Chipsmall
深圳市津利帝科技有限公司
Geelead Technology Limited
Chipsmall
采购部

PCD C Compiler Reference Manual

June 2012

Table of Contents

Overview 1
PCD 1
PCW Overview 1
Technical Support 10
Directories 10
File Formats 10
Invoking the Command Line Compiler 11
Program Syntax 15
Overall Structure 15
Comment 15
Trigraph Sequences 16
Multiple Project Files 17
Multiple Compilation Units 17
Example 28
STATEMENTS 31
Statements 31
if 32
while 32
do 33
do-while 33
for 33
switch 34
return 34
goto 35
label 35
break 35
continue 36
expr 36
; 36
stmt 36
Expressions 37
Expressions 37
Operators 38
Operator Precedence 39
Reference Parameters 40
Variable Argument Lists 40
Default Parameters 41
Overloaded Functions 41
DATA DEFINITIONS 43
Basic and Special types 43
Declarations 46
Non-RAM Data Definitions 46
Using Program Memory for Data 48
Function Definition 49
Functional Overviews 51
I2C 51

ADC 52
Analog Comparator 53
CAN Bus 54
Configuration Memory 58
CRC 58
DAC 59
Data Eeprom 60
DCI 61
DMA 62
Data Signal Modulator 63
Extended RAM 64
General Purpose I/O 65
Input Capture 66
Internal Oscillator 67
Interrupts 68
Linker 69
Output Compare/PWM Overview 69
Motor Control PWM 70
PMP/EPMP 71
Program Eeprom 72
QEI 75
RS232 I/O 75
RTCC 77
RTOS 78
SPI 80
TimerA 81
TimerB 82
Timers 83
Voltage Reference 84
WDT or Watch Dog Timer 85
PRE-PROCESSOR 87
#ASM #ENDASM 87
#BANK_DMA 99
#BANKX 100
#BANKY 100
#BIT 101
#BUILD 101
#BYTE 103
#CASE 103
DATE 104
#DEFINE 104
#DEFINEDINC 105
#DEVICE 106
DEVICE 108
#ERROR 109
#EXPORT (options) 109
__FILE__ 111
__FILENAME__ 111
#FILL_ROM 111

#FUSES 112
#HEXCOMMENT 113
#ID 113
#IF expr #ELSE #ELIF #ENDIF 114
#IFDEF #IFNDEF #ELSE #ELIF #ENDIF 115
#IGNORE_WARNINGS 116
#IMPORT (options) 116
#INCLUDE 117
#INLINE 118
#INT_xxxx 118
#INT_DEFAULT 120
__LINE__ 120
#LIST 120
#LINE 121
#LOCATE 121
#MODULE 122
#NOLIST 122
#OCS 123
#OPT 123
#ORG 123
_ _PCD_ _ 125
#PRAGMA 125
#RECURSIVE 126
#RESERVE 126
#ROM 127
#SEPARATE 128
#SERIALIZE 129
#TASK 130
__ TIME __ 131
#TYPE 131
#UNDEF 133
#USE DELAY 133
#USE DYNAMIC_MEMORY 134
#USE FAST_IO 134
#USE FIXED_IO 135
#USE I2C 135
#USE RS232 137
#USE RTOS 140
#USE SPI 141
#USE STANDARD_IO 142
#USE TIMER 143
#USE TOUCHPAD 144
#WARNING 145
#WORD 145
#ZERO_RAM 146
BUILT-IN FUNCTIONS Directory 147
BUILT-IN-FUNCTIONS 147
abs( ) 151
adc_done( ) adc_done2( ) 152

assert( ) 153
atoe( ) 153
atof( ) atof48( ) atof64( ) 154
atoi( ) atol( ) atoi32( ) atoi48( ) atoi64( ) 155
bit_clear( ) 155
bit_first( ) 156
bit_last( ) 156
bit_set( ) 157
bit_test( ) 158
bsearch( ) 158
calloc( ) 159
ceil( ) 160
clear_interrupt( ) 160
crc_calc( ) crc_calc8( ) crc_calc16( ) crc_calc32( ) 161
crc_init(mode) 162
dac_write( ) 162
dci_data_received( ) 163
dci_read( ) 164
dci_start( ) 165
dci_transmit_ready( ) 165
dci_write( ) 166
delay_cycles( ) 167
delay_ms( ) 167
delay_us( ) 168
disable_interrupts( ) 169
div( ) ldiv( ) 170
dma_start( ) 170
dma_status( ) 171
enable_interrupts( ) 172
erase_program_memory 173
exp( ) 173
ext_int_edge( ) 174
fabs( ) 174
floor( ) 175
fmod( ) 175
free( ) 176
frexp( ) 176
get_capture( ) 177
get_motor_pwm_count( ) 178
get_ticks( ) 178
get_timerA( ) 179
get_timerB( ) 179
get_timerx( ) 180
get_timerxy( ) 180
get_tris_x( ) 181
getc( ) getch( ) getchar( ) fgetc( ) 181
getenv( ) 182
gets( ) fgets( ) 185
goto_address( ) 186

high_speed_adc_done( ) 186
i2c_isr_state( ) 187
i2c_poll( ) 188
i2c_read( ) 188
i2c_slaveaddr( ) 189
i2c_speed( ) 190
i2c_start( ) 190
i2c_stop( ) 191
i2c_write( ) 191
input( ) 192
input_change_x( ) 193
input_state( ) 193
input_x( ) 194
interrupt_active( ) 194
isalnum(char) isalpha(char) isdigit(char) islower(char) isspace(char) isupper(char) isxdigit(char) iscntrl(x) isgraph(x) isprint(x) ispunct(x) 195
isamong( ) 196
itoa( ) 197
kbhit( ) 197
label_address( ) 198
labs( ) 199
ldexp( ) 199
log( ) 200
log10( ) 200
longjmp( ) 201
make8( ) 201
make16( ) 202
make32( ) 203
malloc( ) 203
memcpy( ) memmove( ) 204
memset( ) 205
modf( ) 205
_mul( ) 206
nargs( ) 206
offsetof( ) offsetofbit( ) 207
output_x( ) 208
output_bit( ) 209
output_drive( ) 210
output_float( ) 210
output_high( ) 211
output_low( ) 211
output_toggle( ) 212
perror( ) 212
pmp_address(address) 212
pmp_output_full( ) pmp_input_full( ) pmp_overflow( ) pmp_error( ) pmp_timeout( ) 213
pmp_read( ) 214
port_x_pullups ( ) 216
pow( ) pwr( ) 216
printf( ) fprintf( ) 217

psp_output_full( ) psp_input_full( ) psp_overflow( ) psp_error( ) psp_timeout( ) 219
psp_read( ) 219
psp_write( ) 220
putc( ) putchar( ) fputc( ) 221
puts( ) fputs( ) 221
qei_get_count( ) 222
qei_set_count( ) 222
qei_status( ) 223
qsort( ) 223
rand( ) 224
read_adc( ) read_adc2( ) 224
read_configuration_memory( ) 225
read_eeprom( ) 226
read_extended_ram( ) 226
read_high_speed_adc( ) 227
read_program_memory( ) 228
read_rom_memory( ) 229
realloc( ) 229
reset_cpu( ) 230
restart_cause( ) 230
restart_wdt( ) 231
rotate_left( ) 232
rotate_right( ) 233
rtc_alarm_read( ) 233
rtc_alarm_write( ) 234
rtc_read( ) 234
rtc_write( ) 235
rtos_await( ) 235
rtos_disable( ) 235
rtos_enable( ) 236
rtos_msg_poll( ) 236
rtos_msg_read( ) 237
rtos_msg_send( ) 237
rtos_overrun( ) 238
rtos_run( ) 238
rtos_signal( ) 238
rtos_stats( ) 239
rtos_terminate( ) 239
rtos_wait( ) 240
rtos_yield( ) 240
set_adc_channel( ) set_adc_channel2( ) 241
set_compare_time( ) 241
set_motor_pwm_duty( ) 242
set_motor_pwm_event( ) 242
set_motor_unit( ) 243
set_pullup( ) 243
set_ticks( ) 244
set_timerA( ) 245
set_timerB( ) 245

set_timerx( ) 246
set_timerxy( ) 246
set_tris_x( ) 247
set_uart_speed( ) 247
setjmp( ) 248
setup_adc(mode) setup_adc2(mode).......... 249
setup_adc_ports( ) setup_adc_ports2( ).......... 249
setup_capture( ) 250
setup_comparator( ) 251
setup_compare( ) 251
setup_crc(mode) 252
setup_dac( ) 253
setup_dci( ) 253
setup_dma( ) 255
setup_high_speed_adc( ).......... 255
setup_high_speed_adc_pair( ) 256
setup_low_volt_detect( ) 257
setup_motor_pwm( ) 257
setup_oscillator( ) 258
setup_pmp(option,address_mask) 259
setup_power_pwm_pins( ) 260
setup_psp(option,address_mask) 260
setup_qei( ) 261
setup_rtc( ) 262
setup_rtc_alarm( ) 262
setup_spi( ) setup_spi2( ) 263
setup_timerx( ) 264
setup_timer_A( ) 265
setup_timer_B( ) 266
setup_uart( ) 266
setup_vref( ) 267
setup_wdt( ) 268
shift_left( ) 268
shift_right( ) 269
sin( ) cos( ) tan( ) asin( ) acos() atan() sinh() cosh() tanh() atan2() 270
sleep( ) 271
spi_data_is_in( ) spi_data_is_in2( ) 272
spi_read( ) spi_read2( ) 272
spi_write( ) spi_write2( ) 273
spi_xfer( ) 273
sprintf( ) 274
sqrt( ) 274
srand( ) 275
STANDARD STRING FUNCTIONS( ) memchr( ) memcmp( ) strcat( ) strchr( )
strcmp( ) strcoll( ) strcspn( ) strerror( ) stricmp( ) strlen( ) strlwr( ) strncat( ) strncmp(
) strncpy( ) strpbrk( ) strrchr( ) strspn( ) strstr( ) strxfrm( ) 275
strcat( ) 277
strchr( ) 277
strcmp( ) 277

strcoll( ) 277
strcpy( ) strcopy( ) 278
STRCSPN( ) 278
strlen( ) 278
strlwr( ) 278
strncat( ) 279
strncmp( ) 279
strncpy( ) 279
strpbrk( ) 279
strrchr( ) 279
strcspn( ) 279
strstr( ) 279
strtod( ) strtof( ) strtof48( ) 279
strtok( ) 280
strtol( ) 281
strtoul( ) 282
strxfrm( ) 283
swap( ) 283
tolower( ) toupper( ) 283
touchpad_getc( ) 284
touchpad_hit( ) 285
touchpad_state( ) 285
va_arg( ) 286
va_end( ) 287
va_start 287
write_configuration_memory( ) 288
write_eeprom( ) 289
write_extended_ram( ) 289
write_program_memory( ) 290
Standard C Include Files 291
errno.h 291
float.h 291
limits.h 292
locale.h 293
setjmp.h 293
stddef.h 293
stdio.h 293
stdlib.h 293
Error Messages 295
Compiler Error Messages 295
Compiler Warning Messages 305
Compiler Warning Messages 305
Common Questions and Answers 309
How are type conversions handled? 309
How can a constant data table be placed in ROM? 310
How can I use two or more RS-232 ports on one PIC®? 311
How do I directly read/write to internal registers? 312
How do I do a printf to a string? 312
How do I get getc() to timeout after a specified time? 313

How do I make a pointer to a function? 314
How do I wait only a specified time for a button press? 314
How do I write variables to EEPROM that are not a word? 315
How does one map a variable to an I/O port? 315
How does the compiler determine TRUE and FALSE on expressions? 316
How does the PIC® connect to a PC? 317
How does the PIC® connect to an I2C device? 318
How much time do math operations take? 319
What are the various Fuse options for the dsPIC/PIC 24 chips? 319
What can be done about an OUT OF RAM error? 321
What is an easy way for two or more PICs® to communicate? 322
What is an easy way for two or more PICs® to communicate? 323
What is the format of floating point numbers? 323
Why does the .LST file look out of order? 325
Why is the RS-232 not working right? 326
Example Programs 329
EXAMPLE PROGRAMS 329
Software License Agreement 341
SOFTWARE LICENSE AGREEMENT 341
Index 343

OVERVIEW

C Compiler

PCD

PCD is a C Compiler for Microchip's 24bit opcode family of microcontrollers, which include the dsPIC30, dsPIC33 and PIC24 families. The compiler is specifically designed to meet the unique needs of the dsPIC® microcontroller. This allows developers to quickly design applications software in a more readable, high-level language.

The compiler can efficiently implement normal C constructs, input/output operations, and bit twiddling operations. All normal C data types are supported along with special built in functions to perform common functions in the MPU with ease.

Extended constructs like bit arrays, multiple address space handling and effective implementation of constant data in Rom make code generation very effective.

PCW Overview

Beginning in version 4.XXX of PCW, the menus and toolbars are set-up in specially organized Ribbons. Each Ribbon relates to a specific type of activity an is only shown when selected. CCS has included a "User Toolbar" Ribbon that allows the user to customize the Ribbon for individual needs.

File Menu

Click on this icon for the following items:

New Creates a new File

Open Opens a file to the editor. Includes options for Source, Project, Output, RTF, Flow Chart, Hex or Text. Ctrl+O is the shortcut.

Close Closes the file currently open for editing. Note, that while a file is open in PCW for editing, no other program may access the file. Shift+F11 is the shortcut.

Close All Closes all files open in the PCW.

Save Saves the file currently selected for editing. Crtl+S is the shortcut.

Save Prompts for a file name to save the currently selected file.

As
Save All   All open files are saved.

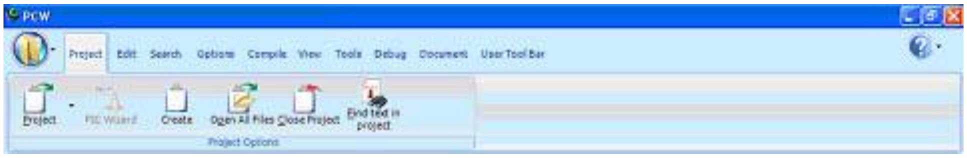
Encrypt   Creates an encrypted include file. The standard compiler #include directive will accept files with this extension and decrypt them when read. This allows include files to be distributed without releasing the source code.

Print   Prints the currently selected file.

Recent Files   The right-side of the menu has a Recent Files list for commonly used files.

Exit   The bottom of the menu has an icon to terminate PCW.

Project Menu Ribbon



Project   Open an existing project (.PJT) file as specified and the main source file is loaded.

PIC Wizard   This command is a fast way to start a new project. It will bring up a screen with fill-in-the-blanks to create a new project. When items such as RS232 I/O, i2C, timers, interrupts, A/D options, drivers and pin name are specified by the user, the Wizard will select required pins and pins that may have combined use. After all selections are made, the initial .c and .h files are created with #defines, #includes and initialization commands required for the project.

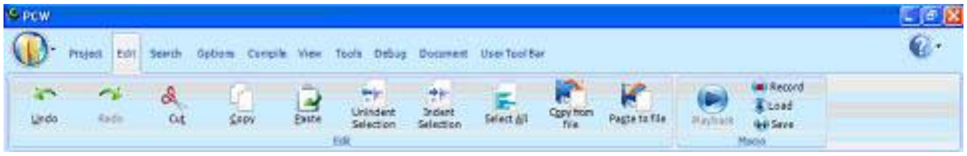
Create   Create a new project with the ability to add/remove source files, include files, global defines and specify output files.

Open All Files   Open all files in a project so that all include files become known for compilation.

Close Project   Close all files associated with project.

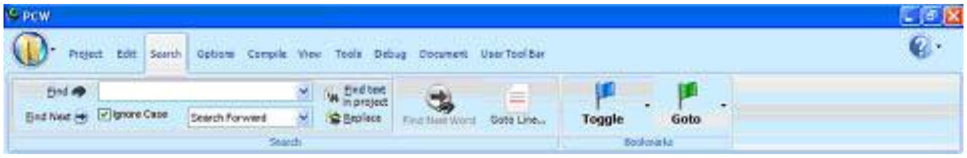
Find Text in Project   Ability to search all files for specific text string.

Edit Menu Ribbon



Undo	Undoes the last deletion
Redo	Re-does the last undo
Cut	Moves the selected text from the file to the clipboard.
Copy	Copies the selected text to the clipboard.
Paste	Applies the clipboard contents to the cursor location.
Unindent Selection	Selected area of code will not be indented.
Indent Selection	Selected area of code will be properly indented.
Select All	Highlighting of all text.
Copy from File	Copies the contents of a file to the cursor location.
Past to File	Applies the selected text to a file.
Macros	Macros for recording, saving and loading keystrokes and mouse-strokes.

Search Menu Ribbon



Find	Locate text in file.
Find Text in Project	Searches all files in project for specific text string.

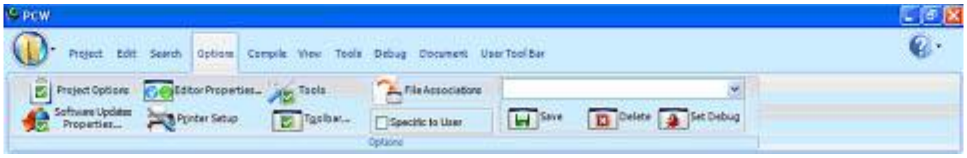
Find Next Word at Cursor
Locates the next occurrence of the text selected in the file.

Goto Line
Cursor will move to the user specified line number.

Toggle Bookmark
Set/Remove bookmark (0-9) at the cursor location.

Goto Bookmark
Move cursor to the specified bookmark (0-9).

Options Menu Ribbon



Project Options
Add/remove files, include files, global defines and output files.

Editor Properties
Allows user to define the set-up of editor properties for Windows options.

Tools
Window display of User Defined Tools and options to add and apply.

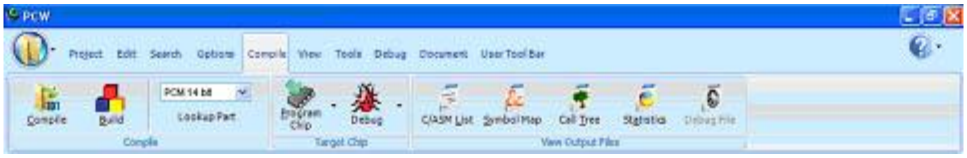
Software Updates Properties
Ability for user to select which software to update, frequency to remind Properties user and where to archive files.

Printer Setup
Set the printer port and paper and other properties for printing.

Toolbar Setup
Customize the toolbar properties to add/remove icons and keyboard commands.

File Associations
Customize the settings for files according to software being used.

Compile Menu Ribbon



Compile
Compiles the current project in status bar using the current compiler.

Build
Compiles one or more files within a project.

Compiler
Pull-down menu to choose the compiler needed.

Lookup Part
Choose a device and the compiler needed will automatically be selected.

Program Chip
Lists the options of CCS ICD or Mach X programmers and will connect to SIOW program.

Debug
Allows for input of .hex and will output .asm for debugging.

C/ASM List
Opens listing file in read-only mode. Will show each C source line code and the associated assembly code generated.

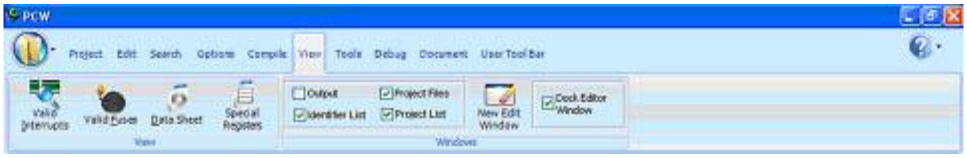
Symbol Map
Opens the symbol file in read-only mode. Symbol map shows each register location and what program variable are saved in each location.

Call Tree
Opens the tree file in read-only mode. The call tree shows each function and what functions it calls along with the ROM and RAM usage for each.

Statistics
Opens the statistics file in read-only mode. The statistics file shows each function, the ROM and RAM usage by file, segment and name.

Debug File
Opens the debug file in read-only mode. The listing file shows each C source line code and the associated assembly code generated.

View Menu Ribbon



Valid Interrupts
This displays a list of valid interrupts used with the #INT_keyword for the chip used in the current project. The interrupts for other chips can be viewed using the

drop down menu.

Valid Fuses
This displays a list of valid FUSE used with the #FUSES directive associated with the chip used in the current project. The fuses for other chips can be viewed using the drop down menu.

Data Sheets
This tool is used to view the Manufacturer data sheets for all the Microchip parts supported by the compiler.

Part Errata
This allows user to view the errata database to see what errata is associated with a part and if the compiler has compensated for the problem.

Special Registers
This displays the special function registers associated with the part.

New Edit Window
This will open a new edit window which can be tiled to view files side by side.

Dock Editor Window
Selecting this checkbox will dock the editor window into the IDE.

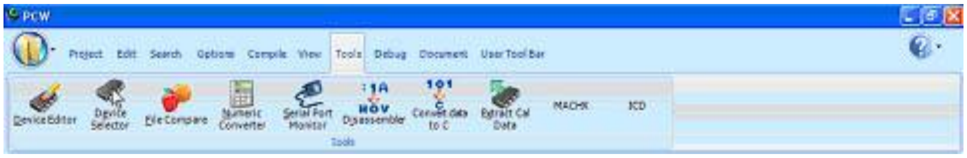
Project Files
When this checkbox is selected, the Project files slide out tab is displayed. This will allow quicker access to all the project source files and output files.

Project List
Selecting this checkbox displays the Project slide out tab. The Project slide out tab displays all the recent project files.

Output
Selecting this checkbox will enable the display of warning and error messages generated by the compiler.

Identifier List
Selecting this checkbox displays the Identifier slide out tab. It allows quick access to project identifiers like functions, types, variables and defines.

Tools Menu Ribbon



Device Editor
This tool is used to edit the device database used by the compiler to control compilations. The user can edit the chip memory, interrupts, fuses and other peripheral settings for all the supported devices.

Device Selector — This tool uses the device database to allow for parametric selection of devices. The tool displays all eligible devices based on the selection criteria.

File Compare — This utility is used to compare two files. Source or text files can be compared line by line and list files can be compared by ignoring the RAM/ROM addresses to make the comparisons more meaningful.

Numeric Converter — This utility can be used to convert data between different formats. The user can simultaneously view data in various formats like binary, hex, IEEE, signed and unsigned.

Serial Port Monitor — This tool is an easy way of connecting a PIC to a serial port. Data can be viewed in ASCII or hex format. An entire hex file can be transmitted to the PIC which is useful for bootloading application.

Disassembler — This tool will take an input hex file and output an ASM.

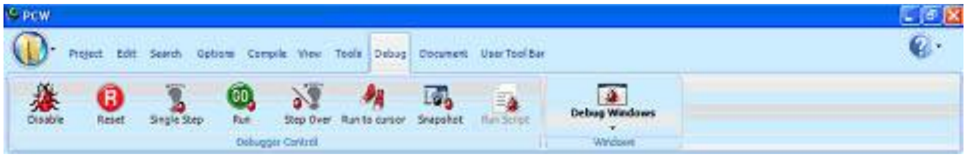
Convert Data to C — This utility will input data from a text file and generate code is form of a #ROM or CONST statement.

Extract Calibration — This tool will input a hex file and extract the calibration data to a C include file. This feature is useful for saving calibration data stored at top of program memory from certain PIC chips.

MACH X — This will call the Mach-X.exe program and will download the hex file for the current project onto the chip.

ICD — This will call the ICD.exe program and will download the hex file for the current project onto the chip.

Debug Menu Ribbon



Enable Debugger — Enables the debugger. Opens the debugger window, downloads the code and on-chip debugger and resets the target into the debugger.

Reset — This will reset the target into the debugger.

Single Step — Executes one source code line at a time. A single line of C source code or ASM code is executed depending on whether the source code or the list file tab in the editor is active.

Step Over This steps over the target code. It is useful for stepping over function calls.

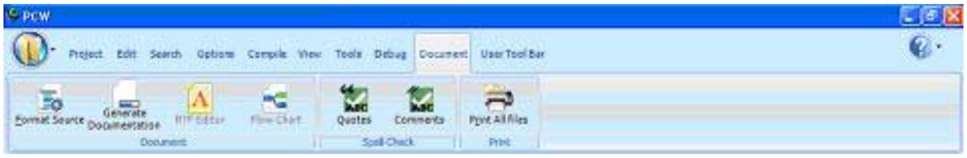
Run to Cursor Runs the target code to the cursor. Place the cursor at the desired location in the code and click on this button to execute the code till that address.

Snapshot This allows users to record various debugging information. Debug information like watches, ram values, data eeprom values, rom values , peripheral status can be conveniently logged. This log can be saved, printed, overwritten or appended.

Run Script This tool allows the IDE's integrated debugger to execute a C-style script. The functions and variable of the program can be accesses and the debugger creates a report of the results.

Debug Windows This drop down menu allows viewing of a particular debug tab. Click on the tab name in the drop down list which you want to view and it will bring up that tab in the debugger window.

Document Menu Ribbon



Format Source This utility formats the source file for indenting, color syntax highlighting, and other formatting options.

Generate Document This will call the document generator program which uses a user generated template in .RTF format to merge with comment from the source code to produce an output file in .RTF format as source code documentation.

RTF Editor Open the RTF editor program which is a fully featured RTF editor to make integration of documentation into your project easier.

Flow Chart Opens a flow chart program for quick and easy charting. This tool can be used to generate simple graphics including schematics.

Quotes Performs a spell check on all the words within quotes.

Comments Performs a spell check on all the comments in your source code.

Print all Files Print all the files of the current project.

Help Menu

Click on this icon for the following items:

Contents	Help File table of contents
Index	Help File index
Keyword at Cursor	Index search in Help File for the keyword at the cursor location. Press F1 to use this feature.
Debugger Help	Help File specific to debugger functionality.
Editor	Lists the Editor Keys available for use in PCW. Shft+F12 will also call this function help file page for quick review.
Data Types	Specific Help File page for basic data types.
Operators	Specific Help File page for table of operators that may be used in PCW.
Statements	Specific Help File page for table of commonly used statements.
Preprocessor Commands	Specific Help File page for listing of commonly used preprocessor commands.
Built-in Functions	Specific Help File page for listing of commonly used built-in functions provided by the compiler.
Technical Support	Technical Support wizard to directly contact Technical Support via email and the ability to attach files.
Check for Software Updates	Automatically invokes Download Manager to view local and current versions of software.
Internet	Direct links to specific CCS website pages for additional information.
About	Shows the version of compiler(s) and IDE installed.

Technical Support

Compiler, software, and driver updates are available to download at:
http://www.ccsinfo.com/download

Compilers come with 30 or 60 days of download rights with the initial purchase. One year maintenance plans may be purchased for access to updates as released.

The intent of new releases is to provide up-to-date support with greater ease of use and minimal, if any, transition difficulty.

To ensure any problem that may occur is corrected quickly and diligently, it is recommended to send an email to "x-text-underline: normal; support@ccsinfo.com or use the Technical Support Wizard in PCW. Include the version of the compiler, an outline of the problem and attach any files with the email request. CCS strives to answer technical support timely and thoroughly.

Technical Support is available by phone during business hours for urgent needs or if email responses are not adequate. Please call 262-522-6500 x32.

Directories

The compiler will search the following directories for Include files.

- Directories listed on the command line
- Directories specified in the .PJT file
- The same directory as the source file

By default, the compiler files are put in C:\Program Files\PICC and the example programs and all Include files are in C:\Program Files\PICC\EXAMPLES.

The compiler itself is a DLL file. The DLL files are in a DLL directory by default in C:\Program Files\PICC\DLL. Old compiler versions may be kept by renaming this directory.

Compiler Version 4 and above can tolerate two compilers of different versions in the same directory. Install an older version (4.xx) and rename the devices4.dat file to devices4X.dat where X is B for PCB, M is for PCM, and H is for PCH. Install the newer compiler and do the same rename of the devices4.dat file.

File Formats

.C	This is the source file containing user C source code.
.H	These are standard or custom header files used to define pins, register, register bits, functions and preprocessor directives.
.PJT	This is the project file which contains information related to the project.

.LST	This is the listing file which shows each C source line and the associated assembly code generated for that line. The elements in the .LST file may be selected in PCW under Options>Project Options>File Formats Match code -Includes the HEX opcode for each instruction SFR names -Instead of an address a name is used. For example instead of 044 is will show CORCON Symbols -Shows variable names instead of addresses Interpret -Adds a pseudo code interpretation to the right of assembly instruction to help understand the operation. For example: `LSR W4,#8,W5   :   W5=W4>>8`
.SYM	This is the symbol map which shows each register location and what program variables are stored in each location.
.STA	The statistics file shows the RAM, ROM, and STACK usage. It provides information on the source codes structural and textual complexities using Halstead and McCabe metrics.
.TRE	The tree file shows the call tree. It details each function and what functions it calls along with the ROM and RAM usage for each function.
.HEX	The compiler generates standard HEX files that are compatible with all programmers.
.COF	This is a binary containing machine code and debugging information.
.COD	This is a binary file containing debug information.
.RTF	The output of the Documentation Generator is exported in a Rich Text File format which can be viewed using the RTF editor or wordpad.
.RVF	The Rich View Format is used by the RTF Editor within the IDE to view the Rich Text File.
.DGR	The .DGR file is the output of the flowchart maker.
.ESYM	This file is generated for the IDE users. The file contains Identifiers and Comment information. This data can be used for automatic documentation generation and for the IDE helpers.
.OSYM	This file is generated when the compiler is set to export a relocatable object file. This file is a .sym file for just the one unit.

Invoking the Command Line Compiler

The command line compiler is invoked with the following command:

```
     CCSC    [options]    [cfilename]
```

Valid options:

+FB	Select PCB (12 bit)	-D	Do not create debug file
+FM	Select PCM (14 bit)	+DS	Standard .COD format debug file

+FH	Select PCH (PIC18XXX)	+DM	.MAP format debug file
+Yx	Optimization level x (0-9)	+DC	Expanded .COD format debug file
+FS	Select SXC (SX)	+EO	Old error file format
+ES	Standard error file	-T	Do not generate a tree file
+T	Create call tree (.TRE)	-A	Do not create stats file (.STA)
+A	Create stats file (.STA)	-EW	Suppress warnings (use with +EA)
+EW	Show warning messages	-E	Only show first error
+EA	Show all error messages and all warnings	+EX	Error/warning message format uses GCC's "brief format" (compatible with GCC editor environments)
+FD	Select PCD (dsPIC30/dsPIC33/PIC24)	+DF	Enables the output of an OFF debug file.

The xxx in the following are optional. If included it sets the file extension:

+LNxxx	Normal list file	+O8xxx	8-bit Intel HEX output file
+LSxxx	MPASM format list file	+OWxxx	16-bit Intel HEX output file
+LOxxx	Old MPASM list file	+OBxxx	Binary output file
+LYxxx	Symbolic list file	-O	Do not create object file
-L	Do not create list file		

+P	Keep compile status window up after compile
+Pxx	Keep status window up for xx seconds after compile
+PN	Keep status window up only if there are no errors
+PE	Keep status window up only if there are errors

+Z	Keep scratch files on disk after compile
+DF	COFF Debug file
I+="..."	Same as I="..." Except the path list is appended to the current list
I="..."	Set include directory search path, for example: I="c:\picc\examples;c:\picc\myincludes" If no I= appears on the command line the .PJT file will be used to supply the include file paths.

-P	Close compile window after compile is complete
+M	Generate a symbol file (.SYM)
-M	Do not create symbol file
+J	Create a project file (.PJT)
-J	Do not create PJT file
+ICD	Compile for use with an ICD
#xxx="yyy"	Set a global #define for id xxx with a value of yyy, example: #debug="true"
+Gxxx="yyy"	Same as #xxx="yyy"
+?	Brings up a help file

-? Same as +?

+STDOUT Outputs errors to STDOUT (for use with third party editors)
+SETUP Install CCSC into MPLAB (no compile is done)
sourceline= Allows a source line to be injected at the start of the source file.
Example: CCSC +FM myfile.c sourceline="#include <16F887.h>"
+V Show compiler version (no compile is done)
+Q Show all valid devices in database (no compile is done)

A / character may be used in place of a + character. The default options are as follows:
+FM +ES +J +DC +Y9 -T -A +M +LNlst +O8hex -P -Z

If @filename appears on the CCSC command line, command line options will be read from the specified file. Parameters may appear on multiple lines in the file.

If the file CCSC.INI exists in the same directory as CCSC.EXE, then command line parameters are read from that file before they are processed on the command line.

Examples:

```
CCSC +FM C:\PICSTUFF\TEST.C
 CCSC +FM +P +T TEST.C
```