro**).

When running the **pr.33** utility and sending the resulting output to a line printer, the following two items should be kept in mind:

1) pr.33 generates n lines of output per page, where n defaults to 66. pr expects to have all n lines as input to itself. Line printer drivers which insert additional lines at the top or the bottom of the page may necessitate passing a different -l# flag to pr.33

2) pr.33 takes care of all top-of-form control, so that printers should be set up not to insert form feeds.