Appendix D: Error Messages

This following sections list the iWarp-specific C compiler error messages.

Error Messages for *send* and *receive*

"foo.c", line *nn*: argument out of range for '_receive*?*'

The *Channel* specified was not *GATE0* or *GATE1*. To correct, make sure that <stdio.h> is present in an #include and change the _receive*?* to specify either *GATE0* or *GATE1*.

"foo.c", line *nn*: argument out of range for '_send*?*'

The *Channel* specified was not *GATE0* or *GATE1*. To correct, make sure that <stdio.h> is present in an #include and change the _send*?* to specify either *GATE0* or *GATE1* as the first parameter.

"foo.c", line *nn*: bad argument for '_receive*?*'

The *Channel* is not specified as a constant *GATE0* or *GATE1*; variables cannot be used. To correct, make sure that <stdio.h> is present in an #include and change the _receive*?* to specify either *GATE0* or *GATE1*.

"foo.c", line *nn*: bad argument for '_send*?*'

The *Channel* is not specified as a constant *GATE0* or *GATE1*; variables cannot be used. To correct, make sure that <stdio.h> is present in an #include and change the _send*?* to specify either *GATE0* or *GATE1*.

"foo.c", line *nn*: warning: '_send*?*' argument has incompatible type

The second argument of the _send*?* does not match the type specified by the type designation '*?*'. The compiler coerces the parameter to the type expected. Apply an explicit type cast to expressions to coerce them silently to the type desired or use the correct type of send pseudo–function.

"foo.c", line *nn*: too few arguments for '_receive*?*'

Receives takes exactly one argument, none were supplied.

"foo.c", line *nn*: too many arguments for '_receive*?*'

Receive takes exactly one argument, too many were specified.

Error messages for *cond* and *condcall*.

"foo.c", line *nn*: argument out of range for '_cond'

The *condition–code* specified in the _cond was not a legal value. Only *condition–code* names defined by #include<iwarp_cc.h> or #include<sys/iwarp_math_cc.h> should be used.

"foo.c", line *nn*: argument out of range for '_condcall'

The *condition–code* specified in the _cond was not a legal value. Only *condition–code* names defined by #include<iwarp_cc.h> or #include<sys/iwarp_math_cc.h> should be used.

"foo.c", line *nn*: '_condcall' argument must be name of function that returns void

The *function–name* argument to _condcall must be explicitly declared (elsewhere) as a function returning void. There are no coercions to apply to a function name to make it acceptable if it is not declared as returning void.

"foo.c", line *nn*: too few arguments for '_cond'

Condition testing with _cond takes exactly one argument, none were supplied.

"foo.c", line *nn*: too few arguments for '_condcall'

Condition testing with _condcall takes exactly one argument, none were supplied.

"foo.c", line *nn*: too many arguments for '_cond'

Condition testing with _cond takes exactly one argument, too many were supplied.

Error messages for internal table size limits

"blocks.c", line *n*: compiler error: nesting too deep

The maximum nesting for block and loops has been exceeded by a block.

"big.c", line *n*: compiler error: too many instructions generated

A single tree, usually representing a single statement in the source code, generated too many instructions for the compiler to handle. This cannot happen when using the iWarp optimizer. Rewrite the program so no single statement is excessively complicated.

"case.c", line *n*: compiler error: switch table overflow

Too many cases (including the default) in a single switch.

"decl.c", line *n*: compiler error: too many arguments

Too many declared formal parameters in a function declaration.

"dim.c", line *n*: compiler error: dimension table overflow

Too many arrays, structures, union and enum declared.

"nest.c", line *n*: compiler error: whiles, fors, etc. too deeply nested

The maximum nesting for block and loops has been exceeded by a loop.

"sym.c", line *n*: compiler error: symbol table full

Too many symbols defined.

"tree.c", line *n*: compiler error: out of tree space, simplify expression

A single tree, usually representing a single statement in the source code, generated too many nodes in the internal representation for the compiler to handle. Rewrite the program so no single statement is excessively complicated.