

***** ICC COMPILER *****

-a[t|d|b|s)=(loc) - Place 'text', 'data', 'bss', or 'string' type data under (loc) location counter
-b=(directory) - Find current and subsequent C Compiler passes in (directory) location.
-c - Override default condition with respect to generation of position dependent/independent code.
-d - Override default condition with respect to generation of position dependent/independent data.
-g(c) - use alternate version of preprocessor pass for compilation (RC6809 and 0C6809 compilers only).
-i=(directory) - Search (directory) location for #include files.
-k - Display progress of compilation/assembly sequence on console.
-m(name)[=(string)] - Define (name) in preprocessor, with value (string) optionally assigned to (name).
-n - Inhibit execution of next compiler pass in the compilation sequence.
-r - Save C Compiler's intermediate assembly language output file.
-s - Disallow nested comments.
-s=(size) - Set maximum size of triple buffer.
-t=(directory) - Place C Compiler's temporary files in (directory) location.
-y=[=(n)] - Strip all identifiers to a maximum length of (n) characters.
-z Interpret "\n" (newline) characters as being carriage returns.

***** R09 RELOCATING ASSEMBLER *****

-a - Place all symbols except those beginning with a "?" character in the object file.
-c - Send Assembler's output listing to console.
-i - Include all included files in output listing.
-j - Include symbols beginning with a "?" character in the symbol table listing.
-l=(filename) - Place output listing in specified file.
-n - Do not produce an output listing.
-o=(filename) - Assign name to Assembler's relocatable output module.
-q=(class) - Assign numeric class identifier (0 through 255) to relocatable output module.
-s - Suppress listing of the symbol table.
-u - Force all undefined symbols to default to imported symbols.
-x - Don't generate an object module.
-z - Delete input file when Assembler has finished using it.

***** ILINK LINKER *****

-b - Do not search the default Standard Library.
-c=(file) - Get additional link-time parameters from command file.
-d[(c)] - Call optional cross-loader named "(c)LD" when Linker finishes.
-e=(symbol) - Set entry point.
-f(string) - Search additional Standard Library named "lib(string).R"
-l[s][x][u][=(file)] - Produce a linker output listing.
-m=(symbol) - Define primary function naming symbol.
-n - Inhibit Linker from automatically calling Loader.
-o=(file) - Assign name to output file.
-p[(c)] - Pipe Linker's output to loader.
-r - Save Linker's output file (during automatic link-and-load operations).
-s - Strip output file of all non-entry-defined symbols.
-t=(classlist) - Use (classlist) classes of modules during linking process, if they are available.

***** HLD LOADER *****

-a=(seg);(placernent)[,(seg);(placement)] - Set segment memory bound (segment may begin, or end, at a specific memory location, or specified to immediately follow, or immediately precede, another segment).
-c=(file) - Get additional parameters from command file.
-g=(type) - Set output format (Motorola S Record, Intel Hex, Tek Hex, or Extendend Tek hex format).
-h - Define EOL character to be carriage return (rather than newline character).
-l[s][=(file)] - Produce a Loader output listing.
-o=(name) - Assign name to output file.
-u=(seg) - Place uninitialized data in specified segment.
-v[(char)] - Modify Loader's symbol changing procedures for symbols beginning with non-alpha characters.
-w - Produce executable output file no matter what.
-x (type):(ext) - Set output filename extension for specified type of hex output format.
-z - Delete loader's input file when Loader has finished using it.

***** ULD LOADER *****

-a=(sec):(seg)[,(seg)] - Assign location counter segment to UniFlex program section (text, data. or bss).
-c=(file) - Get additional parameters from command file.
-l[s] [= (file)] - Generate loader output listing.
-o=(name) - Assign name to output file.
-v=(size) - Set stack section size.
-w - Produce an executable output file no matter what.
-x [= (pagesize)] - Produce output file in UniFlex segmented format.
-y=(origin) - Set text section origin at specified location.
-z - Delete Loader's input file when Loader has finished using it.

***** FLD LOADER *****

-a=(sec):(Seg)[,(seg)] - Assign a location counter segment to a Flex program section (text, data, or bss).
-c=(file) - Get additional parameters from command file.
-l[S] [= (file)] - Produce an output listing.
-o=(name) - Assign name to output file.
-w - Generate executable output file no matter what.
-y[t|d|b] =(origin) - Set origin for text, data, or uninitialized section of output file.
-z - Delete input file after Loader has finished using it.

***** OLD LOADER *****

-a=(sec):(seg)[,(seg)] - Assign a location counter segment to an OS9 program section (text, data, or bss)
-c=(file) - Get additional parameters from command file.
-l[s] [= (file)] - Produce an output listing.
-o=(name) - Assign name to output file.
-V=(size) - Set stack section size.
-w - Generate an executable output file no matter what.
-x - Place executable program module and data initialization information module in separate files.
-z - Delete the input file after the Loader has finished using it.

***** LIBMAN LIBRARY MANAGER COMMANDS *****

a (file),(module)[,(class)] - Add module to library; create new library.
d (module)[,(class)] - Delete module from library.
r (file),(module)[,(class)] - Replace module in library.
q - Quit Library Manager 7after saving library file being edited).
omit - Exit Library Manager (without saving edited file).
l (module)[,(class)] - List information on named file.
sl (module)[,(class)] - List abbreviated information on named file.
h - Provide on-line help.
lo (file) - Explicitly load a library file.
ll (file) - List a loaded library.
sll (file) - Provide abbreviated listing of a loaded library.
s (file) - Save library using the filename indicated by (file).
c (file) - Get additional commands from named command file.
e (strings) - Echo specified strings to the terminal.
f (module)[,(class)] - Find named module.
p (module)[,(class)] - Print information for named module.
sp (module)[,(class)] - Print abbreviated listing of information for named module.
i (file),(module)[,(Class)] - Insert named module in library so it precedes current module.

.....