2 *Invoking the Compiler*

The iWarp C compiler is an optimizing compiler for the iWarp processor. The iWarp C compiler behaves much like the Unix C compiler, but it is a cross-compiler, running on the Sun host and producing code for the iWarp system. The iWarp C compiler translates C programs into iWarp executable load modules or iWarp relocatable binary programs for loading with the iWarp linker. The iWarp C compiler produces object code files in Common Object File Format (COFF).

The Compilation Command: iwcc

To invoke the iWarp C compiler, use the iwcc command as follows:

`iwcc [`*`option`*` ...]  `*`filename`*` ...`

filename can be any of the following:

- A C source file (.c file extension).
- A C source file that is not to be run through the preprocessor (.i file extension).
- An assembly language source file (.s file extension). These files are passed to the assembler and an object file is produced.
- An intra-file inlined intermediate file (.K file extension). These files are used by the global optimizer for inter-file inlining of any specified C programs.

All other files are passed to the linker.

option can be any of the options listed in the following section.

iwcc Options

-#	List each phase of compilation being executed and the command options used to invoke the phase.
-##	Same as -#, but also prints the full path used to load each phase of the compilation.
-###	Same as -##, but do not execute any phases of the compilation.
-c	Suppress the link edit phase of the compilation and produce an object file for each source file.
-C	Prevent the C preprocessor from removing comments.
-D*name* [=def]	Define *name* for the preprocessor as if by a #define directive. If =def is not included, *name* is defined as 1. If both the -D and -U options are used with the same *name* , *name* will be undefined.

-E	Run only the preprocessor on the named C programs and send the result to standard output. The output includes cpp-type line number information.
-fasmwiden	Convert floating-point operands to double-precision operands for asm insert function parameters and function return types.
-fnoasmwiden	Suppress the conversion of floating-point operands to double-precision operands for asm insert function parameters and function return types. This is the default.
-fnowiden	Suppress the automatic conversion of floating-point operands to double-precision operands within expressions, and suppress the conversion of floating-point parameters to functions and function return types. This option is not recommended, and there is no library support for float parameters or function returns.
-fsingle	Suppress the automatic conversion of floating-point operands to double-precision operands within expressions.
-g	Generate information for symbolic debugging. This suppresses all optimization.
-G*c*	Retain intermediate files from pass *c* for debugging, where *c* is one of the following: p preprocessor 0 compiler first pass q PCO pass 1 compiler second pass a assembler l link editor m listing merger
-I*pathname*	Add *pathname* to the list of directories in which to search for #include files with relative pathnames (not beginning with /). The preprocessor first searches for #include files in the directory containing the source file, then in directories named with the -I option, then in /iwarp/include.
-K	Invoke PCO to generate an intra-file inlined intermediate file. Arguments whose names end with .c are compiled, intra-file inlined by PCO, and each intermediate file is left in a file whose name is that of the source file with .K substituted for .c. The output file name cannot be changed with the -o option.
-l*x*	This option is an abbreviation for the library name lib*x*.a, where *x* is a string. This option is used with the -L option.
-L*dir*	Add *dir* to the list of directories containing object library routines. Directories specified with -L are searched before the standard directory, /iwarp/lib. This option only affects -l*x* options which follow on the command line.

-M	Merge the C and generated iWarp assembler files into a file suffixed .m. For a single file, the output file name can be changed with the -o option.
-MM	Same as -M, but includes the text of #include files in the listing file.
-N*letter*	*letter* can be C, L, or S. C suppresses the automatic use of a default link control file. L suppresses automatic linking with libc.a (and libg.a if -g is used). S suppresses the automatic linking with iwcrt0.o (or iwmcrt0.o if -p is used).
-o *filename*	Specifies *filename* as the name of the object file, overriding the default name a.out. Use with -c modifies the name of the object file. Use with -P modifies the name of the preprocessed file. Use with -S modifies the name of the output assembly file. Use with -M modifies the name of the output merged listing file.
-O[*level*]	Invoke optimization as indicated by *level*, where *level* is one of the following: 0 — no optimizations 1 — global optimizations on intermediate code (PCO) 2 — basic block compaction generation of Compute and Access instructions, as well as PCI and PCO optimizations 3 — software pipelining, basic block compaction, and PCO optimizations If *level* is not specified, this option will invoke all available optimizations.
-p	Produce code that counts the number of times each routine is called. If link editing takes place, replace the standard startoff routine by one that automatically calls the monitor at the start and arranges to write a mon.out# file at normal termination of execution of the object program (# is the cell number). This allows generation of an execution profile with the iwprof command.
-P	Run only the preprocessor on the specified C programs and leave results in the corresponding files suffixed .i. For a single file, the output file name can be changed with the -o option.
-Q*options*	Invoke optimizations as defined by *options*, where *options* can be any or all of the following: G — invoke global optimizations (PCO) C — compaction of basic block for Compute and Access instruction S — software pipelining If compile time is an issue, but you want to do some optimization, use the -O*level* option to reduce the amount of optimization done.

-S	Compile the specified C programs and leave the assembler-language output in corresponding files suffixed .s. For a single file, the output file name can be changed with the -o option.
-T	Causes the compiler to retain all intermediate files in the same directory as the source code.
-u*name*	Causes the linker to enter *name* as an undefined symbol in the symbol table.
-U*name*	Remove any initial definition of *name* , where *name* is a reserved symbol predefined by the preprocessor.
-V	Display the version of the C compiler and all invoked passes, and perform compilation as specified.
-VV	Display the version of the C compiler and all possible passes, but do not perform compilation.
-Wc, *arg1* [, *arg2* ...]	Hand off any specified arguments to pass *c*, where *c* is one of the following:

p	preprocessor
0	compiler first pass
q	PCO pass
1	compiler second pass
a	assembler
l	link editor
m	listing merger

-x	Preserve only global (non-.globl) symbols in the output symbol table. Enter only external symbols.
-Yc, *dir*	Get *c* from directory *dir* , where *c* is one of the following:

p	preprocessor
0	compiler first pass
q	PCO pass
1	compiler second pass
a	assembler
l	link editor
m	listing merger
C	link.ctl
L	default search directory for libraries (/iwarp/lib)
S	startup code (iwcrt0 or iwmcrt0)

Files

The following sections list files that are part of the compilation process. These files include:

- input files
- intermediate and output files
- elements of the compiler

Input Files

file.a	library (of archived object code)
file.c	C source file
file.ctl	linker control file
file.K	intra–file inlined intermediate file (previously produced with –K)
file.o	object code
file.s	assembly language source file
/iwarp/lib/iwcrt0.o	runtime startup code
/iwarp/lib/iwmcrt0.o	profiled runtime startup code
/iwarp/lib/libc.a	C library
/iwarp/lib/libm.a	C math library
/iwarp/lib/libp/libc.a	profiled libc.a
/iwarp/lib/libp/libm.a	profiled libm.a

Intermediate and Output Files

a.out	executable output file
file.1	pass 1 output
file.i	preprocessor output (iwcpp(1) output)
file.I	assembly inlined file output by pass 1 and read by pass 2
file.K	intra–file inlined intermediate file (produced by –K)
file.m	C/asm listing merger output (iwmrg(1) output)
file.o	object file (assembler output)
file.q	PCO output
file.s	assembler source (pass 2 output, PCI optimizer input and output)

file.w	inline file (list of functions to inline)
file.x	noinline file (list of functions to not inline)
/tmp/ctm *	temporary files (a different directory can be used by setting the TMPDIR environment variable to the name of a directory)

Parts of the compiler

/iwarp/bin/iwcc	C compiler driver, iwcc(1)
/iwarp/lib/iwcpp	C preprocessor iwcpp(1)
/iwarp/lib/iwccom0	pass 1
/iwarp/lib/iwccom1	pass 2
/iwarp/lib/iwccomq	the global optimizer
/iwarp/lib/iwliw	LIW optimizer
/iwarp/lib/iwas	assembler script
/iwarp/bin/iwas	iWarp assembler iwas(1)