Appendix A: Quick-reference of Commands

Invoking the iWarp Manual Page Tool: iwman

```
iwman [man_option] ...  filename ...
```

Invoking the C Compiler: iwcc

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

iwcc Options

-#	List each phase of compilation being executed and the command options used to invoke the phase.
-##	Same as -#, but also print 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.
-E	Run only the preprocessor on the named C programs and send the result to standard output.
-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.
-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.
-fsingle	Suppress the automatic conversion of floating-point operands to double-precision operands within expressions.
-g	Generate information needed for use of sdb.
-G*c*	Retain intermediate files from pass *c* for debugging.
-I*pathname*	Add *pathname* to the list of directories in which to search for #include files with relative pathnames (not beginning with /).
-K	Invoke PCO to generate an intra-file inlined intermediate file.
-l*x*	This option is an abbreviation for the library name lib*x*.a, where *x* is a string.
-L*dir*	Add *dir* to the list of directories containing object library routines.

-M	Merge the C and generated iWarp assembler files into a file suffixed .m.
-MM	Same as -M, but includes the text of #include files in the listing file.
-N*letter*	Suppress automatic linking. *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*	Override the default object file name. Specifies *filename* as the name of the object file, overriding the default name a.out.
-O	Invoke all available optimizers.
-p	Produce code that counts the number of times each routine is called.
-P	Run only the preprocessor on the specified C programs and leave results in the corresponding files suffixed .i.
-Q*options*	Invoke optimizations as defined by *options*, where *options* can be: C compaction of basic block for Compute and Access instruction G invoke global optimizations (PCO) S software pipelining
-S	Compile the specified C programs and leave the assembler-language output in corresponding files suffixed .s.
-u*name*	Cause 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 pass *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
S	startup code
L	default library directory

Invoking the Fortran Compiler: iwf77

```
iwf77 [option ...]  filename ...
```

iwf77 Options

-# List each phase of compilation being executed and the command options used to invoke that phase.

-## Same as -#.

-### List each phase of compilation and the command options used to invoke that phase, but do not execute.

-1 Compile DO loops so they are performed at least once if reached This is the same as -onetrip.

-66 Disable FORTRAN 77 and VMS F77 language features to enhance FORTRAN 66 compatibility.

-77 Disable VMS extensions.

-auto Make local variables automatic (non-persistent) by default. This option does not affect variables explicitly declared as persistent using SAVE, variables initialized with DATA statements, or variables equivalenced to persistent objects. This option is the default condition.

-c Suppress the link-edit phase of the compilation and produce an object file for each source file. For a single file, you can change the output file name with the -o option.

-C Generate code for run-time subscript range checking. If an array subscript is outside the declared array bounds, this option causes compiled code to abort with an error message.

-E*arg* Use *arg* as an EFL flag argument when processing a .e file.

-f *filename* Specify *filename* as an input file.

-F Run the EFL and RATFOR preprocessor on relevant files, and put the result in files with a .f file extension.

-g Generate additional information needed by the debugger. This option suppresses all optimizations.

-G*c* Stop after the specified pass and retain all intermediate files. *c* can be either of the following:

l	first pass
q	PCO pass

-iwarp — Specify that the code is to be executed on the iWarp system. This option is useful for makefiles.

-K — Invoke PCO to generate an intra-file inlined intermediate file. Arguments whose names end with .c or .f are compiled and intra-file inlined by PCO. Each intermediate file is left in a file whose name is that of the source file with .K substituted for .c or .f.

-l*x* — Link with the object library lib*x*.a.

-L*dir* — Add *dir* to the list of directories searched for libraries by the linker. Directories specified by the -L option are searched before the standard directory, /iwarp/lib.

-m — Apply the M4 preprocessor to each EFL or RATFOR source file before transforming with the efl or ratfor processors. This option can also be specified as -m4.

-M — Merge the C or FORTRAN and generated iWarp assembler files into a file suffixed .m. For a single file, you can change the output file name with the -o option.

-MM — Same as -M, but incorporate the text of included files in the listing file.

-N*letter nnn* — Change the size of the table defined by *letter* to *nnn*. The compiler provides a diagnostic message when a table overflows. Letter can be any of the following (the default table size, *nnn*, is included:

q	equivalences (150)
x	common blocks, subroutine, and function names (200)
s	statement numbers (2000)
c	depth of loops, if-then-elses (20)
n	variable and common block names (401)
l	labels for computed and assigned gotos and the number of alternate returns (125)

-onetrip — Compile DO loops so they are performed at least once if reached. Without -onetrip, FORTRAN DO loops are not performed at all if the upper limit is smaller than the lower limit.

-o *filename* — Specify *filename* as the name of the object file, overriding the default name a.out. When used with the -c, -S, or -M options, the default output file name will be overridden only if a single source file is specified.

-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, 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 startup routine with 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.
-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 blocks, generation of Compute and Access instructions S software pipelining, compaction of basic blocks, and generation of Compute and Access instructions If compile time is an issue, but you want to do some optimization, use the -O*level* option to reduce the amount of optimization.
-R*arg*	Use *arg* as a RATFOR flag argument when processing a .r file.
-S	Compile the specified FORTRAN 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.
-static	Make local variables persistent. This option does not affect variables explicitly declared as AUTOMATIC. Use this option only when it is necessary to run non-standard programs that require static variables. Use of this option negatively impacts optimization.
-T	Retain all intermediate files in the same directory as the source code.
-target iwarp	Specify that the compiled code will be executed on the iWarp system. This option is useful for makefiles.
-u	Make the default type of a variable undefined.
-U	Treat upper and lower cases as distinct.
-V	Display the version of the F77 compiler and all invoked passes, and perform compilation as specified.
-V*options*	Invoke the specified *options* during compilation. *options* can be any of the following. The default conditions are in bold typeface:

check	same as check=bounds, check=overflow, check=underflow
nocheck	same as check=nobounds, check=nooverflow, check=nounderflow
check=all	same as -Vcheck
check=none	same as -Vnocheck
check=bounds	check array subscript bounds

check=nobounds	do not check array subscript bounds
check=overflow	check for floating-point overflow
check=nooverflow	do not check for floating-point overflow
check=underflow	check for floating-point underflow
check= nounderflow	do not check for floating-point underflow
d_lines	compile lines with “d” or “D” in column 1
nod_lines	treat lines with “d” or “D” in column 1 as comments
debug	same as -Vd_lines
nodebug	same as -Vnod_lines
extend_source	extend the range of FORTRAN source to columns 1-132; lines over 132 characters will stop compilation
noextend_source	restrict source to columns 1-72
f77	compile as standard FORTRAN 77 code
nof77	compile using FORTRAN 66 interpretations in cases of incompatibility

-vms
Enable VMS extensions. This is the default.

-VV
Display the version of the F77 compiler and all possible passes, but do not perform compilation.

-w
Suppress all warning messages.

-w66
Suppress only FORTRAN 66 compatibility warnings.

-Wc, *arg1* [, *arg2*...]
Hand off any specified arguments to pass *c*, where *c* is one of the following:

0	compiler first pass
q	PCO pass
1	compiler second pass
a	assembler
l	link editor
f	iwcrt0 and mcrt0
m	listing merger
s	sort
c	iwcc
e	EFL
r	RATFOR

-Yc, *dir*
Get pass *c* from directory *dir*, where *c* is one of the following:

0	compiler first pass
q	PCO pass
1	compiler second pass
a	assembler
l	link editor
f	iwcrt0 and mcrt0
m	listing merger
s	sort

c	iwcc
e	EFL
r	RATFOR

Invoking the Assembler: iwas

```
iwas [option] ...  filename
```

Assembler Options

-#	Print the active command line options.
-##	Print the active command line options and a message listing permitted command line options.
-dl	Remove the line number entries from the symbol table of the object file.
-L	Create a listing file with the extension .l.
-m	Use the m4 preprocessor on the source file.
-o *filename*	Specify *filename* as the name of the object file, overriding the default name.
-O	List intermediate file information for debugging.
-rom	Prohibit data from straddling a 64-byte boundary.
-unlink	Remove the source file on completion of assembly.
-V	Write the version number of the assembler.
-VV	Write the version number of the assembler on standard error output and suppress assembling the source file.
-W	Suppress warning messages.
-XA	Enable the use of the alternate register names r0-r127 (general purpose registers) and cs000-cs3ff.
-XD	Dump the internal tables of machine-specific names.
-XN	Suppress printing of the date on listings.

Invoking the Linker: iwld

```
iwld [option] ...  filename ...
```

Linker Options

-e *entry*	Define *entry* as the entry point of the program.
-l*x*	This option is an abbreviation for the library name lib*x*.a, where *x* is a string.
-L*dir*	Add *dir* to the list of directories containing object library routines.
-m	Produce a simple load map, listing the names of the files which will be loaded plus their start addresses, input sections, output sections, and lengths.
-o *filename*	Specify *filename* as the name of the object file, overriding the default name a.out.
-r	Prevent final definitions from being given to common symbols, and suppress undefined symbol diagnostics.

-s	Removes the symbol table and relocation information to save space.
-u *name*	Enter *name* as an undefined symbol.
-x	Preserve only global symbols in the output symbol table.

Invoking the Disassembler: diwas

```
diwas [option] ...  filename
```

Disassembler Options

-#	Print version information and executes disassembly.
-##	Print version information, but does not execute disassembly.
-b	Produce byte mode addresses. This is the default.
-B	Suppress printing of byte mode addresses.
-c	Decode the given disassembly range as instructions rather than data.
-d	Dump the symbol table for debugging.
-D	Produce listing style output using obsolete double word addresses.
-f	Treat the input file as if the object file being disassembled had been assembled with the .flags pseudo-operation.
-H *path*	Define *path* as the path to the directory where the disassembler creates its symbol and hash table.
-i	Force the specified disassembly range to be initialized data rather than separate data and text sections.
-l	Invoke listing mode, producing hexadecimal location and object code output. This is the default.
-L *path*	Define *path* as the path to the standard symbol mnemonics (registers, condition codes).
-n	Create a new hash table for symbols.
-o *filename*	Specify *filename* as the name of the disassembly output file, overriding the default name.
-q	Suppress listing mode, causing the output to look like assembly language source code.
-r	Disable symbolization of addresses.
-R *start:end*	Define a range between address *start* and address *end* for disassembly.
-s	Enable symbolization of addresses. The disassembler uses what symbol table information is available to try to replace numeric addresses by labels in instructions. This is the default.
-t	Print elapsed user and system times for the phases of disassembly.
-u	Reuse the symbol hash table constructed at the last invocation of the disassembler.
-V	Write the version number of the assembler.
-VV	Write the version number of the assembler on standard error output and suppress assembling the source file.

-w	Produce listing style output using obsolete word addresses.
-W	Specify that the labels stored in the symbol table are given as word addresses.
-XA	Handle unknown condition codes and control space registers by using generic names.
-XD	Compile the standard symbol names from the text output produced by the assembler into a fast binary format.

Invoking the Profiler: iwprof

iwprof [*option*] ... *filename*

Profiler Options

-a	Sort by increasing symbol address.
-c	Sort by decreasing number of calls.
-h	Suppress the heading normally printed on the report.
-m *mdata*	Use the file *mdata* instead of mon0.out as the input profile file.
-n	Sort lexically by symbol name.
-o	Print each symbol address (in octal) along with the symbol.
-s	Print a summary of the monitoring parameters and statistics on the standard error output.
-t	Sort by decreasing percentage of total time.
-V	Print iwprof version information on the standard error output.
-x	Print each symbol address (in hexadecimal) along with the symbol name.
-z	Include all symbols in the profile range, even if associated with zero number of calls and zero time.

Invoking the Debugger: iwdb

iwdb [*option*] ...

iwdb Options

–l	Wait for lock: if another user has locked the array, wait until the array is unlocked.
–q	Execute quietly: do not print unnecessary status messages.
–b	Batch mode: quietly execute commands in initialization file *./.iwdbinit* and then exit.
–h	Print a help message, and then exit.
–nx	Don't execute commands in the initialization file ./.iwdbinit.
–H hostname	Start iwdbgrd on the iWarp host machine hostname, to which the iWarp array is attached. If this option is not used, the value of the environment variable IWARP_HOSTNAME is used. If IWARP_HOSTNAME is undefined, the name of the local host is used.
–C cfg_dir	Locate the configuration file in the directory cfg_dir. If this option is not used, the value of the environment variable IWARP_CFGPATH is used. If IWARP_CFGPATH is undefined, the path /iwarp/etc3.0/cfg is used.

–S n	Use iWarp array n on the named host. If this option is not used, the value of the environment variable IWARP_SIBNUM is used. If IWARP_SIBNUM is undefined, 0 is used.
–R root_dir	Locate the executable et3/iwdbgrd in the directory root_dir. If this option is not used, the value of the environment variable IWARP_ROOTPATH is used. If IWARP_ROOTPATH is undefined, /iwarp is used.
–x command_file	On startup, execute commands from the text file command_file.
–d directory	Add directory to the list of directories to search for source files in.
–w directory	Change directories to directory before executing.
–a	Attach mode. Assume that the array is locked, and programs are already running on one or more cells. Delay server initialization. After the first context command, kill programs running on cells in the context. Then, initialize the server. The symbol–file command can then be used to load the symbol table of the program. Debugging can then continue as if the program had been started under iwdb.

Invoking the Archiver: iwar

`iwar` *`key`* `[`*`posname`*`] [`*`option`*`] ...` *`afile`* `[`*`filename`*`] ...`

Archiver Keys

d	Delete the named files from the archive file.
m	Move the named files either to the end of the archive, or the position specified by a positioning argument and posname as in the case of the r key.
p	Print the named files.
q	Append the named files to the end of the archive file without searching the archive for duplicate names.
r	Replace the named files in the archive.
t	Print a table of contents of the archive file.
x	Extract the named files.

Archiver Options

-a	Place new files after *posname*.
-b	Place new files before *posname*.
-c	Create an archive.
-i	Identical to the b option.
-l	Place temporary files in the current working directory rather than in the default temporary directory, TMPDIR.
-s	Force the regeneration of the archive symbol table even if the archiver is not invoked with a command that will modify the archive contents.
-u	Replace those files that have changed since they were put in the archive.
-v	Use verbose mode.

Invoking the Symbol Table Lister: iwnm

```
iwnm [option] ...  filename ...
```

Symbol Table Lister Options

-e	Print only external and static symbols.
-f	Print redundant symbols (.text, .data, .lib, and .bss) normally suppressed.
-h	Do not display the output header data.
-n	Sort external symbols by name before they are printed.
-o	Print the value and size of a symbol in octal instead of decimal.
-p	Produce easily parsable, terse output.
-r	Prepend the name of the object file or archive to each output line.
-T	Truncate every name which would otherwise overflow its column and place an asterisk as the last character in the display name to mark it as truncated.
-u	Print only undefined symbols.
-v	Sort external symbols by value before they are printed.
-V	Print the version of the iwnm command executing on the standard error output.
-x	Print the value and size of a symbol in hexadecimal instead of decimal.

Invoking the Dumper: iwdump

```
iwdump [option] ...  filename ...
```

Dumper Options

-a	Dump the archive header of each member of each archive file specified.
-c	Dump the string table.
-f	Dump the header file.
-g	Dump the global symbols in the symbol table of an archive.
-h	Dump section headers.
-l	Dump line number information.
-L	Interpret and print the contents of the .lib sections.
-o	Dump each optional header.
-r	Dump relocation information.
-s	Dump sections contents.
-t	Dump symbol table entries.
-z *name*	Dump line number entries for the named function.

Dumper Option Modifiers

-d *number*	Dump the section number, number, or the range of sections starting at number and ending the number specified by +d.

+d *number*	Dump sections in the range either beginning with first section or beginning with section specified by -d.
-n *name*	Dump information pertaining only to the named entity. This modifier can be used with -h, -s, -r, -l, -t
-p	Suppress printing of the headers.
-t *index*	Dump only the indexed symbol table entry. The -t used in conjunction with +t, specifies a range of symbol table entries.
+t *index*	Dump the symbol table entries in the range ending with the indexed entry. The range begins at the first symbol table entry or at the entry specified by the -t modifier.
-u	Underline the name of the file for emphasis.
-v	Dump information in symbolic representation rather than numeric. This modifier can be used with all the above options except -s and -o options of iwdump.
-z *name, number*	Dump line number entry or range of line numbers at *number* for the function *name*.
-+z *number*	Dump line numbers starting at either function *name* or *number* specified by -z, up to *number* specified by +z.

Invoking the Sizer: iwsize

```
iwsize [option] ...  filename ...
```

Sizer Options

-f	Print the size of every loaded section, followed by the section name in parentheses.
-n	Include NOLOAD sections in the size.
-o	Print numbers in octal.
-V	Supply version information on the iwsize command.
-x	Print numbers in hexadecimal.

Executing a Program: lgo

```
lgo  [options] program
lgo  [options] -r program cell_list [% args]
[-r program cell_list [% args]] ...
```

lgo Options

-S *n*	Selects array number *n*.
-H *hostname*	Selects array host *hostname*.
-C *configpath*	Selects an alternate root pathname for the iWarp configuration files. The default is /iwarp/etc/cfg. If the environment variable IWARP_CFGPATH is defined, its value overrides the default, but doesn't override the command line option.

-ne	Skips the run phase; for use with -o.
-nl	Prevents locking and unlocking of the array. lgo assumes that the array is already locked and leaves the array locked after the run completes. When this option is used, lgo does not launch a Host Proxy Interface daemon.
-w *n*	Waits until *n* programs have reported exit status. By default, lgo waits for all programs to report exit status.
-s	This option is for backward compatibility.
-o *script*	Produces a script of the iwmon commands used to run the program and writes it to *script.*
-R rootpath	Selects an alternate root pathname for the iWarp software distribution. The default is /iwarp. If the environment variable IWARP_ROOTPATH is defined, its value overrides the default, but doesn't override this command line option.
-r *program cell_list* [% *args*]	Loads and runs a.out with arguments specified by argument array *args* on a cell specified by cellid or cells specified by a range of cellids. The first element of the argument array has to be a non-numeral. The lgo command line can have multiple -r switches.
–t timelimit	Set a timelimit on your lock. By default, after 10 minutes your lock on the array will expire and the next user who tries to get the array will break your lock. If timelimit is specified, the lock will expire after timelimit minutes. If timelimit is zero, the lock will never expire.
–m message	During an iwstat, a message will be displayed. The message will be displayed while you have the array locked. Note that message must appear as a single word on the command line; because of this you will usually want to put quotes around message and avoid using special shell characters such as !, ', etc.
-V	Prints command version information.
-VV	Prints command version information, and exits without

Locking an iWarp Array: iwlock

`iwlock [`*`option`*`] ...`

Options

-s	Check for locks by self (the uid of the user)
-w	Await an unlock.
–t timelimit	Set a timelimit on your lock. By default, after 10 minutes your lock on the array will expire and the next user who tries to get the array will break your lock. If timelimit is specified, the lock will expire after timelimit minutes. If timelimit is zero, the lock will never expire.
–m message	During an iwstat, a message will be displayed. The message will be displayed while you have the array locked. Note that message must appear as a single word on the command line; because of this you will usually want to put quotes around message and avoid using special shell characters such as !, ', etc.
-S *n*	Selects the array defined by SIB number *n*.
-H *hostname*	Selects array host *hostname*.
-R *rootpath*	Selects an alternate root pathname for the iWarp software distribution.
-C *configpath*	Selects root pathname for the iWarp configuration files.

Breaking a Lock on an iWarp Array: iwbrklock

`iwbrklock [`*`option`*`] ...`

Options

-S *n*	Selects the array defined by SIB number *n*.
-H *hostname*	Selects array host *hostname*.
-R *rootpath*	Selects an alternate root pathname for the iWarp software distribution.
-C *configpath*	Selects root pathname for the iWarp configuration files.
-V[V]	Print version information.

Unlocking an iWarp Array: iwunlock

```
iwunlock [option] ...
```

Options

-S *n*	Selects the array defined by SIB number *n*.
-H *hostname*	Selects array host *hostname*.
-R *rootpath*	Selects an alternate root pathname for the iWarp software distribution.
-C *configpath*	Selects root pathname for the iWarp configuration files.
-V[V]	Print version information.

Perform a Hard Reboot on an iWarp array: iwreset

```
iwreset [option] ...
```

Options

-S *n*	Selects the array defined by SIB number *n*.
-H *hostname*	Selects array host *hostname*.
-R *rootpath*	Selects an alternate root pathname for the iWarp software distribution.
-C *configpath*	Selects root pathname for the iWarp configuration files.
-V[V]	Print version information.

Load iWarp Executable Files on Cells: iwload

```
iwload [-ts] [-td] [-ds] [-dd] [-bs] [-bd] [-hs] [-hd] [-ss] [-sd]
[-S n] [-H hostname] [-V[V]] -r cells [-r cells] ... program [args]
```

Options

–ts	Place the text section in static RAM.
–td	Place the text section in dynamic RAM
–ds	Place the data section in static RAM.
–dd	Place the data section in dynamic RAM.
–bs	Place the bss section in static RAM.
–bd	Place the bss section in dynamic RAM.
–hs	Place the heap in static RAM.
–hd	Place the heap in dynamic RAM.
–ss	Place the stack in static RAM.
–sd	Place the stack in dynamic RAM.
-S *n*	Select the array defined by SIB number *n*.
-H *hostname*	Select array host *hostname*.
-R *rootpath*	Selects an alternate root pathname for the iWarp software distribution.
-C *configpath*	Selects root pathname for the iWarp configuration files.
-V[V]	Print version information.
-r cells	Specify a list of cells to receive the iWarp executable object file a.out.
program	Specify the iWarp executable object file.
args	Specify a list of arguments to be supplied to the cell program via the argv data structure.

Starting Cell Program Execution: iwrun

Syntax

```
iwrun [-S n] [-H hostname] [-R rootpath] [-C configpath] [-V[V]]
-r cells [-r cells] ...
```

Options

-S *n*	Selects the array defined by SIB number *n*.
-H *hostname*	Selects array host *hostname*. The default is the local host on which you logged in.
-R *rootpath*	Selects an alternate root pathname for the iWarp software distribution.
-C *configpath*	Selects root pathname for the iWarp configuration files.
-V[V]	Print version information.
-r cells	Specifies a list of cells to run.

Displaying status information: iwstat

Syntax

```
iwstat [option] ...
```

Options

-S *n*	Selects the array defined by SIB number *n*. The default SIB number is 0.
-H *hostname*	Selects array host *hostname*. The default is the local host on which you logged in.
-R *rootpath*	Selects an alternate root pathname for the iWarp software distribution.
-C *configpath*	Selects root pathname for the iWarp configuration files.
-su	Show 'user' status as a per-cell table of status values.
-se	Show exit status as a per-cell table of status values.
-e *N*	Cause iwstat's exit status to indicate whether or not N cells have reported exit status.
-V[V]	Print version information.

Clearing the Monitor's Cell Exit Status: iwclrstat

```
iwclrstat [option] ...
```

Options

-S *n*	Selects the array defined by SIB number *n*.
-H *hostname*	Selects array host *hostname*.
-R *rootpath*	Selects an alternate root pathname for the iWarp software distribution.
-C *configpath*	Selects root pathname for the iWarp configuration files.
-V[V]	Print version information.

Reporting Cell Program Execution Time: iwtime

```
iwclrstat [option] ...
```

Options

-S *n*	Selects the array defined by SIB number *n*.
-H *hostname*	Selects array host *hostname*.
-R *rootpath*	Selects an alternate root pathname for the iWarp software distribution.
-C *configpath*	Selects root pathname for the iWarp configuration files.
-V[V]	Print version information.

Displaying the Version of *the iWarp Monitor Daemon: iwvers*

```
iwvers [option] ...
```

Options

-S *n*	Selects the array defined by SIB number *n*.
-H *hostname*	Selects array host *hostname*.
-R *rootpath*	Selects an alternate root pathname for the iWarp software distribution.
-C *configpath*	Selects root pathname for the iWarp configuration files.
-V[V]	Print version information.

Terminating the Execution of a Cell Program: iwkill

`iwkill [`*option*`] ... ` *cell*

Options

-S *n*	Selects the array defined by SIB number *n*.
-H *hostname*	Selects array host *hostname*.
-R *rootpath*	Selects an alternate root pathname for the iWarp software distribution.
-C *configpath*	Selects root pathname for the iWarp configuration files.
-V[V]	Print version information.
cell	Specify a decimal cell number in the range 0 to array-size.

Starting the Host Proxy Interface (HPI) Daemon: iwhpi

`iwhpi [`*option*`] ...`

Options

-S *n*	Selects the array defined by SIB number *n*.
-H *hostname*	Selects array host *hostname*.
-R *rootpath*	Selects an alternate root pathname for the iWarp software distribution.
-C *configpath*	Selects root pathname for the iWarp configuration files.
-V[V]	Print version information.
-W *dir*	Select a working directory for the standard-IO files.
-I *stdin*	Select an alternate base name for the standard-in files.
-O *stdout*	Select an alternate basename for standard-out files.
-E *stderr*	Select an alternate basename for standard-error files.
-A	Append on output, rather than truncate.

Depositing Data to the Control Space Register on a Cell: iwdepcsr

`iwdepcsr [`*`option`*`] ... `*`cell csreg value`*

Options

-S *n*	Selects the array defined by SIB number *n*.
-H *hostname*	Selects array host *hostname*.
-V[V]	Print version information.
-R *rootpath*	Selects an alternate root pathname for the iWarp software distribution.
-C *configpath*	Selects root pathname for the iWarp configuration files.
cell	Specify a decimal cell number in the range 0 to array-size.
csreg	Specify a control space register name as defined by the assembler.
value	Specify a hexadecimal constant.

Depositing Data into Memory Locations on a Cell: iwdepmem

`iwdepmem [`*`option`*`] ... `*`cell address data-words`*

Options

-S *n*	Selects the array defined by SIB number *n*.
-H *hostname*	Selects array host *hostname*.
-R *rootpath*	Selects an alternate root pathname for the iWarp software distribution.
-C *configpath*	Selects root pathname for the iWarp configuration files.
-V[V]	Print version information.
cell	Specify a decimal cell id in the range 0 to the size of the array.
address	Specify a hexadecimal word-aligned byte address in cell's local memory.
datawords	Specify data in hexadecimal form.

Examining a Control Space Register on a Cell: iwexcsr

```
iwexcsr [option] ... cell csreg
```

Options

-S *n*	Selects the array defined by SIB number *n*.
-H *hostname*	Selects array host *hostname*.
-R *rootpath*	Selects an alternate root pathname for the iWarp software distribution.
-C *configpath*	Selects root pathname for the iWarp configuration files.
-V[V]	Print version information.
cell	Specify a decimal cell id in the range 0 to the size of the array.
csreg	Specify a control space register name as defined by the assembler, iwas(1).

Examine Memory Starting at a Specified Address: iwexmem

```
iwexmem [option] ... cell address bytecount
```

Options

-S *n*	Selects the array defined by SIB number *n*.
-H *hostname*	Selects array host *hostname*.
-R *rootpath*	Selects an alternate root pathname for the iWarp software distribution.
-C *configpath*	Selects root pathname for the iWarp configuration files.
-V[V]	Print version information.
cell	Specify a decimal cell id in the range 0 to the size of the array.
address	Specify a hexadecimal word-aligned byte address in cell's local memory.
bytecount	Specify a decimal count of words to examine.

Invoking the PCS Preprocessor: iwpp

```
iwpp [option] ... program
```

Preprocessor Options

-c *error_level*	Set the error level of the included code. Error level 1 includes all error checking. Error level 0 includes no error checking.
-D define	Run the C preprocessor with define option specified.
-d *debug*	Set the debug level for the preprocessor.
-I *dir*	Search the *dir* directory in addition to the standard directories for include files.
-V	Print the version number of iwpp and continue.
-VV	Print the version number and exit.

Invoking the PCS Array Combiner: iwcb

```
iwcb [option] ... array_program
```

Array Combiner Options

-d	Print debugging information and supress removing the executable array program and the array network description file.
-col *num*	Specify an array grid with *num* columns. By default the grid has 8 columns.
-row *num*	Specify an array grid with *num* rows. By default the grid has 8 rows.
-o *output*	Write the array module to a file specified by *output*. The default file name is *array.out*.
-w	Specify the array grid as a torus and use the back edges in routes. By default, an iWarp array does not use back edges.
–D*define*	Compile the array program with *define* defined for the preprocessing phase.
-I *dir*	Search the *dir* directory for files to be included in the array program, in addition to the standard include files.
-L *dir*	Search the *dir* directory for library files, in addition to the standard library directories .
-M *dir*	Search the *dir* directory for modules to be included in the array module, in addition to the current directory and the */iwarp/lib/modules* directory.
-V	Print the version number and continue.
-VV	Print the version number, print the sub-passes as they execute, and exit.
–Y *dir*	Get pass *c* from directory *dir*, where *c* is one of [A]indicating the location of the combiner library.

Invoking the PCS Array Loader: iwamld

```
iwamld array_module [option] ...
```

Array Loader Options

-c *error_level*	Link in the PCS library with the specified error level. Legal error levels are 0 and 1. 1 means all error checking. 0 means no error checking.
-d	Print additional debugging information.
-e	Execute the array module. This is the default.
-n	Do not execute the array module. Creates a script to run array module later.
-*row*[..*row*],*col*[..*col*],arg[,*arg*...]	Pass the specified arguments to the program on cell(row, col). Specify ranges using *row*..*row*,*co*l..*co*l for contiguous blocks only. For example, using 0..1,0..1 specifies giving the same arguments to the four cells in the upper left corner of the array.
-*row*,*col* [*args*]	Pass the specified arguments to the program on cell(row, col).
-H *host_name*	Execute the array module on the iWarp array attached to the host system with the name *host_name*. By default iwload tries to execute the array module on the current machine. This option overrides IWARP_HOSTNAME environment variable.
–S *sib_num*	Execute the array module on the iWarp system attached to the SIB, *sib_num*. By default, iwalmd tries to execute the array module attached to SIB 0. This option overrides the IWARP_SIBNUM environment variable.
–W*c*,*arg*[,*arg*...]	Hand off the argument(s) *arg* to pass *c* where *c* is one of [01] indicating iwcc compilation of init files or iwcc final link of the executable, respectively.
–l*x*	This option is an abbreviation for the library name libx.a, where *x* is a string. You can provide additional libraries for linking using this switch. At link time, the PCS array loader searches for the library in any directories provided with the –L switch (on the command line or using the –W1 switch). This is a shortcut alternative to using –W1.
–L*di*r	Add *dir* to the list of directories in which libraries are searched by iwld during the link phase. Directories specified with –L are searched before the standard directory /iwarp/lib. This is a shortcut alternative to using –W1.
–Y*c,dir*	Get pass *c* from directory *dir*, where *c* is one of [01ID] indicating the iwcc compilation of init files, iwcc final link of executables, PCS initialization library, or PCS data movement library, respectively.
-V	Print the version number and continue.
-VV	Print the version number, print the sub-passes as they execute, and exit.

Invoking the PCS Array Module Display: iwmd

```
iwmd array_module
```