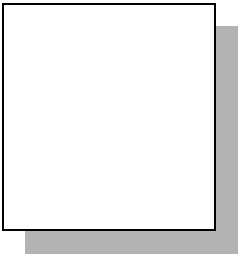


Installation for Win 95/98/NT

This chapter describes how to install the Cosmic *C Cross Compiler for MC68HC11* on your Win 95/98/NT host system and how to run the test script that verifies the installation procedure. This chapter is composed of:

- Preparing for Installation
- Running the Installation Program
- Directory Structure
- Setup Search Paths
- Verifying Installation
- Using the Compiler

Preparing for Installation

This package is the Cosmic C Cross Compiler hosted on PC and compatible running Windows 95/98 or NT 4.0 and targeting the MC68HC11 processor. The package consists of: compiler components, an assembler, linker, libraries, utilities and an example.

In order to run the compiler, your system must meet the following minimum hardware and software requirements:

- PC with an 80386 or better microprocessor
- MS-Windows 95/98 or Windows NT 4.0 operating system
- CD-ROM drive or 3.5” High Density 1.44 Mb diskette drive
- Hard disk drive with at least 10 Mb of free space
- 8 Mb of RAM

Installation media

Your Cosmic C cross compiler is shipped on a CD-ROM or on double-sided, high density 3.5 inch diskettes. The label identifies the product, the product version number and the license serial number. The manual “*Cross Compiler Users' Guide for MC68HC11*” is also included:

You should check your package to ensure that the CD or the diskettes and the manual have been included. Before you proceed with the installation, read the entire installation guide, as well as the on-screen instructions provided during the installation.

Installation process

In the installation instructions that follow, we assume that your CD disk drive is designated by **D:**, the floppy disk drive is designated by **A:** and your hard disk partition by **C:**. If your system uses different designations, you should adjust the installation instructions accordingly.

The Compiler is installed by an *Installshield Setup* utility program. Throughout the installation procedure, there is an assumed default directory in which the Compiler will be installed. This directory is

C:\COSMIC\CX11. If you choose to install the Compiler in a different directory or on a different hard disk drive, you must substitute your specified location wherever you see **C:\COSMIC\CX11**.

The Compiler directory must be in your system command search path (see documentation regarding the **PATH** command in your Windows-manual). The utilities provided with the compiler may have names that conflict with the names of utilities from other vendors. If a conflict occurs, add the Compiler directory before all other vendor directories in your system's command search path.

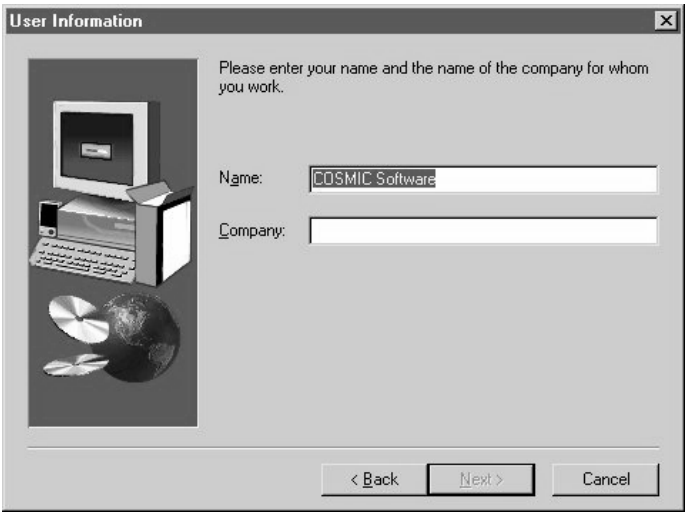
Running the installation program

1) Insert the Compiler CD-ROM into your CD-driver or the **disk 1** into your floppy disk drive.

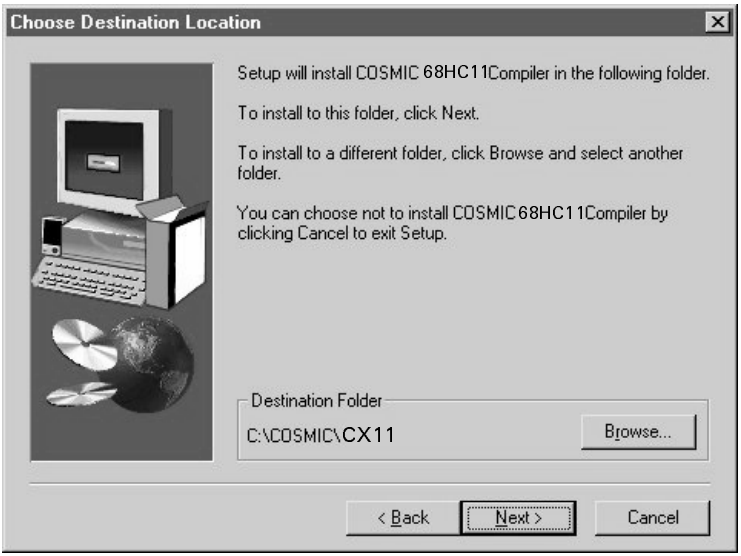
2) Open the Windows Explorer and select the appropriate CD-ROM or floppy disk drive letter by double clicking on it.

3) Double click on *Setup.exe* to run the Compiler installation and setup program. As an alternative to steps 2 and 3 you can use the **Run** command from the Windows Start Menu and type **`D:\SETUP`** (or **A:\SETUP**) to run the installation program.

4) Follow the on-screen instructions provided by the installation. Read the license agreement and if you agree to it click on **Yes** and fill out the User information screen including your serial number and click next.



5) The next window allows you to choose the **Destination** Folder for the Compiler using the **Browse** button. Click on Next to use the selected folder.



The installation will now copy and configure the compiler on your system. After all of the files are copied, the installation will check your system to see if the Compiler folder selected is already in your Command Execution path. If it is not in your PATH, you will be given the option to have it added. If you choose to let the install modify your PATH you will need to reboot for the change to take effect.

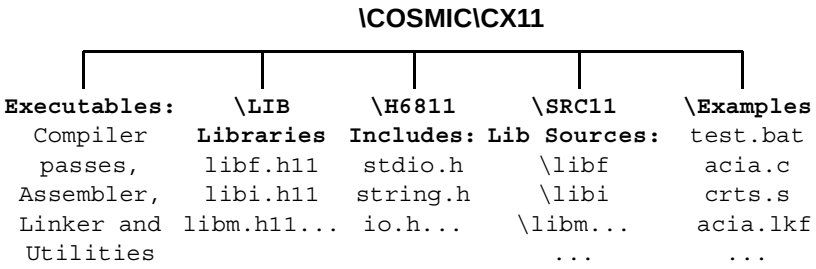
If you are installing under Windows NT you will be given a choice of registering the Compiler Path in the Current User or the Local Machine section. If you register the compiler in the Local Machine section (Requires Administrator privileges), the compiler will be available to all user profiles.

NOTE

*The Compiler folder must be in your system command search path for the compiler to work properly (see documentation regarding the **PATH** command in your Windows manual).*

Directory Structure

The following is the directory structure for the compiler installation on your hard disk. The default directory of **COSMIC\CX11** is shown as the root for the installation. If you chose to install the compiler in another location substitute your selected path for "**\COSMIC\CX32**".



Library Source Code

The Source code to the compiler libraries are copied by the installation into several subfolders as shown below. Each folder contains all of the source code used to create each library and a batch or script file to rebuild each library. See the linker chapter for information on Library usage.

Folder Name	Library Source Code
LIBD	Double Precision Floating Point Library
LIBF	Single Precision Floating Point Library
LIBI	Integer Only Library
LIBM	Machine Library

Setup Search Paths

Include Files

The search process for compiler include files may be specified 3 different ways listed in descending order of priority.

1) The compiler first searches all explicit **-i** options specified on the command line starting with the first (leftmost) **-i** option. For example:

```
CX6811 -i"PATH1" -i"PATH2" -vl test.c.
```

2) Next the compiler searches any **-i** options specified in the Configuration file (*e.g.* **CX6811.cxf.**) from top to bottom.

3) If any of the include files are still not found, the compiler will search the system environment variable **CX6811** if it exists. This symbol may contain several paths separated by the usual path separator of the host operating system (*i.e.* '**;**' for DOS and '**:**' for UNIX). *e.g.*:

```
CX6811=c:\compiler\include\path1;d:\include\path2.
```

The search process for assembler include files is very similar to the search process for C includes. The following is the process listed in descending order of priority.

1) The assembler first searches all explicit **-i** options specified on the command line starting with the first (leftmost) **-i** option. For example:

```
CA6811 -i"PATH1" -i"PATH2" test.s
CX6811 -ai"PATH1" -ai"PATH2" test.s
```

2) Next the assembler searches the **-ai** options in the Configuration file (*e.g.* **CX6811.cxf**.) from top to bottom.

3) If any of the include files are still not found, the compiler will search the system environment variable **CXLIB** if it exists. This symbol may contain several paths separated by the usual path separator of the host operating system ('**;**' for DOS and '**:**' for UNIX). *e.g.*: **CXLIB=c:\assembler\include\path1;c:\include\path2**.

Libraries

The linker (**CLNK**) uses the environment variable **CXLIB** to search for objects and library files. If you don't specify the full path to the objects and/or libraries in the link command file AND they are not found in the local directory, the linker will then search all paths specified by the **CXLIB** environment variable. This allows you to specify just the names of the objects and libraries in your link command file.

For example, setting the **CXLIB** environment variable to the **C:\COSMIC\CX32\LIB** directory is done as follow:

```
C>set CXLIB=C:\COSMIC\CX32\LIB
```

Verifying Installation

A simple program which handles input/output with interrupts is provided as an example:

- **acia.c** test program
- **acia.lkf** link file
- **vector.cs** file containing the definition of the interrupt vectors.

When the installation of the compiler is completed, go to the **C:\COSMIC\CX11\EXAMPLES** folder and double click on **TEST.BAT** to run the test batch file. This file compiles and links two C source files and generates a hex image. If the compiler is installed properly, the batch file should return the message "*The Compiler Installation is Successful*". If you do not receive this message, try reinstalling the compiler and let the installation modify your Command execution path and reboot your system.

Using The Cross Compiler

This section explains briefly how to configure and use your compiler package. Detailed information describing compiler options and usage is provided in your "*Cross Compiler Users' Guide for MC68HC11*". The Cosmic compiler is designed to be very flexible. It can be used with virtually any editor, IDE, Make utility, Source Code Control System or use it directly from a Command shell. For seamless integration, choose

the Cosmic Integrated Development Environment (**IDEA**) or Premia's Codewright Editor.

Installing IDEA

IDEA is Cosmic's own integrated development environment that provides a high level interface to the compiler and debugger products. **IDEA** includes a Windows editor, program analyzer, project manager, option builder, integrated error checking and link manager. **IDEA** is automatically installed with the compiler, unless it has been sold separatly. In this case, to install **IDEA**, run setup on disk one of the Cosmic IDEA distribution. Refer to the "*IDEA User's Manual and Quick Start Guide*" for details. The installation program will ask for the location where the compiler is installed so the compiler **must** be installed first.

Configuring Codewright

To configure the Cosmic compiler for use with Codewright first install Codewright and the Compiler and then locate the free disk titled "*Cosmic Codewright Integrator*" or download it from the Cosmic web site. Run *setup.exe* to install and configure the error parser DLL, GNU Make and example projects. See the file "*readme_cw.pdf*" for more information.

Command Shell

The compiler may also be used from a Command or Shell prompt on the host machine. You can run each compiler component separately or use a make, batch or script file. The following is a simple compiler command line which will compile and assemble the file prog1.c and produce a relocatable object (*prog1.o*) and a listing file (*prog1.ls*). For details on compiler usage and options refer to the "*Compiler User's Manual*".

```
C>cx6811 -vl prog1.c
```