

Introducing High C/C++

This chapter presents an overview of the High C/C++ compiler for the PowerPC. It contains the following sections:

§1.1: *About the C Language*

§1.2: *About the C++ Language*

§1.3: *About the High C/C++ Compiler*

§1.4: *Tools and Utilities Used with High C/C++*

§1.5: *Standards, Conformance, and Portability*

1.1 About the C Language

C is a mixed-level systems language designed by Dennis Ritchie at AT&T's Bell Laboratories. The C language grew in popularity because of its use in implementing the UNIX operating system, its elegant (and deceptive) simplicity, and its close-to-the-machine features. Many software developers now use C for applications as well as for systems software.

Later implementations of C were extended to add enumeration types and a few other features. More recently, many extensions have been proposed to make C a safer language while still being consistent with the philosophy of the original language.

1.1.1 High C Extensions to Standard C

High C includes ANSI Standard C and also provides extensions carefully designed to be consistent with the philosophy of C. Incompatibilities were minimized by introducing a minimum of new keywords and by retaining the original syntax. The extensions are flagged when the compiler is run in ANSI mode.

“Lint”-like checking In addition, the compiler provides many checking features usually available only in a separate “lint” program. You get both efficiency *and* reliability.

Important High C language extensions include the following:

- nested functions complete with up-level references, as in Pascal
- nested functions passable as parameters to other functions, as in Pascal
- intrinsic functions, such as `_abs()`, `_min()`, and `_max()`, for efficiency
- iterators: a powerful construct for traversing data structures
- in-line versions of ANSI library function calls, such as `strlen()`, `memset()`, and `memcpy()`

1.2 About the C++ Language

Developed by Bjarne Stroustrup at AT&T's Bell Laboratories in the early 1980s, C++ is largely a superset of the C language. C++ is a general purpose language that provides support for data abstraction and object-oriented programming while retaining C's simplicity of expression and speed of execution.

Full implementation of C++ High C/C++ integrates C with incremental strengths of C++. High C/C++ is a full implementation of the C++ language as defined in the **Annotated C++ Reference Manual** by Ellis and Stroustrup.

1.3 About the High C/C++ Compiler

The `hc` driver invokes the High C/C++ compiler, the ELF assembler, or the ELF linker (depending on the input file-name extension).

Typically, the driver associates the following file types with file-name extensions:

- `.c` C source file
- `.cpp` C++ source file
- `.s` Assembly source file
- `.o` object file

See Table 2.1, *Default File-Name Extensions*, for a complete list of the file-type assumptions associated with file-name extensions.

1.4 Tools and Utilities Used with High C/C++

This section lists tools and utilities (other than the compiler) supplied with the High C/C++ toolset. For additional information, consult the appropriate manual in the document set.

The ELF assembler reads PowerPC assembly-language files and generates relocatable object files conforming to the Executable and Linking Format (ELF).

The ELF linker/locator links ELF-conformant object files.

The ELF archiver groups independently developed object files into an archive library to be accessed by the linker.

Additional stand-alone utilities:

- **elf2hex** converts binary to an ASCII hexadecimal format.
- **elfdump** dumps information from ELF object files and executables.

1.5 Standards, Conformance, and Portability

ANSI C High C/C++ supports ANSI Standard C (ANSI document X3.159) and several extensions. High C programs can be compiled with an ANSI option that turns off the extensions and reduces the language to the ANSI C Standard core.

You can upgrade ANSI Standard C programs to High C gradually by *not* choosing the ANSI option, and by adding the extensions as required.

C++ MetaWare is participating in the ANSI committee's formal definition of the C++ language. High C/C++ tracks the ANSI X3J16 working draft as it progresses.

IEEE High C/C++ conforms to the IEEE 754 standard (1985) for binary floating-point arithmetic, to the extent necessary to meet the ANSI C/C++ requirement for floating-point arithmetic and logical operations.

Additional Information See *Specifications and ABI Documents* in the *About This Book* chapter for more information.