Microchip Debugger (MDB) User's Guide

Notice to Development Tools Customers

Important:
All documentation becomes dated, and Development Tools manuals are no exception. Our tools and documentation are constantly evolving to meet customer needs, so some actual dialogs and/or tool descriptions may differ from those in this document. Please refer to our website (www.microchip.com/) to obtain the latest version of the PDF document.

Documents are identified with a DS number located on the bottom of each page. The DS format is DS<DocumentNumber><Version>, where <DocumentNumber> is an 8-digit number and <Version> is an uppercase letter.

For the most up-to-date information, find help for your tool at onlinedocs.microchip.com/.

© 2023 Microchip Technology Inc. and its subsidiaries

Table of Contents

© 2023 Microchip Technology Inc. and its subsidiaries

1. Preface

This chapter contains general information that will be useful to know before using the Microchip Debugger (MDB).

1.1 Conventions Used in This Guide

The following documentation conventions may appear in this document:

Table 1-1. Documentation Conventions

Description	Represents	Examples
Arial font:		
Italic characters	Referenced books	*MPLAB® X IDE User's Guide*
	Emphasized text	...is the *only* compiler...
Initial caps	A window	the Output window
	A dialog	the Settings dialog
	A menu selection	select Enable Programmer
Quotes	A field name in a window or dialog	"Save project before build"
Underlined, italic text with right angle bracket	A menu path	*File>Save*
Bold characters	A dialog button	Click **OK**
	A tab	Click the **Power** tab
N'Rnnnn	A number in verilog format, where N is the total number of digits, R is the radix and n is a digit.	4'b0010, 2'hF1
Text in angle brackets < >	A key on the keyboard	Press <Enter>, <F1>
Courier New font:		
Plain Courier New	Sample source code	`#define START`
	Filenames	`autoexec.bat`
	File paths	`c:\mcc18\h`
	Keywords	`_asm, _endasm, static`
	Command-line options	`-Opa+, -Opa-`
	Bit values	`0, 1`
	Constants	`0xFF, 'A'`
Italic Courier New	A variable argument	*`file`*`.o`, where *`file`* can be any valid filename
Square brackets [ ]	Optional arguments	`mcc18 [options] file [options]`
Curly brackets and pipe character: { | }	Choice of mutually exclusive arguments; an OR selection	`errorlevel {0|1}`
Ellipses...	Replaces repeated text	`var_name [, var_name...]`
	Represents code supplied by user	`void main (void) { ... }`

1.2 Recommended Reading

This user's guide describes how to use the MDB. Other useful documents are listed below. The following Microchip documents are available and recommended as supplemental reference resources.

Microchip Command-line Debugger Webinar

© 2023 Microchip Technology Inc. and its subsidiaries

This webinar gives an introduction to the command-line debugger and provides useful examples. The webinar is available on Microchip's web site:

www.microchip.com/webinars.microchip.com/WebinarDetails.aspx?dDocName=en565588

Multi-Tool Design Advisory (DS51764)

This small document presents guidelines and implementation considerations to ensure proper interfacing to the various development tools.

MPLAB X IDE WebHelp

This is an essential document to be used with any Microchip hardware tool.

This is an extensive help file for the MPLAB X IDE. It includes an overview of embedded systems, installation requirements, tutorials, details on creating new projects, setting build properties, debugging code, setting configuration bits, setting breakpoints, programming a device, etc. This help file is generally more up-to-date than the printable PDF of the user's guide (DS50002027) available as a free download at www.microchip.com/mplabx/.

Processor Extension Pak and Header Specification (DS50001292)

This booklet describes how to install and use headers. Headers are used to better debug selected devices, without the loss of pins or resources. See also the PEP and Header online Help file.

Transition Socket Specification (DS51194)

Consult this document for information on transition sockets available for use with headers.

Release Notes for MDB

For the latest information on using the MDB, go to the MPLAB X IDE Learn & Discover tab, click the Users Guide & Release Notes icon, and locate the Readme for MDB. The release notes (readme) contain updated information and known issues that may not be included in this user's guide.

MDB WebHelp

A comprehensive online help for the MDB is available on `onlinedocs.microchip.com`. This help file may be more up-to-date than the printed documentation.

© 2023 Microchip Technology Inc. and its subsidiaries

2. Introduction

The Microchip Debugger (MDB) is a command-line debugger interface to Microchip's hardware and software development tools. As an alternative to using the Microchip MPLAB® X IDE (Integrated Development Environment) graphical interface, the MDB facilitates debugging devices through a Command Prompt interface and can program a production image for testing purposes.

The MDB is designed for engineers who prefer to use the Command Prompt. The command-line interface to the debugger is faster and allows more extensive testing to be performed. This is especially helpful when a task is repetitive, such as debugging an issue that is difficult to resolve, or when there is automation of a testing procedure.

The MDB can be used with a script or batch file. The MDB can be used with these tools:

- MPLAB® ICD 3 In-Circuit Debugger
- MPLAB® ICD 4 In-Circuit Debugger
- PICkit™ 3 In-Circuit Debugger/Programmer
- MPLAB® PICkit 4 In-Circuit Debugger
- MPLAB® Snap In-Circuit Debugger
- MPLAB® REAL ICE™ In-Circuit Emulator
- MPLAB® ICE 4 In-Circuit Emulator
- MPLAB® PM3 Device Programmer
- MPLAB SIM Software Simulator
- Licensed third party programmers and debuggers
- SK (Starter Kits PKOB)
- EDBG (embedded debugger)
- Curiosity/Starter Kits (PKOB4)
- Microchip Curiosity Nano Debugger
- MCHV
- MICROSTICK
- Power Debugger
- JTAGICE3
- J-32
- J0Link
- Atmel-ICE
- Atmel Embedded Debugger (EDBG)
- Mini Embedded Debugger
- SEGGER SAME-ICE

2.1 Installation and Documentation

2.1.1 Install MPLAB X IDE

The MDB is automatically installed with the MPLAB X IDE. To download the latest version, go to the Microchip web site (www.microchip.com).

Generate an .elf file for debugging (if simply programming a device, a hex file is sufficient). The project can be built with MPLAB X IDE or using third-party compilers, as long as an .elf file is generated. The .elf file is a linked executable file that contains symbolic debugging information.

© 2023 Microchip Technology Inc. and its subsidiaries

2.1.2 Find MDB Documentation

The MDB supporting documentation are automatically installed with the MPLAB X IDE. There are several ways to access MDB documentation:

1. After invoking the MDB (see 2.2.2. Invoking the MDB), type `help doc`. This command displays the instructions on where to locate MDB documentation.The help found here is provided in a command line prompt screen. For more extensive help, refer to the online help located at onlinedocs.microchip.com/ where you can search for “Microchip Debugger.”
2. A PDF of the MDB User's Guide can be found on the MPLAB X IDE web page www.microchip.com/en-us/development-tools-tools-and-software/mplab-x-ide, under the **Documentation** tab.
3. View latest release notes for the MDB in the `docs` folder under the latest installed version of MPLAB X IDE, e.g., `C:\Program Files\Microchip\MPLABX\<version>\docs`.

2.2 Getting Started

Typically, you can use the defaults when invoking the MDB. For more detailed information see 2.3. Debugging Methods.

2.2.1 Command Line Parameters

Before invoking the MDB, you may want to set certain command line options and arguments.

To view the options, type `mdb --help` in the Command Prompt. This only displays help information (refer to the table below) for the command line parameters (options and arguments) and then exits MDB. To find documentation about the MDB, such as online help, user’s guide (PDF) or the Readme for MDB, type `help doc`. This command displays the instructions on where to locate MDB documentation.

To set any parameters, use the following format in the Command Prompt (put a space between the entries as shown):

```
mdb [options] [commandFile]
```

You can use these commands to pass a command file to the MDB.

Table 2-1. Command Line Parameters Help

Option	Meaning	Examples

`-h,--help`	Show the list of classes of commands	`mdb -h`
		or
		`mdb --help`

Argument	Meaning	Example
`commandFile`	Run the specified file with the MDB commands for scripting. Also see 2.4. Running a Command File Method	`mdb MyScriptingFile.txt`

2.2.2 Invoking the MDB

Use the Command Prompt to invoke MDB.

In Windows® 10, use the Command Prompt (*Start>All Programs>Accessories>Command Prompt*),

The path to the MDB may vary depending on where the MPLAB X IDE is installed and which operating system is installed. See the following table for the various operating systems and paths. The `vn.nn` in the path represents the version number, for example v5.50. These paths are long so you may want to add them to your path variable.

© 2023 Microchip Technology Inc. and its subsidiaries

Table 2-2. Paths to the MDB by Operating System

Windows 64-bit Operating System
`c:\Program Files\Microchip\MPLABX\vn.nn\mplab_platform\bin>mdb.bat`
Linux® Operating System
`/opt/microchip/mplabx/vn.nn/mplab_platform/bin/mdb.sh`
macOS™ Operating System
`/Applications/microchip/mplabx/vn.nn/mplab_ide.app/Contents/Resources/mplab_platform/bin/mdb.sh`

Note: The `mdb.bat` and `mdb.sh` scripts do not need to be run from the directory where they were installed. If the directory where these scripts are installed is added to the system path, then `mdb.bat` and `msb.sh` may run from any directory.

2.3 Debugging Methods

You can run a test using either of the following methods:

- Entering Commands Method
- Running a Command File Method

Entering commands is the preferred method to run a test with MDB. It allows you to interact with the target application as it executes in simulation or on actual hardware. The result of each command is displayed one at a time, so that mistakes are more easily understood and corrected (see 2.3.1. Entering Commands Method).

The Running a Command File method cannot be used after the MDB has been invoked. The command file is included as a parameter in the command line when invoking the MDB (see 2.4. Running a Command File Method).

2.3.1 Entering Commands Method

Note: Although the MPLAB X IDE can run multiple tools simultaneously, the MDB will run only one tool at a time. However, you can have multiple instances of the MDB running. Refer to 2.5. Using Multiple Instances of the MDB for details.

Entering commands is a step-by-step method to run a test with MDB. Once the MDB is running, you can start entering commands. Note that while the MDB commands are not case-sensitive, the property options and file names are case-sensitive.

Type help for a list of classes of commands in MDB. Refer to 3. MDB Reference, Table 3-1.

For other commands available, see 3. MDB Reference, Table 2-2 through Table 2-10.

The following sections describe these topics:

- Programming a Production Image for Testing Purposes
- Debugging a Device

2.3.2 Programming a Production Image for Testing Purposes

The MDB can be used to program a production image for testing purposes.

Note: ***The MDB should be used only for debugging purposes.*** For programming devices, use the IPECMD tool or the IPECMDBoost (for improved speed). Refer to the Release Notes for IPE Command Line Interface (Readme for IPECMD) located in the MPLABX install folder, for example, `\Microchip\MPLABX\vx.xx\docs`, where `vx.xx` represents the version of MPLAB X IDE.

Note: When programming a device, you must select a device first.

The file or hardware tool you need to use for MDB cannot be active or open simultaneously in the MPLAB X IDE, IPE, or a third party program. Make sure you close (or make inactive) the file or hardware tool before you attempt to use it with the MDB.

© 2023 Microchip Technology Inc. and its subsidiaries

1. Select the device by entering the command:
   `Device [device name]`
   For example: `Device PIC18F66K22`
2. Use the set command to select any options you want to use (see Table 3-7 or Table 3-8).
3. Select the hardware tool. To verify the supported tools, type:
   `Hwtool Supported`
   The MPLAB ICD 3/4, MPLAB REAL ICE, PICkit 3/4 and Simulator are for programming and debugging, while the MPLAB PM3 is for programming only. To select the hardware tool, type the command:
   `Hwtool [tool name]`
   For example: `Hwtool SIM`
4. If the project was already built, an elf file was generated. To program the device with the elf or hex file, enter the command:
   `Program "[location of the elf or hex file]"`
   For example:
   ```
   Program
   "C:\MDBTestExample\Build\test\preprocess\files\dist\test_IO_Button.elf"
   ```
   If you are using SIM (Simulator) as the hardware tool and the project needs an scl file, it can be set up by using the command:
   `Stim "[location of the scl file]"`
   For more information, use the command Help Stim. You can use Stimulus to set pin injection and/or register injection.

A "Program succeeded" message displays after programming is complete. A verify is automatically performed during a programming sequence.

2.3.3 Debugging a Device

Use the following commands to debug a device.

- **Reset** – refer to the device data sheet for Reset information. If a Reset is needed for debugging purposes: first, halt the target; then, enter the command:
  `Reset`
- **Set Breakpoint** – there are two ways to set a breakpoint for debugging:
  Set a breakpoint by source-line-number using the command:
  `Break filename: linenumber`
  For example: `Break main.c:53`
  Set a breakpoint at an absolute address using command:
  `Break *address`
  For example: `Break *0x108`
- **Set Watchpoint** – to set a watchpoint for debugging:
  Set a watchpoint by specifying an address and the type of watch using the command:
  `Watch address breakontype`
  For example: `Watch 0xa0007ff0 R`
  or
  `Watch address breakontype[:value] [passcount]`
  For example: `Watch 0xa0007ff0 R:0xf 1`
- **Delete Breakpoint** – to delete a breakpoint, use the command:
  `Delete [breakpoint number]`

© 2023 Microchip Technology Inc. and its subsidiaries

If no argument is specified in this command, it will delete all breakpoints.

- **Run Program** – the Run command can be used to run the program until it reaches a breakpoint.
- **Step Through** – to step through the program, use the Step command or Next command.
- **See Variable Value** – a Print [variable] command can be used to see the value of a variable or an SFR.
- **Exit** – use the Quit command to exit the MDB.

2.4 Running a Command File Method

Note: Although the MPLAB X IDE can run multiple tools simultaneously, the MDB will run only one tool at a time. However, you can have multiple instances of the MDB running. Refer to 2.5. Using Multiple Instances of the MDB for details.

If programming and debugging needs to be done frequently or multiple times, run the test by running a command file. This is more efficient than entering the commands repeatedly. Put all the commands in a file and run the MDB using this command file in the Command Prompt, for example:

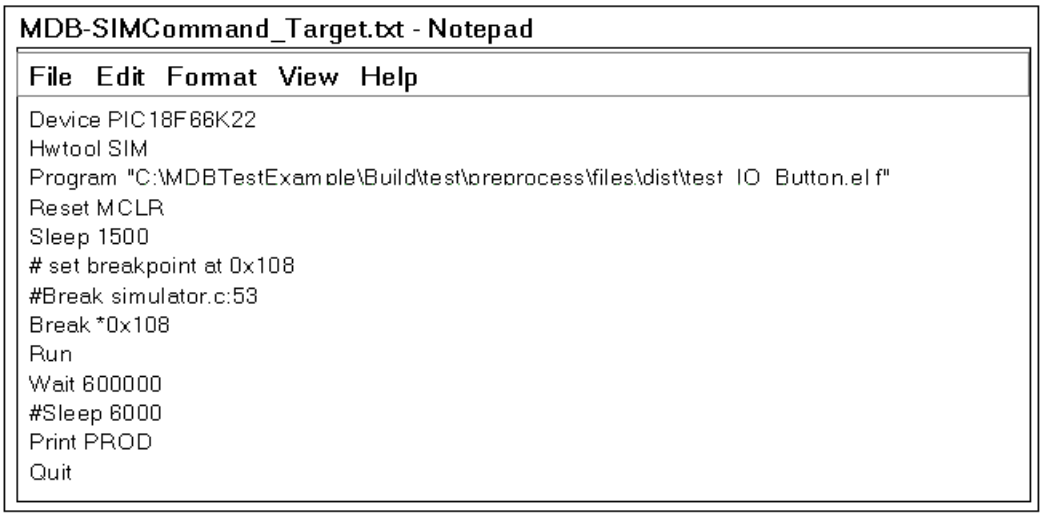
```
C:\Program Files\Microchip\MPLABX\vn.nn\mplab_ide\bin>mdb.bat
<commandfile.txt>
```

The following is an example of a command file:

```
C:\MDB-SIMCommand_Target.txt
```

A line starting with # means that it is a comment. A Sleep command should be added to make sure the MDB has enough time to finish the previous command before it executes the next command. The MDB will run all the commands in the command file sequentially.

Figure 2-1. Example of Running a Command Line



Creating a Printable Log File

Redirecting output to a file is a general option that can be executed from the command prompt and is not specific to the MDB batch file. Redirecting output to a printable text file can be more useful for examining errors than looking at the Command Prompt window.

To create a printable file, open the MDB.bat file, and modify it by adding `>>%mplabx_dir%\bin\mdblog.txt` at the end of the batch file. This instructs the batch file to create the mdblog.txt file, which can be printed.

© 2023 Microchip Technology Inc. and its subsidiaries

2.5 Using Multiple Instances of the MDB

Using multiple instances of the MDB is similar to using multiple instances of the MPLAB X IDE. Some set up is required before using hardware tools (PICkit 4, etc.) with an instance of the MDB. Refer to the MPLAB X IDE online help “Before You Begin,” and “Launch Multiple Instances of the IDE” for instructions on setting up the hardware tools and formatting the MCHPDEFPORT file. After any hardware tool setup is complete (to assign the appropriate driver for the tool), an instance of the MDB may be invoked from the bin directory of the installation.

© 2023 Microchip Technology Inc. and its subsidiaries

3. MDB Reference

3.1 Help Commands

There are help commands available in MDB.

Type `help` followed by a class name for a list of commands in that class (see 3.2. Classes of Commands).

Type `help` followed by a command name for full documentation (see 3.3. List of commands Within Classes).

Type `help doc` to see how to access MDB documentation.

To access more detailed documentation for the MDB:

1. The MDB Help file provides the most **up-to-date** information and is located at: `onlinedocs.microchip.com` where you can search for "Microchip Debugger."
2. The *Microchip Debugger (MDB) User's Guide* (`.pdf`) can be found on the MPLAB X IDE web page www.microchip.com/en-us/development-tools-tools-and-software/mplab-x-ide, under the **Documentation** tab.

3.2 Classes of Commands

Type `help` for a list of classes of commands in MDB.

Table 3-1. MDB Classes of Commands

Class	Description
breakpoints	Making program stop at certain points
data	Examining/changing data
deviceandtool	Selecting debug tool and device
networktool	Configure network tools
others	Miscellaneous commands
programming	Programming device and its relative functions
running	Running the program
stack	Examining stack

3.3 List of commands Within Classes

Note: MDB commands are not case-sensitive. However, when using the SET command, where tool option properties are passed as parameters, the parameter portion of the command line entered is case-sensitive.

For a list of all commands within a particular class, type `help` followed by the class name. The `help` command can also be abbreviated to `h`. See the following tables for information about each class of commands.

For documentation on a particular command, type `help [command or class of commands]` to display information about the command. For example, if you type `help breakpoints` or `h Breakpoints` the MDB displays information about the break, watch, delete and halt commands.

The following sections provide information on commands.

3.3.1 Breakpoint Commands

To display information about the breakpoints commands available in MDB, type `help breakpoints`. The following table provides additional information for this class of commands

© 2023 Microchip Technology Inc. and its subsidiaries

Table 3-2. Breakpoint Commands

Command	Description

Break	Sets a breakpoint at the specified source line number.
	Command format:
	`break filename:linenumber [passCount]`
	Example:
	`break newmain.c:142 4`

	Sets a breakpoint at an absolute address.
	Command format:
	`break *address [passCount]`
	• `address` – the address of the program memory to break on – use the command: `'print /a'` to get a symbol address.
	• `passCount` – the parameter is optional. Indicates the number of times the break 'on condition' is met before the program halts.
	Example:
	`break *0x9d0000cc 5`
	MDB assigns a breakpoint number and returns:`Breakpoint 0 at 0x9d0000cc: file newmain.c, line 16.`

	Sets a breakpoint at the beginning of the function.
	Command format:
	`break function_name [passCount]`
	Example:
	`break function_foo 5`

Delete	Deletes a breakpoint – if no argument is specified, this deletes all breakpoints.
	You can abbreviate this command as `d`.
	Command format:
	`delete [breakpoint number]`
	`d [breakpoint number]`
	The breakpoint number is generated by MDB for the `Break` and/or `Watch` commands.
	Examples:
	`delete` or `D`
	`delete 1` or `d 1`

Halt	Stops the debugger program.

© 2023 Microchip Technology Inc. and its subsidiaries

.....continued

Command	Description

Watch	Sets a data breakpoint at the specified memory address, variable name, or an SFR (special function register).
	Command format:
	`Watch address breakonType[:value] [passCount]`
	• `address` – the name of a global variable, SFR, or data memory address to be watched.
	Use command `'print /a'` to get a variable address.
	• `breakonType:`
	`R` -- Read.
	`W` -- Write.
	`RW` -- Read or Write.
	• `value` – this parameter is optional. If it is specified, the program will break only when the value held in the data memory matches the specified value.
	• `passCount` – this parameter is optional. The number of times the breakon condition is met before the program breaks.
	Examples:
	`watch 0xa0007ff0 R:0xf 1`
	`watch 0xa0007ff0 R:10 1`
	`watch my_Variable W 4`
	MDB will assign and return the watchpoint number, for example: `Watchpoint 1`.

© 2023 Microchip Technology Inc. and its subsidiaries

3.3.2 Data Commands

To display information about the data commands available in the MDB, type `help data`. The table below provides additional information for this class of commands.

Table 3-3. Data Commands

Command	Description

Print	Prints a variable with optional formatting.
	Command format:
	`print [/f] [/datasize:value] variable`
	• `f` - Optional format letter.
	The format letters supported are:
	`x` - Print as integer in signed hexadecimal.
	`d` - Print as integer in signed decimal.
	`a` - Print the address of a symbol.
	• `datasize:value` – optional data size.
	Variable in assembly code might not have data size information. The user can specify the data size if the `.elf` file does not have the size information.
	The values supported are:
	`1` - The data size is 1 byte.
	`2` - The data size is 2 bytes.
	`4` - The data size is 4 byte.
	Use this command (not case sensitive) to display the pin information.
	Command format:
	`print pin pinName`
	Example:
	`print pin RA0`
	This command will print Pin, mode, Value, and Owner or Mapping.
	• For Pin, it displays the name of the signal that the user types to find the pin.
	• For Mode, it displays the A/D state and I/O state.
	• For Value, it displays HIGH/LOW for Digital mode or the HIGH/LOW nominal voltage for Analog mode.
	• For Owner or Mapping, it displays the pin owner and all the signals in this pin.
	• The owner of the pin is the signal with parentheses.

Stim	Specifies a simulator SCL stimulus file to use.
	This loads the specified SCL stimulus file into the simulator, or if no path to the file is specified, it clears a loaded file.
	Note: If the path or filename has spaces in it, you must use the quotation marks, as shown below. If there are no spaces in the path of filename, the quotation marks are not needed.
	Command format:
	`Stim "[path to file]"`
	`Stim`

© 2023 Microchip Technology Inc. and its subsidiaries

.....continued

Command	Description

Write	Use this command to write to memory.
	Command format:
	`write [/t] addr word1 word2 ... wordn`
	• `t` - the type of memory. The type of memory is any of the following:
	`r` – File Registers (RAM) memory. This is the initial default.
	`p` – Program (flash) memory.
	`e` – EE Data memory.
	Each time you specify a memory type with `write`, that type becomes the default memory the next time you use `write`.
	• `addr` – the starting address where you want MDB to begin writing to memory.
	• `word` – the following values will be written to successive words of memory.
	Use this command to set a pin high or low when the simulator is used as a debug tool.
	Command format:
	`write pin pinName pinState`
	Example:
	`write pin RA0 high`
	Use this command to set a the voltage of a pin when the simulator is used as a debug tool.
	Command format:
	`write pin pinName pinVoltage`
	Example:
	`write pin RA0 3.3V`

© 2023 Microchip Technology Inc. and its subsidiaries

.....continued

Command	Description
x	Examine memory. You can use the command `x` (for examine) to examine memory in any of several formats, independent of your program's data types. **Command format:** `x [/tnfu] [addr]` • `t` – the type of memory. **Note:** Each time you specify a memory type with `x`, that type becomes the default memory the next time you use `x`. The type of memory is any of the following: `r` – File Registers (RAM) memory. This is the initial default. `p` – Program (flash) memory. `m` – Memory-mapped control registers (PIC32 peripheral memory). `e` – EE Data memory. `u` – User ID memory. `U` – Unique Device Identifier (UDID). **Note:** This is case sensitive. `D` – Device ID memory. **Note:** This is case sensitive. • `n` – the repeat count. Repeat count is a decimal integer; the default is `1`. It specifies how much memory (counting by units `u`) to display. `f` – the display format. This is one of the formats used by print (`x, d, o, f, s`), as well as "`i`" (for machine instructions). The default is '`x`' (hexadecimal) initially. The default changes each time you use `x`. `u` – the unit size. Each time you specify a unit size with `x`, that size becomes the default unit the next time you use `x` (for the '`s`' and '`i`' formats, the unit size is ignored and is normally not written). The unit size is any of following: `b` – Bytes. `h` – Halfwords (two bytes). `w` – Words (four bytes). This is the initial default. • `addr` - the starting display address where you want MDB to begin displaying memory. The `addr` can be a literal or a symbol name. The default for `addr`, if not specified, is taken as the value just after the last address examined. However, several other commands also set the default address: `info breakpoints` (to the address of the last breakpoint listed); `info line` (to the starting address of a line); and `print` (if you use it to display a value from memory).

© 2023 Microchip Technology Inc. and its subsidiaries

3.3.3 Device and Tool Commands

To display information about the device and tool commands available in MDB, type `help deviceandtool`. The table below provides additional information about these commands.

Table 3-4. Device and Tool Commands

Command	Description

Device	Sets the name of the target device.
	Command format:
	`Device devicename`
	Example:
	`Device PIC32MX795F512L`

© 2023 Microchip Technology Inc. and its subsidiaries

.....continued

Command	Description

Hwtool	Sets the debug tool or lists all the available hardware tools on the system.
	Command format:
	`hwtool [toolType] [-p] [index]`
	Use the `-p` option to connect to the tool for programming only.
	Use the `index` option to select the tool if there are more than one instance of a tool type.
	Use the `hwtool` command to find the index number of the tool.
	Use the `hwtool supported` command to see all the supported tool types.
	To find a tool using the tool's serial number, provided it starts with `<sn>`, use the following command format.
	Command format:
	`hwtool toolType <sn>`
	Example:
	`hwtool icd4 <sn>BUR204420097`
	To find a network tool using the tool's IP address, provided it starts with `<ipa>`, and uses the format `xxx.xxx.xxx.xxx`, use the following command format.
	Command format:
	`hwtool toolType <ipa>`
	Example:
	`hwtool icd4 <ipa>010.016.024.031`
	Use the `hwtool supported` command to see all the supported tool types. This is a list of supported tooltypes:
	• `Licensedprogrammer` – third party programmer
	• `atmelice` - Atmel-ICE
	• `EDBG` - Atmel Embedded Debugger (EDBG)
	• `MEDBG` - Mini Embedded Debugger
	• `PKOBNANO` - Microchip Curiosity Nano Debugger
	• `powerdebugger` - Power Debugger
	• `jtagice3` - JTAGICE3
	• `ICD3` – MPLAB ICD 3 In-Circuit Debugger
	• `ICD4` – MPLAB ICD 4 In-Circuit Debugger
	• `ICD5` - MPLAB ICD 5 In-Circuit Debugger
	• `j32` - J-32
	• `jlink` - J-Link
	• `seggersameice` - SEGGER SAME-ICE
	• `PICkit3` – MPLAB PICkit 3 In-Circuit Debugger
	• `PICkit4` – MPLAB PICkit 4 In-Circuit Debugger
	• `PICkit5` - MPLAB PICkit 5 In-Circuit Debugger
	• `Licenseddebugger` – third party debugger
	• `PKOB4` - Curiosity/Starter Kits (PKOB4)
	• `SK` - Microchip Starter Kit (PICkit On Board - PKOB)
	• `PM3` – MPLAB PM3 Programmer
	• `RealICE` – MPLAB REAL ICE In-Circuit Emulator
	• `ICE4` – MPLAB ICE 4 In-Circuit Emulator
	• `SIM` – Simulator
	• `mchv` - MCHV
	• `microstick` - MICROSTICK
	• `SNAP` – MPLAB Snap In-Circuit Debugger

© 2023 Microchip Technology Inc. and its subsidiaries

3.3.4 Network Tool Commands

The network tool commands are used for tools not physically connected to the local computer but through a network via ethernet, wifi, USB, etc. The default is USB.

To display information about the network tool commands for a selected tool available in the MDB, type `help networktool`. The table below provides additional information for this class of commands.

Table 3-5. Network Tool Commands

Command	Description

AddIP	Adds a user-specified newtork tool with the user-specified IP address.
	Command format:
	`addip userSpecifiedIPAddress xxx.xxx.xxx.xxx`
	Note: The IP address needs to be in the form `xxx.xxx.xxx.xxx`.
	Example:
	`Addip 192.170.001.001`

Configneworktool	Reads or writes the network settings for the selected tool. **Command format:** `configneworktool [toolType] [index] [read\write]` • toolType - If MDB is not currently connected to a tool (via the `hwtool` command), tooltype must be supplied (e.g. ice4). Otherwise, toolType should not be used. • index - The index of the specific tool (as shown by the `hwtool` command). The default index is zero and will be applied if no index is provided. If MDB is already connected to a tool (via the `hwtool` command), the index parameter should not be used. • read/write - The read/write parameter is optionial and indicates whether MDB will read the current network settings off of the tool or write settings to the tool. Use, `read` to read settings and, `write` to write settings. the default is to write the settings and will be applied if no read/write setting is provided. This setting is valid regardless of whether or not MDB is currently connected to a tool (via the `hwtool` command). However, if MDB is currenttly connected to a tool (via the `hwtool` command), MDB will disconnect from the tool after a successful write of the network settings. Use the set command to set each specific network setting. The syntax for the network settings is as follows: – Connection Type: `name = networktool.connection.type` valid values = ethernet, usb, ethernetstaticip, wifista, wifiap, (default is usb). – Static IP address: `name = networktool.static.ip` Data must be in the form (`x.x.x.x`) where x's are positive numbers from 0 to 255 (default = `0.0.0.0`). – Gateway: `name = networktool.gateway` Data must be in the form (`x.x.x.x`) where x's are positive numbers from 0 to 255 (defualt = `0.0.0.0`). – Subnet Mask: `name = networktool.subnet.mask` Data must be in the form (`x.x.x.x`) where x's are positive numbers from 0 to 255 (default = `255.255.255.255`). – Network SSID: `name = networktool.ssid` value = any valid network name (default is an empty string). – Security Type: `name = networktool.security.type` valid values = wep, enterprise, personal, none, (default is none). – User Name: `name = networktool.user.name` Any valid user name (default is an empty string). – Password: `name = networktool.password` Any valid password (default is an empty string). When writing new settings, MDB performs a read modify write operation so that only the settings that have been explicitly declared, through the "set" command, will be applied to the tool.

© 2023 Microchip Technology Inc. and its subsidiaries

.....continued	
Command	**Description**
RemoveIP	Remove a user-specified network tool with the user-specified IP address. **Command format:** `removeip [userSpecifiedIPAddress]` **Examples:** To remove a user-specified IP address:`removeip 192.170.001.001` To remove all user-specified network tools: `removeip all`

3.3.5 Others Commands

To display information about the others commands available in MDB, type `help others`. The following table provides additional information for these commands.

Table 3-6. Others Commands

Command	Description

Echo	Echo is a command typically used in command files and batch files to output status text to the screen or a file. The echo command will print text surrounded by `/* */`. Use `\n` in the text to print a new line.
	Command format:
	`echo text`
	Example:
	`echo Hello World`
	Result:
	`/*Hello World*/`
	This command prints text only. To print variables or other information, use commands such as `print`, `info`, `list`, etc.

Help	`help others` – Prints a list of commands.
Quit	`quit` - Exits the debugger.

Set	The tool property name and value are from the project properties that are selected when creating the project in MPLAB X IDE. **IMPORTANT:** *The Set command, including the tool property options, must be executed before the Hwtool command is issued, otherwise the changes to the tool properties will be ignored.*
	Command format:
	`Set tool-property-name value`
	Example:
	`Set programoptions.eraseb4program true`
	Refer to Table 3-7 for other tool properties options that can used with the `Set` command.
	Refer to Table 3-8 for simulator options that can be used with the `Set` command.

Sleep	Makes the current script processor sleep until specified milliseconds have elapsed.
	Command format:
	`Sleep milliseconds`
	Example:
	`Sleep 10`

Wait	The `Wait` command makes the current script processor wait until the debugger halts before processing the next command.
	Command format:
	`Wait`
	`Wait Milliseconds` makes the processor process the next command if the debugger does not halt and milliseconds have elapsed.
	Command format:
	`Wait [milliseconds]`

© 2023 Microchip Technology Inc. and its subsidiaries

.....continued

Command	Description
cd	This command changes the directory that you are currently working in to the directory you designate. **Command format:** `cd [directory]`

info	Prints a table of all breakpoints that have been set and not deleted.
	Optional argument `n` means "print information only" about the specified breakpoint.
	For each breakpoint the following columns are printed:
	• Breakpoint Numbers
	• Enabled or Disabled: Enabled breakpoints are marked with `'y'` . Disabled breakpoints are marked with `'n'`
	• Address - Where the breakpoint is in your program, as a memory address.
	• What - Where the breakpoint is in the source for your program: as a file and line number.
	Command format:
	`info breakpoints [n]`
	`info break [n]`

list	The list command prints (displays) the source code for the current PC location, or a different file, if specified. The `list` command displays lines from a source file. By default, 10 lines are displayed. `list` - displays 10 lines (5 above, 5 below) around the current line `list linenum` - displays 10 lines around a given line `list first,` - displays 10 lines from first line specified `list ,last` displays 10 lines up to last line specified `list first,last` - displays all lines from the first to the last line specified `list –` - displays 10 previous lines from the last output `list +` - displays 10 more lines from the last output `list function` - displays 10 lines around the given function `list file:linenum` - displays 10 lines around the given line in a given file `list file:function` - displays 10 lines around the given function in a given file `set system.listsize count` - changes the number of lines shown Using 0 or -1 means unlimited list size.
pwd	The `pwd` command displays the current working directory. **Command format:** `pwd`

© 2023 Microchip Technology Inc. and its subsidiaries

.....continued	
Command	**Description**

Pack	Display the information (version and path) of the pack currently selected, as well as available packs.
	Command format:
	`pack`
	Example: (Use PIC12F675 as example)
	`pack`
	Result:
	`PIC10-12Fxxx_DFP, version: 1.6.66`
	`path: C:\earlydfpsoften\packs\Microchip\PIC10-12Fxxx_DFP`
	`Available packs:`
	`1.6.66 (currently selected)`
	`1.5.61`
	Switch pack to specify the pack that the user wants to use.
	Command format:
	`pack [version]`
	Example: (Use PIC12F675 as example)
	`pack 1.5.61`
	Result:
	`Change the in used pack to 1.5.61`

Group	Create a register group for Display command.
	Command format:
	`Group groupName <registers>`
	Example:
	`Group xregs <W0,W1,W2,W3,W4>`

Display	Examine a group of registers defined by user (refer to the Group command). **Command format:** Display /g groupName **Example:** `Display /g xregs`

© 2023 Microchip Technology Inc. and its subsidiaries

3.3.5.1 Tool Property Options Used with the Set Command

The following table provides additional information for tool property options used with the Set command.

Important: The `set` command, including the tool property options, must be executed before the `Hwtool` command is issued, otherwise the changes to the tool properties will be ignored.

Notes:

1. MDB commands are not case-sensitive. However, when using the `set` command, where tool option properties are passed as parameters, the parameter portion of the command line entered is case-sensitive.
2. Tool property options that you want to use with the `set` command must be selected before using the `Hwtool` command.

Table 3-7. Tool Property Name Options Used by the Set Command

Tool Property Name	Value

`AutoSelectMemRanges`
Determines whether the debugger will automatically select the areas of memory and program memory ranges to program.
If set to `auto`, the debugger will automatically select the memory and ranges. `Manual` means the memories and ranges will be determined by the memories properties below.
Example:
`set AutoSelectMemRanges auto`
	`auto` or `manual`

`communication.interface`
Set the communication interface to *nnn* (*nnn* is the value).
Example:
`set communication.interface jtag`
	`jtag, swd, updi, dw, isp, pdi, tpi`

`communication.interface.jtag`
Select the JTAG method.
Example:
`set communication.interface. jtag 2wire`
	`2wire` or `4wire`

`communication.speed`
Set the communication speed to *nnn* (*nnn* is the value).
Example:
`set communication.speed 0.100`
	a decimal value (in MHz), dependent on the device

`debugoptions.useswbreakpoints`
`True` indicates that software breakpoints will be used for program address breakpoints, `false` indicates that hardware breakpoints will be used (does not apply to PICkit 3).
Example:
`set debugoptions.useswbreakpoints true`
	`true` or `false`

`debugoptions.simultaneous.debug`
`True` indicates that the project is part of a multi-core simultaneous debug session. If `false`, it is not.
Example:
`set debugoptions.simultaneous.debug true`
	`true` or `false`

`freezeperiphs` Check to freeze all other peripherals not specifically mentioned above. Some peripherals have no freeze configurability and cannot be controlled by the debugger. Select peripherals to freeze, or not freeze, on program halt. Options available depend on device chosen.	`true` or `false`

© 2023 Microchip Technology Inc. and its subsidiaries

.....continued

Tool Property Name	Value

`hwtoolclock.frcindebug`
If `true`, it will use FRC in debug mode (requires reprogramming); if `false`, it will not. Switch to the internal oscillator when the target is halted and in debug mode. This option can be used to speed up debugging operations if the internal oscillator is faster than the run time oscillator the target is using. The target will automatically be switched back to the run time oscillator when the target exits debug mode (runs).
Example:
`set hwtoolclock.frcindebug true`
	`true` or `false`

`hwtoolclock.instructionspeed` This will tell MPLAB X what speed (in instructions per second) the target will be running at. This does not set the target instruction speed; that is done through the configuration bits. This setting is used in conjunction with the Instruction Speed Units property. **Example:** `set hwtoolclock.instructionspeed 4`	

`hwtoolclock.units`
Determines the units of the Target run-time instruction speed property entered directly above this one.
Example:
`set hwtoolclock.units mips`
	`mips` or `kips`

`memories.configurationmemory`
If `true`, the configuration memory will be programmed; if `false`, it will not (always programmed in debug mode).
Example:
`set memories.configurationmemory true`
	`true` or `false`

`memories.configurationmemory2`If `true`, the configuration memory partition 2 will be programmed; if `false`, it will not (always programmed in debug mode). **Example:** `set memories.configurationmemory2 true`	`true` or `false`

`memories.dataflash`
This is an area of flash that is also used to store data. Conceptually, this is the same a flashdat, but is different in that the firmware must know the difference between the two types of memory so they have different designations. If the database returns `true` to the `hasDataFlashSpace()` call, the memory is present for this device.
Example:
`set memories.dataflash true`
	`true` or `false`

`memories.flashdata`
This is an area of flash that is used to store data. If the database returns `true` to the `hasFlashData()` call, the memory is present for the device.
Example:
`set memories.flashdata true`
	`true` or `false`

`memories.instruction.ram`
If `true`, the instruction ram will be programmed; if `false`, it will not (for debugging only).
Example:
`set memories.instruction.ram true`
	`true` or `false`

`memories.instruction.ram.ranges`
Instruction ram Range(s). Each range must be two hex numbers, representing the start and end addresses of the range, separated by a dash. Ranges must be separated by a comma (e.g. 0-ff, 200-2ff).
Example:
`set memories.instruction.ram.ranges 0-ff,200-2ff`
	Hex value

© 2023 Microchip Technology Inc. and its subsidiaries

.....continued

Tool Property Name	Value

`programoptions.preserveprogramrange`
If `true`, preserve program memory; if `false`, it will not.
Example:
`set programoptions.preserveprogramrange true`
	`true` or `false`

`programoptions.preserveprogram.ranges`
Define preserve program range(s). Areas are reserved by reading them into MPLAB and then programming them back down when a program operation occurs. Thus the preserved areas must lie within a memory range that will be programmed. Ranges must be aligned on a program memory address boundary. Each range must be two hex numbers, representing the start and end addresses of the range, separated by a dash. Ranges must be separated by a comma (e.g. 0-ff, 200-2ff).
Example:
`set programoptions.preserveprogram.ranges 0-ff,200-2ff`
	Hex value

`memories.programmemory`
If `true`, the program memory will be programmed; if `false`, it will not.
Example:
`set memories.programmemory true`
	`true` or `false`

`memories.programmemory.ranges`
Program Memory Range(s). Enter ranges in hex (e.g. 0-ff, 200-2ff).
Example:
`set memories.programmemory.ranges 200-2ff`
	Hex value

`memories.programmemory.start`
The value represents the starting program memory address that the debug tool will begin programming.
Example:
`set memories.programmemory.start 0x0000`
	a string representing a long value

`memories.programmemory.end`
The value represents the ending program memory address that the debug tool will end programming.
Example:
`set memories.programmemory.end 0xFFFF`
	a string representing a long value

`memories.eeprom`
If `true`, the EEPROM memory will be programmed; if `false`, it will not.
Example:
`set memories.eeprom true`
	`true` or `false`

`memories.id`
If `true`, the user ID memory will be programmed; if `false`, it will not.
Example:
`set memories.id true`
	`true` or `false`

`memories.bootflash`
If `true`, the boot flash (PIC32 only) memory will be programmed; if `false`, it will not.
Example:
`set memories.bootflash true`
	`true` or `false`

`memories.aux`
If `true`, the auxiliary program memory (dsPIC®/PIC24 EP parts only) will be programmed; if `false`, it will not.
Example:
`set memories.aux true`
	`true` or `false`

© 2023 Microchip Technology Inc. and its subsidiaries

.....continued	
Tool Property Name	**Value**
networktool.connection.type	ethernet, usb, ethernetstaticip, wifista, wifiap(default is usb).
programoptions.donoteraseauxmem If true, the auxiliary memory will not be erased; if false it will. **Example:** set programoptions.donoteraseauxmem true	
programoptions.eraseb4program If true, the device will be erased before it is programmed; if false it will not. **Example:** set programoptions.eraseb4program true	true or false
programoptions.ledbrightness Sets the brightness of the LEDs on the hardware tool. Setting 1 is darkest and 10 is the brightest. The default is 5. **Example:** set programoptions.ledbrightness 7	1 to 10
programoptions.pgcconfig Sets the type of resistance to be applied to the PGC line. The default is pull down. The value of the resistance is set by the PGC resistor option. **Example:** set programoptions.pgcconfig pullup	none or pull up or pull down
programoptions.pgcresistor.value Sets the value of the resistance on the PGC line. Maximum value is 50 kohms. If PGC configuration is set to none, this value is ignored. **Example:** set programoptions.pgcresistor.value 4.7	0.1 to 50.0
programoptions.pgdconfig Sets the type of resistance to be applied to the PGD line. The default is pull down. The value of the resistance is set by the PGD resistor option. **Example:** set programoptions.pgdconfig pullup	none or pull up or pull down
programoptions.pgdresistor.value Sets the value of the resistance on the PGD line. Maximum value is 50 kohms. If PGD configuration is set to none, this value is ignored. **Example:** set programoptions.pgdresistor.value 4.7	0.1 to 50.0
programoptions.pgmentry.voltage Sets the method the hardware tool will use to put the target device in programming mode. For the low voltage method, Vpp will not exceed the Vdd supply voltage. Instead, a test pattern will be used on Vpp. For the high voltage method, a voltage in excess of 9 volts will be placed on Vpp. **Example:** set programoptions.pgmentry.voltage low	Low or High
programoptions.pgmspeed Sets the speed that the hardware tool will use to program the target. If programming fails, try a slower speed. The default is Med. **Example:** set programoptions.pgmspeed Min	Min or Med or Max

© 2023 Microchip Technology Inc. and its subsidiaries

.....continued

Tool Property Name	Value

`programoptions.preservedataflash`
If `true`, preserve data flash; if `false`, it will not.
Example:
`set programoptions.preservedataflash true`
	`true` or `false`

`programoptions.preservedataflash.ranges`
Define preserve program range(s). Areas are reserved by reading them into MPLAB and then programming them back down when a program operation occurs. Thus EEPROM must be selected for programming. Ranges must be aligned on a program memory address boundary. Each range must be two hex numbers, representing the start and end addresses of the range, separated by a dash. Ranges must be separated by a comma (e.g. 0-ff, 200-2ff).
Example:
`set programoptions.preservedataflash.ranges 0-ff,200-2ff`
	Hex value

`programoptions.preserveeeprom`
If `true`, preserve EEPROM memory; if `false`, it will not.
Example:
`set programoptions.preserveeeprom true`
	`true` or `false`

`programoptions.preserveuserid` If `true`, preserve ID memory; if `false`, it will not. **Example:** `set programoptions. preserveuserid true`	`true` or `false`

`programoptions.programcalmem`
If `true`, it will program calibration memory; if `false`, it will not. (The calibration value can be set on the linker properties page.)
Example:
`set programoptions.programcalmem true`
	`true` or `false`

`Program programoptions.programuserotp`
If `true`, it will cause the debug tool to program user OTP memory on the target device; if `false`, it will not. **WARNING:** Once programmed, user OTP memory can never be programmed again.
Example:
`set programoptions.programoptions.programuserotp true`
	`true` or `false`

`poweroptions.powerenable`
If `true`, the debug tool will supply target power at the default voltage for the tool. If `false` it will not supply target power.
Note: This property does not apply to MPLAB REAL ICE.
To set a non-default voltage for the target power, first set the `poweroptions.powerenable` to `true`, then set the voltage value where `n.n` represents the desired voltage:
`set voltagevalue n.n`
Example:
`set poweroptions.powerenable true`
`set voltagevalue 3.3`
	`true` or `false`

`SecureSegment.SegmentProgramming SegmentProgrammingAll`
If true, it permits programming to “Program Over Secure and Protected FLASH.” This property must be set prior to using the program operation on the MDB. Use the -p option to set the tool as a programmer if it’s for a production final image and not just a debug image.
Example:
`set SecureSegment.SegmentProgramming SegmentProgrammingAll true`
	`true` or `false`

© 2023 Microchip Technology Inc. and its subsidiaries

.....continued

Tool Property Name	Value

`system.disableerrormsg`
This option has been deprecated as of v3.15 but will still function with existing projects. For new projects, use the `<Link>`
`system.disableoutput` and
`<Link>system.yestoalldialog` options.
If `true`, the system will disable warnings and error messages and answer "yes" to all dialogs; if `false` the system will enable warning and error messages (this is the default).
Example:
`set system.disableerrormsg true`
	`true` or `false`

`system.disableoutput`
If `true`, the system will disable warnings and error message outputs but not disable dialogs; if `false` the system will enable warning and error message outputs and dialogs (this is the default).
Example:
`set system.disableoutput true`
	`true` or `false`

`system.yestoalldialog`
If `true`, the system will disable dialogs and answer "yes" to all of them; if `false` the system will enable dialogs (this is the default).
Example:
`set system.yestoalldialog true`
	`true` or `false`

`toolpack.updateoptions` Select to use the latest tool pack or a different version to support the project device.	

`xccodecoverage.reporttype`
Sets the xc code coverage report type. The report type can be `html`, `gcov`, `all`.
Example:
`set xccodecoverage.reporttype html`
`set xccodecoverage.reporttype gov`
`set xccodecoverage.reporttype all`
	a string representing the xc report type

`xccodecoverage.htmlreportpath`
Sets the html report path to generate report. If report path is not defined or null, then report will be generated in project location.
Example:
`set xccodecoverage.htmlreportpath d:\report`
	a string representing the xc report path

`xccodecoverage.replacehtmlreport`
Sets the property to true for overwriting the html report files, if already exists in the specified location.
Example:
`set xccodecoverage.replacehtmlreport true`
	true or false

© 2023 Microchip Technology Inc. and its subsidiaries

3.3.5.2 Simulator Options Used with the Set Command

The following table provides additional information for simulator options used with the set command.

Important: The `set` command, including the tool property options, must be executed before the `Hwtool` command is issued, otherwise the changes to the tool properties will be ignored.

Note: For the following table, the break options allow you to set the conditions that will cause program execution to halt. In general, the program will either break on option, ignore the option, or report the option.

Table 3-8. Simulator Options Used with the Set Command

Simulator Options	Values	Device or Runtime Dependent
warningmessagebreakoptions.[*see list in* Table 3-9] Sets the condition if an error or warning occurs. **Example:** `set warningmessagebreakoptions.W0001_CORE_BITREV_MODULO_EN Break`	`Break, Ignore, Report`	No

`codecoverage.enabled`
Enables or disables code coverage
Example:
`set codecoverage.enabled Disable`
	`Disable,`
	`Enabled_Reset_on_POR`
	`Enabled_Reset_on_Run`
		No

`codecoverage.enableoutputtofile`
Enables write to file
Example:
`set codecoverage.enableoutputtofile true`
	`true, false`	No

`codecoverage.outputtofile`
Absolute path to output file
Example:
`set codecoverage.outputtofile "c:\path\to\file.txt"`
	`String path`	No

`oscillator.auxfrequency`
Auxiliary PLL Frequency, used by PWM and ADC
Example:
`set oscillator.auxfrequency 4400`
	`Numeric`	Yes

`oscillator.auxfrequencyunit`
Auxiliary PLL Frequency Units
Example:
`set oscillator.auxfrequencyunit None`
	`Mega, Kilo, None`	Yes

`oscillator.frequency`
Instruction Execution Frequency
Example:
`set oscillator.frequency 4700`
	`Numeric`	No

`oscillator.frequencyunit`
Instruction Frequency Units
Example:
`set oscillator.frequencyunit Kilo`
	`Mega, Kilo, None`	No

© 2023 Microchip Technology Inc. and its subsidiaries

.....continued

Simulator Options	Values	Device or Runtime Dependent

`oscillator.rcfrequency`
RC Oscillator Frequency
Example:
`set oscillator.rcfrequency 4500`
	`Numeric`	No

`oscillator.rcfrequencyunit`
RC Oscillator Frequency Units
Example:
`set oscillator.rcfrequencyunit None`
	`Mega, Kilo, None`	No

`periphADC1.altscl`
Use MPLAB 8 style ADC
Example:
`set periphADC1.altscl true`
	`true, false`	Yes

`periphADC1.minTacq`
Specifies minimum acquisition time (Tacq) in seconds
Example:
`set periphADC1.minTacq 10`
	`Numeric`	Yes

`periphADC1.tacqunits`
Units for minimum acquisition time (Tacq)
Example:
`set periphADC1.tacqunits nanoseconds`
	`milliseconds, microseconds, nanoseconds`	Yes

`periphADC2.altscl`
Use MPLAB 8 style ADC
Example:
`set periphADC2.altscl true`
	`true, false`	Yes

`periphADC2.minTacq`
Specifies minimum acquisition time (Tacq) in seconds
Example:
`set periphADC2.minTacq 20`
	`Numeric`	Yes

`periphADC2.tacqunits`
Units for minimum acquisition time (Tacq)
Example:
`set periphADC2.tacqunits milliseconds`
	`milliseconds, microseconds, nanoseconds`	Yes

`uartNio.output`
Specifies location of UART output
`N` represents the UART number 1 through 6
Example:
`set uart1io.output file`
	`file, window`	Yes

`uartNio.uartioenabled`
If `true`, the system will enable the UART I/O; if `false` the system will disable it
`N` represents the UART number 1 through 6
Example:
`set uart1io.uartioenabled false`
	`true, false`	Yes

© 2023 Microchip Technology Inc. and its subsidiaries

.....continued		
Simulator Options	**Values**	**Device or Runtime Dependent**

`uartNio.outputfile`
Passes in a string containing the root (absolute path) of the file system to the file used for UART output
`N` represents the UART number 1 through 6
Example:
`set uart1io.outputfile "c:\path\to\outputfile.txt"`
	`Absolute path to file`	Yes

3.3.5.3 List of Error and Warning Messages

Table 3-9. Error and Warning Messages

Message	Meaning
Error Messages	
E0101_SIM_FAILED_TO_DISASSEMBLE_INSTRUCTION	E0101-SIM: Failed to disassemble instruction
E0102_SIM_INVALID_INSTRUCTION	E0102-SIM: Invalid Instruction address
E0103_SIM_FAILED_TO_PARSE_SCL	E0103-SIM: Failed to parse SCL
E0104_SIM_NO_VALUE_AVAILABLE	E0104-SIM: No value available
E0105_SIM_FAILED_TO_INIT_PERIPHERAL	E0105-SIM: Failed to init peripheral
E0106_SIM_ATTEMPT_TO_SET_PIN	E0106-SIM: Attempt to set pin instead of signal
E0107_SIM_SFR_UPDATE_EXCEPTION	E0107-SIM: An SFR Update method has thrown an exception: {0}
E0108_SIM_FAILED_OPERATION	E0108-SIM: Failed simulator operation: {0}
E0109_SIM_FAILED_TO_CLOSE_FILE	E0109-SIM: Failed to close UART file.
E0110_SIM_FAILED_TO_EXECUTE_INSTRUCTION	E0110-SIM: Failed to execute instruction at {0}.
E0111_SIM_UNEXPECTED_INSTRUCTION	E0111-SIM: Unexpected instruction {0}.
E0112_SIM_NO_PINS_IN_PIC_FILE	E0112-SIM: No pins were found in device data file (PIC).
Warning Messages	
W0001_CORE_BITREV_MODULO_EN	W0001-CORE: Bit-reversed and Modulo addressing are both enabled.
W0002_CORE_SECURE_MEMORYACCESS	W0002-CORE: Secure memory access, occurred from instruction at {0}.
W0003_CORE_SW_RESET	W0003-CORE: Software Reset Instruction called at PC
W0004_CORE_WDT_RESET	W0004-CORE: Watchdog Timer has caused a Reset.
W0005_CORE_IOPUW_RESET	W0005-CORE: Illegal opcode or uninitialized WREG has caused a reset.
W0006_CORE_CODE_GUARD_PFC_RESET	W0006-CORE: Illegal Code Guard Access caused Illegal opcode Reset.
W0007_CORE_DO_LOOP_STACK_UNDERFLOW	W0007-CORE: Nested do loop stack pop resulted in stack underflow.
W0008_CORE_DO_LOOP_STACK_OVERFLOW	W0008-CORE: Nested do loop stack push resulted in stack overflow.
W0009_CORE_NESTED_DO_LOOP_RANGE	W0009-CORE: The nested do level in CORCON<DL> was out of range: 0 <
W0010_CORE_SIM32_ODD_WORDACCESS	W0010-CORE: An odd address WORD was accessed in RAM or PROM (flash).
W0011_CORE_SIM32_UNIMPLEMENTED_RAMACCESS	W0011-CORE: Access attempt to unimplemented RAM memory.
W0012_CORE_STACK_OVERFLOW_RESET	W0012-CORE: Device Resets on stack overflow.
W0013_CORE_STACK_UNDERFLOW_RESET	W0013-CORE: Device Resets on stack underflow.

© 2023 Microchip Technology Inc. and its subsidiaries

.....continued	
Message	**Meaning**
W0014_CORE_INVALID_OPCODE	W0014-CORE: Invalid opcode {0}
W0015_CORE_INVALID_ALT_WREG_SET	W0015-CORE: Invalid alternate register set specified in CTXTSTAT.CCTXI {0}
W0016_CORE_STACK_ERROR	W0016-CORE: Trap due to stack error, occurred from instruction at {0}.
W0017_CORE_ODD_RAMWORDACCESS	W0017-CORE: Trap due to misaligned data word access, occurred from instruction at {0}.
W0018_CORE_UNIMPLEMENTED_RAMACCESS	W0018-CORE: Trap due to unimplemented RAM memory access, occurred from instruction at {0}.
W0019_CORE_UNIMPLEMENTED_PROMACCESS	W0019-CORE: Trap due to unimplemented FLASH memory access, occurred from instruction at {0}.
W0020_CORE_ACCESS_NOTIN_X_SPACE	W0020-CORE: Trap due to memory access outside X data space, occurred from instruction at {0}.
W0021_CORE_ACCESS_NOTIN_Y_SPACE	W0021-CORE: Trap due to memory access outside Y data space, occurred from instruction at {0}.
W0022_CORE_XMODEND_LESS_XMODSRT	W0022-CORE: XMODEND is less than XMODSRT.
W0023_CORE_YMODEND_LESS_YMODSRT	W0023-CORE: YMODEND is less than YMODSRT.
W0024_CORE_BITREV_MOD_IS_ZERO	W0024-CORE: In bit-reversed mode and the modifier is 0.
W0025_CORE_HARD_TRAP	W0025-CORE: Reset due to Hard Trap from Low Priority Trap while processing Higher Priority Trap at {0}.
W0026_CORE_UNIMPLEMENTED_MEMORYACCESS	W0026-CORE: Trap due to unimplemented RAM or PSV memory access, occurred from instruction at {0}.
W0027_CORE_UNIMPLEMENTED_EDSACCESS	W0027-CORE: Trap due to Page 0 EDS memory access, occurred from instruction at {0}.
W0028_TBLRD_WORM_CONFIG_MEMORY	W0028-CORE: Simulator does not support WORM Configuration memory. Read WORM config memory in range 0x{0} to 0x{1} probably will not give you the same result as real device.
W0029_TBLRD_DEVICE_ID	W0029-CORE: Simulator does not support Device ID section. Read address 0x{0}, which is in the Device ID section, probably will not give you the same result as real device.
W0030_CORE_UNIMPLEMENTED_MEMORY_ACCESS	W0030-CORE: Instruction at {0} attempted unimplemented memory access ({1})
W0031_BSLIM_INSUFFICIENT_BOOT_SEGMENT	W0031-CORE: Boot segment should be at least 2 pages to support AIVT.
W0032_BSLIM_LIMITS_EXCEEDS_PROG_MEMORY	W0032-CORE: BSLIM value exceeds available program memory space.
W0033_CORE_UNPREDICTABLE_OPCODE	W0033-CORE: Unpredictable opcode at {0}
W0034_CORE_UNALIGNED_MEMORY_ACCESS	W0034-CORE: Unaligned memory access at {0}
W0035_CORE_UNIMPLEMENTED_RAMACCESS_NOTRAP	W0035-CORE: Unimplemented RAM memory access, occurred from instruction at {0}.
W0036_UNIMPLEMENTED_INSTRUCTION	W0036-CORE: Unimplemented instruction:
W0040_FPU_DIFF_CP10_CP11	W0040-FPU: At {0}, UNPREDICTABLE due to different values in CPACR.CP10 and CPACR.CP11.
W0041_FPU_ACCESS_DENIED	W0041-FPU: At {0}, attempted access FPU, but FPU access privilege is access denied.
W0042_FPU_PRIVILEGED_ACCESS_ONLY	W0042-FPU: At {0}, attempted access FPU in unprivileged mode, but FPU access privilege is privileged access only.
W0043_FPU_CP_RESERVED_VALUE	W0043-FPU: At {0}, attempted access FPU, but CPACR.CP10 has reserved value.
W0044_FPU_OUT_OF_RANGE	W0044-FPU: At {0}, but ARM Cortex M7 only supports S0-S31 and D0-D15.

© 2023 Microchip Technology Inc. and its subsidiaries

.....continued

Message	Meaning
W0051_INSTRUCTION_DIV_NOT_ENOUGH_REPEAT	W0051-INSTRUCTION: New lightning divide instruction requires REPEAT with an iteration count of 5, but your program only have {0}. The result may not be correct.
W0052_INSTRUCTION_DIV_TOO_MANY_REPEAT	W0052-INSTRUCTION: New lightning divide instruction only needs REPEAT with an iteration count of 5, but iteration count of {0} will also give the correct result.
W0053_INVALID_INTCON_VS_FIELD_VALUE	W0053-INTCON: VS Field value selected is reserved.
W0101_SIM_UPDATE_FAILED	W0101-SIM: Notified Update() method failed: {0}
W0102_SIM_PERIPH_MISSING	W0102-SIM: Peripheral Missing: {0}
W0103_SIM_PERIPH_FAILED	W0103-SIM: Peripheral Failed: {0}
W0104_SIM_FAILED_TO_INIT_TOOL	W0104-SIM: Failed to init tool
W0105_SIM_INVALID_FIELD	W0105-SIM: Invalid field value: {0}
W0106_SIM_PERIPH_PARTIAL_SUPPORT	W0106-SIM: This device only has partial support for {0} peripheral. {1}
W0107_SIM_NOT_SUPPORTED	W0107-SIM: {0} may not be supported because {1}.
W0108_SIM_RESERVED_SETTING	W0108-SIM: Reserved setting: {0}
W0109_SIM_PERIPHERAL_IN_DEVELOPMENT	W0109-SIM: {0} peripheral still in development - should not be used.
W0110_SIM_UNEXPECTED_EVENT	W0110-SIM: Something unexpected happened, tell Microchip about it: {0}
W0111_SIM_UNSUPPORTED_SELECTION	W0111-SIM: The selection is not supported: {0}
W0112_SIM_INVALID_OPERATION	W0112-SIM: The operation is not valid: {0}
W0113_SIM_WRITE_TO_PROTECTED_SFR	W0113-SIM: A write to a protected SFR was attempted: {0} {1}.
W0114_SIM_INVALID_KEY	W0114-SIM: Invalid KEY {0}. The correct PASSWD should be {1}.
W0115_SIM_FAILED_TO_PARSE_DEVICE_FILE	W0115-SIM: Failed to parse device file: {0}
W0116_SIM_STACK_OVERFLOW	W0116-SIM: Last push caused a stack overflow
W0117_SIM_STACK_UNDERFLOW	W0117-SIM: Last pop caused a stack underflow
W0118_SIM_INVALID_FIELD_VALUE	W0118-SIM: Invalid field value {0}
W0119_SIM_SAMPLING_RATE_VIOLATION	W0119-SIM: Sampling rate too high for current limit set in CTRLB.CURRLIMIT
W0201_ADC_NO_STIMULUS_FILE	W0201-ADC: No stimulus file attached to the ADC output buffer: {0}
W0202_ADC_GO_DONE_BIT	W0202-ADC: The Go/Done bit must not be set in the same cycle as the Enable bit.
W0203_ADC_MINIMUM_2_TAD	W0203-ADC: A Minimum of 2 TADs are required before another conversion should be started.
W0204_ADC_TAD_TOO_SMALL	W0204-ADC: Tad time is less than: {0}
W0205_ADC_UNEXPECTED_TRANSITION	W0205-ADC: Unexpected state transition (contact Microchip): {0}
W0206_ADC_SAMP_TIME_TOO_SHORT	W0206-ADC: Sample time too short: {0}
W0207_ADC_NO_PINS_SCANNED	W0207-ADC: No pins scanned.
W0208_ADC_UNSUPPORTED_CLOCK_SOURCE	W0208-ADC: Simulator does not support clock source, system clock used: {0}
W0209_ADC_ANALOG_CHANNEL_DIGITAL	W0209-ADC: Analog channel pin is configured as digital.
W0210_ADC_ANALOG_CHANNEL_OUTPUT	W0210-ADC: Analog channel pin is configured as output.
W0211_ADC_PIN_INVALID_CHANNEL	W0211-ADC: Selected channel is an invalid channel. The channel selected: {0}

© 2023 Microchip Technology Inc. and its subsidiaries

.....continued

Message	Meaning
W0212_ADC_BAND_GAP_NOT_SUPPORTED	W0212-ADC: Bandgap reference voltage not supported by simulator
W0213_ADC_RESERVED_SSRC	W0213-ADC: SSRC value not supported: {0}
W0214_ADC_POSITIVE_INPUT_DIGITAL	W0214-ADC: Positive input pin is configured as digital. {0}
W0215_ADC_POSITIVE_INPUT_OUTPUT	W0215-ADC: Positive input pin is configured as output. {0}
W0216_ADC_NEGATIVE_INPUT_DIGITAL	W0216-ADC: Negative input pin is configured as digital. {0}
W0217_ADC_NEGATIVE_INPUT_OUTPUT	W0217-ADC: Negative input pin is configured as output. {0}
W0218_ADC_REFERENCE_HIGH_DIGITAL	W0218-ADC: ADC reference high pin is configured as digital.
W0219_ADC_REFERENCE_HIGH_OUTPUT	W0219-ADC: ADC reference high pin is configured as output.
W0220_ADC_REFERENCE_LOW_DIGITAL	W0220-ADC: ADC reference low pin is configured as digital.
W0221_ADC_REFERENCE_LOW_OUTPUT	W0221-ADC: ADC reference low pin is configured as output.
W0222_ADC_OVERFLOW	W0222-ADC: ADC input voltage high. ADC output overflow.
W0223_ADC_UNDERFLOW	W0223-ADC: ADC input voltage low. ADC output underflow.
W0224_ADC_CTMU_NOT_SUPPORTED	W0224-ADC: CTMU not supported.
W0225_ADC_INVALID_CH0S	W0225-ADC: Invalid CH0S value.
W0226_ADC_VBAT_NOT_SUPPORTED	W0226-ADC: Vbat not supported.
W0227_ADC_INVALID_ADCS	W0227-ADC: Invalid ADCS value.
W0228_ADC_INVALID_ADCS	W0228-ADC: No high reference pin found.
W0229_ADC_INVALID_ADCS	W0229-ADC: No low reference pin found.
W0230_ADC_TRIGSEL_NOT_SUPPORTED	W0230-ADC: TRIGSEL values other than default not supported.
W0231_ADC_NOT_WARMED	W0231-ADC: Core({0}) enabled before warmed.
W0232_ADC_CALIBRATION_ABORTED	W0232-ADC: Core({0}) calibration aborted before completed.
W0233_ADC_CORE_POWERED_EARLY	W0233-ADC: Core({0}) xxxPWR bit set before ADON.
W0234_ADC_ALREADY_CALIBRATING	W0234-ADC: Core({0}) Already calibrating when another calibration request was made.
W0235_ADC_CAL_TYPE_CHANGED	W0235-ADC: Core({0}) Calibration type changed while already calibrating.
W0236_ADC_CAL_INVALIDATED	W0236-ADC: Core({0}) Calibration invalidated.
W0237_ADC_UNKNOWN_DATASHEET	W0237-ADC: ADC emulation does not support current datasheet {0} - contact Microchip.
W0238_ADC_INVALID_SFR_FIELD_VALUE	W0238-ADC: Invalid value for {0} ({1})
W0239_ADC_UNSUPPORTED_INPUT	W0239-ADC: ADC emulation does not support input from {0}
W0240_ADC_NOT_CALIBRATED	W0240-ADC: Conversion started when not calibrated {0}
W0241_ADC_FRACTIONAL_NOT_ALLOWED	W0241-ADC: Fractional format not allowed while using oversampling digital filter
W0242_ADC_BG_INT_BEFORE_PWR	W0242-ADC: REFCIE or REFERCIE enabled before ADC enabled - could cause spurious interrupt
W0243_ADC_INVALID_TAD	W0243-ADC: Invalid TAD {0} {1}
W0244_ADC_CONVERSION_ABORTED	W0244-ADC: Conversion aborted
W0245_ADC_BUFREGEN_NOT_ALLOWED	W0245-ADC: BUFREGEN has to be used with pins selected as input; using FIFO instead.
W0246_ADC_ACCUMULATION_BAD_RESSEL	W0246-ADC: > 1 samples selected in SAMPLENUM but RESSEL field not set for 16 bits output - accumulating 12 bits samples.
W0247_ADC_CONVERSION_BAD_RESSEL	W0247-ADC: 1 sample selected in SAMPLENUM but RESSEL field set for 16 bits output - taking a 12 bits sample.
W0248_ADC_WR_BEFORE_KEY_SEQ	W0248-ADC: WR bit set before key sequence completed

© 2023 Microchip Technology Inc. and its subsidiaries

.....continued	
Message	**Meaning**
W0400_PWM_PWM_FASTER_THAN_FOSC	W0400-PWM: PWM{0} clock faster than Fosc
W0600_WDT_2ND_WDT_MR_WRITE	w0600-WDT: WDT_MR written more than once at 0x{0}
W0601_WDT_EXPIRED	W0601-WDT: WDT expired at {0}
W0601_WDT_RESET_OUTSIDE_WINDOW	W0602-WDT: WDT cleared outside 0-WDD window at {0}
W0700_CLC_GENERAL_WARNING	W0700-CLC: Simulator does not simulate the propagation delay, so the output timing may not be accuracy and using the CLC output as its own input will create a infinite circular loop.
W0701_CLC_CLCOUT_AS_INPUT	W0701-CLC: Simulator does not support using the CLC output as its own input.
W0702_CLC_CIRCULAR_LOOP	W0702-CLC: Circular loop detected. This CLC output loopback as one of its inputs. Simulator CLC output may become incorrect.
W0800_ACC_INPUT_INVALID_CONFIG	W0800-ACC: Selected input pin configured incorrectly: {0}
W0801_ACC_INPUT_NOT_SUPPORTED	W0801-ACC: Input {0} not supported
W0802_ACC_INVERTED_WINDOW_LIMITS	W0802-ACC: Comparator pair {0} has upper limit below lower limit
W0803_ACC_MISMATCHED_POS_INPUTS	W0803-ACC: Comparator pair {0} has mismatched positive inputs
W0804_ACC_WINDOW_COMP_DISABLED	W0804-ACC: Comparator pair {0} has at least one of its comparators disabled
W0805_ACC_WINDOW_COMPS_MODES	W0805-ACC: Comparator pair {0} comparators have different COMPCTRL#.SINGLE values
W0806_ACC_FEATURE_NOT_SUPPORTED	W0806-ACC: Feature {0} is not supported in the simulator
W1201_DATAFLASH_MEM_OUTSIDE_RANGE	W1201-DATAFLASH: Attempt to {0} memory at {1} outside of dataflash memory range.
W1202_DATAFLASH_ERASE_WHILE_LOCKED	W1202-DATAFLASH: Attempt to erase dataflash memory while locked.
W1203_DATAFLASH_WRITE_WHILE_LOCKED	W1203-DATAFLASH: Attempt to write dataflash memory while locked.
W1401_DMA_PERIPH_NOT_AVAIL	W1401-DMA: The desired peripheral is not available: {0}.
W1402_DMA_INVALID_IRQ	W1402-DMA: The desired IRQ is not associated with a DMA peripheral: {0}.
W1403_DMA_INVALID_SFR	W1403-DMA: The peripheral address is not SFR expected with this IRQ: {0}.
W1404_DMA_INVALID_DMA_ADDR	W1404-DMA: The address is not a valid DMA address: {0}.
W1405_DMA_IRQ_DIR_MISMATCH	W1405-DMA: The selected DMA direction is not that expected for the selected IRQ: {0}.
W1600_PPS_INVALID_MAP	W1600-PPS: Invalid attempt to map pin ({0}) to pin ({1}).
W1601_PPS_INVALID_PIN_DESCRIPTION	W1601-PPS: Suspect database entry for PPS pin {0}. Contact Microchip.
W1800_PWM_TIMER_SELECTION_NOT_AVIALABLE	W1800-PWM: PWM timer selection {0} is not available or not supported.
W1801_PWM_TIMER_SELECTION_BAD_CLOCK_INPUT	W1801-PWM: TMR{0} clock input is not FOSC/4. It is required to have FOSC/4 as the clock input to TMR{0} for correct PWM operation.
W1802_PWM_TIMER_MISSING_PERSCALER_INFO	W1802-PWM: Simulator could not handle the PWM timer selection prescaler. The PWM output could be invalid.

© 2023 Microchip Technology Inc. and its subsidiaries

.....continued	
Message	**Meaning**
W2001_INPUTCAPTURE_TMR3_UNAVAILABLE	W2001-INPUTCAPTURE: TMR3 is not available for this part. Capture buffer will have a value of 0 for each TMR3 capture.
W2002_INPUTCAPTURE_CAPTURE_EMPTY	W2002-INPUTCAPTURE: The capture buffer was empty.
W2003_INPUTCAPTURE_SYNCSEL_NOT_AVIALABLE	W2003-INPUTCAPTURE: Source {0} is reserved or the source peripheral is not available.
W2004_INPUTCAPTURE_BAD_SYNC_SOURCE	W2004-INPUTCAPTURE: Source {0} should only be used as trigger sources
W2501_OUTPUTCOMPARE_SYNCSEL_NOT_AVIALABLE	W2501-OUTPUTCOMPARE: Source {0} is reserved or the source peripheral is not available.
W2502_OUTPUTCOMPARE_BAD_SYNC_SOURCE	W2502-OUTPUTCOMPARE: Source {0} should only be used as trigger sources.
W2503_OUTPUTCOMPARE_BAD_TRIGGER_SOURCE	W2503-OUTPUTCOMPARE: Source {0} should not be used as {1}trigger source.
W2700_MPU_ILLEGAL_DREGION	W2700-MPU: MPU_TYPE.DREGION was {0}. Only legal values are 8 and 16.
W2701_MPU_INVALID_REGION	W2701-MPU: MPU_RNR.REGION was {0}. It must be less than {1}.
W3000_LPM_READ_PROTECTION_SECTION	W3000-LPM: LPM executing at {0} is not allowed to read {1} with current Boot Lock bits setting.
W3010_SPM_WRITE_PROTECTION_SECTION	W3010-SPM: SPM is not allowed to write to {1} with current Boot Lock bits setting.
W6001_RTT_FORBIDDEN_RTPRES	W6001-RTT: Programming RTPRES to 1 or 2 is forbidden.
W6002_RTT_BAD_WRITING_ALMV	W6002-RTT: Try to write a new ALMV value when ALMIEN in RTT_MR is set.
W6003_RTT_BAD_WRITING_RTPRES	W6003-RTT: Try to write a new RTPRES value when RTTINCIEN bit is set.
W7001_SMT_CLK_SELECTION_NOT_SUPPORT	W7001-SMT: SMT partial support for this device. If {0} is not the clock selection for {1}, this clock selection is not support yet.
W7002_SMT_SIG_SELECTION_NOT_SUPPORT	W7002-SMT: SMT partial support for this device. If {0} is not the signal selection for {1}, this signal selection is not support yet.
W7003_SMT_WIN_SELECTION_NOT_SUPPORT	W7003-SMT: SMT partial support for this device. If {0} is not the window selection for {1}, this window selection is not support yet.
W8001_OSC_INVALID_CLOCK_SOURCE	W8001-OSC: Clock source selection invalid; OSC is not enabled.
W8002_OSC_RESERVED_FEXTOSC	W8002-OSC: Reserved External Oscillator mode Selection. Oscillator not enabled.
W9001_TMR_GATE_AND_EXTCLOCK_ENABLED	W9001-TMR: Gate control cannot be enabled at the same time as external clock control.
W9002_TMR_NO_PIN_AVAILABLE	W9002-TMR: Gate or external clock mode is enabled, but no pin is available.
W9003_TMR_INVALID_CLOCK_SOURCE	W9003-TMR: Clock source selection invalid; timer will not increment.
W9201_UART_TX_OVERFLOW	W9201-UART: Write attempted to a full FIFO buffer, data lost.
W9202_UART_TX_CAPTUREFILE	W9202-UART: Error opening Tx UART capture file: {0}.
W9203_UART_TX_INVALIDINTERRUPTMODE	W9203-UART: Invalid Tx UART interrupt mode set: {0}.
W9204_UART_RX_EMPTY_QUEUE	W9204-UART: Rx UART FIFO queue empty.

© 2023 Microchip Technology Inc. and its subsidiaries

.....continued	
Message	**Meaning**
W9205_UART_TX_BADFILE	W9205-UART: Could not write file: {0}.
W9206_UART_RESERVED_MODE	W9206-UART: Selected Mode is either reserved or not yet supported: {0}.
W9207_UART_UNABLETOCLOSE_FILE	W9207-UART: Unable to close file: {0}.
W9401_CVREF_INVALIDSOURCESELECTION	W9401-CVREF: Positive Source Selection option(11) is reserved.Do not use.Positive Source is floating.
W9402_CVREF_INPUT_OUTPUTPINCONFLICT	W9402-CVREF: Potential Scenario where output pin is looped back as input.
W9601_COMP_FVR_SOURCE_UNAVAILABLE	W9601-COMP:FVR Voltage source not available as input to this Comparator.
W9602_COMP_DAC_SOURCE_UNAVAILABLE	W9602-COMP:DAC Voltage Source Peripheral not yet implemented to act as input to Comparator.
W9603_COMP_CVREF_SOURCE_UNAVAILABLE	W9603-COMP:CVREF Voltage Source Peripheral not available as input to this Comparator.
W9604_COMP_SLOPE_SOURCE_UNAVAILABLE	W9604-COMP:Slope Generator Voltage Source Peripheral not available as input to this Comparator.
W9605_COMP_PRG_SOURCE_UNAVAILABLE	W9605-COMP:Programmable Ramp Generator Voltage Source Peripheral not available as input to this Comparator.
W9607_COMP_DGTL_FLTR_OPTION_UNAVAILABLE	W9607-COMP:Digital Filter selection reserved,SYSCLK will be used instead.
W9609_COMP_DGTL_FLTR_CLK_UNAVAILABLE	W9609-COMP:REFCLK3 not implemented,SYSCLK will be used instead.
W9801_FVR_INVALID_MODE_SELECTION	W9801-FVR:FVR Voltage Selection bits(11) not Valid.Reserved, Do not use.
W9901_RTSP_INVALID_OPERATION_SELECTION	W9901-RTSP:Reserved NVMOP Operation Selection bits.
W9902_RTSP_FLASH_PROGRAM_WRITE_PROTECTED	W9902-RTSP:Flash Program Memory Location appears to be Write Protected using CodeGuard.
W10001_RESERVED_IRQ_HANDLER_INVOKED	W10001-IRQ:RESERVED IRQ is set for handling.Will be ignored.
W10002_UNSUPPORTED_CLK_SOURCE	W10002-SYSTICK:External Clock selection is not yet supported.
W10101_UNSUPPORTED_CHANNEL_MODE	W10101-UART:Channel Mode Selected is not yet supported.
W10102_UNSUPPORTED_CLK_SOURCE	W10102-UART:PMC-Programmable Clock source is not yet supported.
W10103_UNSUPPORTED_RECEIVER_FILTER	W10103-UART:Receiver digital filter is not yet supported.PPS_MISSING_SFR
W10301_NO_PORT_PINS_FOUND	W10301-PORT:No port pins were found in device datafile
W10500_UNSUPPORTED_SOURCE	W10500-EVSYS: Channel source selection unsupported by simulator (channel {0})

3.3.6 Programming Commands

To display information about the programming commands available in the MDB, type `help programming`. The following table provides additional information for these commands.

Important: The `set` command, including the tool property options, must be executed before the `Hwtool` command is issued, otherwise the changes to the tool properties will be ignored.

© 2023 Microchip Technology Inc. and its subsidiaries

Table 3-10. Programming Commands

Command	Description
Dump	Writes the device memory to a hex file. **Command format:** `Dump [-m] filename` The `m` is an optional argument that specifies which memories to write to the hex file. It can be any combination of the following: • `p` - Program Memory (Flash) • `e` - EE Data • `c` - Configuration Bits • `u` - User ID memory • `b` - Boot Memory • `f` - Flash Data The `filename` is the full path and name to the hex file.
Execute	Execute the specified function from the loaded script. **Command format:** `Execute fuctionname`
LoadScript	Load the target script. **Command format:** `LoadScript scriptname`
Program	Programs device memory with the image specified by the file. **Note:** If the path or filename has spaces in it, you must use the quotation marks. If there are no spaces in the path of filename, the quotation marks are not needed, as shown below. **Command format:** `Program executableImageFile`
Upload	Uploads the executable image to MDB memory. The source of the instructions to be loaded is the contents of the memory of an attached PIC® device through the programmer or debugger. **Command format:** `Upload`

3.3.7 Running Commands

To display information about the running commands available in the MDB, type help running. The following table provides additional information for these commands.

Command	Description
Continue	Resumes program being debugged, after breakpoint. **Command format:** `Continue`
Halt	Stops the debugged program. **Command format:** `Halt`
Next	Step program, proceeding through subroutine calls. Like the “step” command as long as subroutine calls do not happen; when they do, the call is treated as one instruction. **Command format:** `Next`
Run	Start the debugged program. **Command format:** `Run`

© 2023 Microchip Technology Inc. and its subsidiaries

.....continued

Command	Description
Step	Step program until it reaches a different source line. The step command only enters a function if there is a line number information for the function. **Command format:** `Step`
Stepi	Execute one machine instruction, then stop and return to the debugger. The optional argument count is a repeat count. **Command format:** `Stepi [count]`

3.3.8 Stack Commands

To display information about the stack commands available in MDB, type help backtrace. The following table provides additional information for these commands.

Table 3-11. Stack Commands

Command	Description
Backtrace	Print a backtrace of the entire stack, one line per frame for all frames in the stack. **Command format:** `Backtrace [full] [<n, -n>]` • `full` – prints the values of local variables • `n` – prints the innermost n frames • `-n` – prints the outermost n frames

© 2023 Microchip Technology Inc. and its subsidiaries

4. Revision History

4.1 Revision A (November 2012)

Initial release of this document.

4.2 Revision B (April 2013)

- added note in Invoking the MDB section
- added Tool Property Name Options for the Set command
- added Simulator Options for the Set command
- added `-p` option
- added note on running multiple tools
- removed example of using commands to debug a project
- added section on creating a printable log file

4.3 Revision C (March 2014)

- relocated Revision History from Preface to it's own appendix.
- added a Document Layout section to the Preface.
- added new section "Using Multiple Instances of the MDB."
- moved reference tables to "MDB Reference."
- added notes about case-sensitivity for commands in "MDB Reference."
- added tool column to table "Tool-Property-Name Options Used with the Set Command."
- added new table: "Simulator Options Used With the Set Command."

4.4 Revision D (February 2017)

- added more documents to the Recommended Reading section in Preface.
- revised the title to "How to Use Microchip Debugger," revised the description of the MDB and added a note to the revised Programming a Production Image for Testing Purposes section.
- revised section "Getting Started" to add information on Command Line Parameters.
- added information in the Help Commands section.
- revised multiple tables in the MDB Reference chapter.

4.5 Revision E (October 2018)

- added information for the MPLAB PICkit 4 and MPLAB Snap In-Circuit Debuggers in "Introduction."
- renamed, reorganized and added information in "Installation and Documentation" and "Getting Started".
- updated paths in table "Paths to the MDB by Operating System."
- updated tools in table "Device and tool Commands" and table "Tool-Property-Name Options Used with the Set Command."

4.6 Revision F (February 2019)

- added SK (Starter Kits PKOB) and EDBG (embedded debugger) to list of MDB supported tools in "Introduction."
- added two EDBG options to a table "Tool-Property-Name Options Used with the Set Command."
- added EDBG tool to applicable options in table "Tool-Property-Name Options Used with the Set Command."

© 2023 Microchip Technology Inc. and its subsidiaries

4.7 Revision G (June 2020)

- Entire document has been reformatted and renumbered.
- Updated references to online help to WebHelp.
- Updated paths to documentation locations.

4.8 Revision H (December 2021)

- Updates Windows version to 10.
- Updated supported hardware tools list to include MPLAB ICE 4 in Table 3-4.
- Added new section for 3.3.4. Network Tool Commands.
- Added new commands for support of the XC code coverage and the MPLAB ICE 4 in Table 3-7.
- Updated Table 3-8.
- Added a new section and table for 3.3.5.3. List of Error and Warning Messages.
- Removed references to .cof files throughout the document. All Microchip compilers/assemblers produce .elf by default.

4.9 Revision I (May 2023)

- Added the following new commands to List of Other Commands in Section 3.3.5.
  - Pack
  - Group

Revision J (August 2023)

- Added the Display command to List of Other Commands in Section 3.3.5.

4.9.1 Revision K (October 2023)

- Added fix to the examine command to **List of Commands within Classes** > **Data Commands.**

© 2023 Microchip Technology Inc. and its subsidiaries

The Microchip Website

Microchip provides online support via our website at www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip design partner program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Product Change Notification Service

Microchip's product change notification service helps keep customers current on Microchip products. Subscribers will receive email notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, go to www.microchip.com/pcn and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

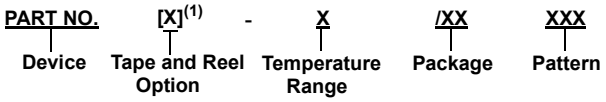
Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: www.microchip.com/support

© 2023 Microchip Technology Inc. and its subsidiaries

Product Identification System

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.



Device:	PIC16F18313, PIC16LF18313, PIC16F18323, PIC16LF18323	
Tape and Reel Option:	Blank	= Standard packaging (tube or tray)
	T	= Tape and Reel(1)
Temperature Range:	I	= -40°C to +85°C (Industrial)
	E	= -40°C to +125°C (Extended)
Package:(2)	JQ	= UQFN
	P	= PDIP
	ST	= TSSOP
	SL	= SOIC-14
	SN	= SOIC-8
	RF	= UDFN
Pattern:	QTP, SQTP, Code or Special Requirements (blank otherwise)	

Examples:

- PIC16LF18313- I/P Industrial temperature, PDIP package
- PIC16F18313- E/SS Extended temperature, SSOP package

Notes:

1. Tape and Reel identifier only appears in the catalog part number description. This identifier is used for ordering purposes and is not printed on the device package. Check with your Microchip Sales Office for package availability with the Tape and Reel option.
2. Small form-factor packaging options may be available. Please check www.microchip.com/packaging for small-form factor package availability, or contact your local Sales Office.

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip products:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip product is strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is “unbreakable”. Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.

Legal Notice

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information

© 2023 Microchip Technology Inc. and its subsidiaries

in any other manner violates these terms. Information regarding device applications is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. Contact your local Microchip sales office for additional support or, obtain additional support at www.microchip.com/en-us/support/design-help/client-support-services.

THIS INFORMATION IS PROVIDED BY MICROCHIP "AS IS". MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION.

Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, Adaptec, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, CryptoMemory, CryptoRF, dsPIC, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Kleer, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AgileSwitch, ClockWorks, The Embedded Control Solutions Company, EtherSynch, Flashtec, Hyper Speed Control, HyperLight Load, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet-Wire, SmartFusion, SyncWorld, TimeCesium, TimeHub, TimePictra, TimeProvider, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, Augmented Switching, BlueSky, BodyCom, Clockstudio, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, Espresso T1S, EtherGREEN, EyeOpen, GridTime, IdealBridge, IGaT, In-Circuit Serial Programming, ICSP, INICnet, Intelligent Paralleling, IntelliMOS, Inter-Chip Connectivity, JitterBlocker, Knob-on-Display, MarginLink, maxCrypto, maxView, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, mSiC, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, Power MOS IV, Power MOS 7, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, RTAX, RTG4, SAM-ICE, Serial Quad I/O, simpleMAP, SimpliPHY, SmartBuffer, SmartHLS, SMART-I.S., storClad, SQI, SuperSwitcher, SuperSwitcher II, Switchtec, SynchroPHY, Total Endurance, Trusted Time, TSHARC, Turing, USBCheck, VariSense, VectorBlox, VeriPHY, ViewSpan, WiperLock, XpressConnect, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

The Adaptec logo, Frequency on Demand, Silicon Storage Technology, and Symmcom are registered trademarks of Microchip Technology Inc. in other countries.

© 2023 Microchip Technology Inc. and its subsidiaries

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2023, Microchip Technology Incorporated and its subsidiaries. All Rights Reserved.

ISBN: 978-1-6683-3360-0

Quality Management System

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

© 2023 Microchip Technology Inc. and its subsidiaries

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office	**Australia - Sydney**	**India - Bangalore**	**Austria - Wels**
2355 West Chandler Blvd.	Tel: 61-2-9868-6733	Tel: 91-80-3090-4444	Tel: 43-7242-2244-39
Chandler, AZ 85224-6199	**China - Beijing**	**India - New Delhi**	Fax: 43-7242-2244-393
Tel: 480-792-7200	Tel: 86-10-8569-7000	Tel: 91-11-4160-8631	**Denmark - Copenhagen**
Fax: 480-792-7277	**China - Chengdu**	**India - Pune**	Tel: 45-4485-5910
Technical Support:	Tel: 86-28-8665-5511	Tel: 91-20-4121-0141	Fax: 45-4485-2829
www.microchip.com/support	**China - Chongqing**	**Japan - Osaka**	**Finland - Espoo**
Web Address:	Tel: 86-23-8980-9588	Tel: 81-6-6152-7160	Tel: 358-9-4520-820
www.microchip.com	**China - Dongguan**	**Japan - Tokyo**	**France - Paris**
Atlanta	Tel: 86-769-8702-9880	Tel: 81-3-6880- 3770	Tel: 33-1-69-53-63-20
Duluth, GA	**China - Guangzhou**	**Korea - Daegu**	Fax: 33-1-69-30-90-79
Tel: 678-957-9614	Tel: 86-20-8755-8029	Tel: 82-53-744-4301	**Germany - Garching**
Fax: 678-957-1455	**China - Hangzhou**	**Korea - Seoul**	Tel: 49-8931-9700
Austin, TX	Tel: 86-571-8792-8115	Tel: 82-2-554-7200	**Germany - Haan**
Tel: 512-257-3370	**China - Hong Kong SAR**	**Malaysia - Kuala Lumpur**	Tel: 49-2129-3766400
Boston	Tel: 852-2943-5100	Tel: 60-3-7651-7906	**Germany - Heilbronn**
Westborough, MA	**China - Nanjing**	**Malaysia - Penang**	Tel: 49-7131-72400
Tel: 774-760-0087	Tel: 86-25-8473-2460	Tel: 60-4-227-8870	**Germany - Karlsruhe**
Fax: 774-760-0088	**China - Qingdao**	**Philippines - Manila**	Tel: 49-721-625370
Chicago	Tel: 86-532-8502-7355	Tel: 63-2-634-9065	**Germany - Munich**
Itasca, IL	**China - Shanghai**	**Singapore**	Tel: 49-89-627-144-0
Tel: 630-285-0071	Tel: 86-21-3326-8000	Tel: 65-6334-8870	Fax: 49-89-627-144-44
Fax: 630-285-0075	**China - Shenyang**	**Taiwan - Hsin Chu**	**Germany - Rosenheim**
Dallas	Tel: 86-24-2334-2829	Tel: 886-3-577-8366	Tel: 49-8031-354-560
Addison, TX	**China - Shenzhen**	**Taiwan - Kaohsiung**	**Israel - Ra'anana**
Tel: 972-818-7423	Tel: 86-755-8864-2200	Tel: 886-7-213-7830	Tel: 972-9-744-7705
Fax: 972-818-2924	**China - Suzhou**	**Taiwan - Taipei**	**Italy - Milan**
Detroit	Tel: 86-186-6233-1526	Tel: 886-2-2508-8600	Tel: 39-0331-742611
Novi, MI	**China - Wuhan**	**Thailand - Bangkok**	Fax: 39-0331-466781
Tel: 248-848-4000	Tel: 86-27-5980-5300	Tel: 66-2-694-1351	**Italy - Padova**
Houston, TX	**China - Xian**	**Vietnam - Ho Chi Minh**	Tel: 39-049-7625286
Tel: 281-894-5983	Tel: 86-29-8833-7252	Tel: 84-28-5448-2100	**Netherlands - Drunen**
Indianapolis	**China - Xiamen**		Tel: 31-416-690399
Noblesville, IN	Tel: 86-592-2388138		Fax: 31-416-690340
Tel: 317-773-8323	**China - Zhuhai**		**Norway - Trondheim**
Fax: 317-773-5453	Tel: 86-756-3210040		Tel: 47-72884388
Tel: 317-536-2380			**Poland - Warsaw**
Los Angeles			Tel: 48-22-3325737
Mission Viejo, CA			**Romania - Bucharest**
Tel: 949-462-9523			Tel: 40-21-407-87-50
Fax: 949-462-9608			**Spain - Madrid**
Tel: 951-273-7800			Tel: 34-91-708-08-90
Raleigh, NC			Fax: 34-91-708-08-91
Tel: 919-844-7510			**Sweden - Gothenberg**
New York, NY			Tel: 46-31-704-60-40
Tel: 631-435-6000			**Sweden - Stockholm**
San Jose, CA			Tel: 46-8-5090-4654
Tel: 408-735-9110			**UK - Wokingham**
Tel: 408-436-4270			Tel: 44-118-921-5800
Canada - Toronto			Fax: 44-118-921-5820
Tel: 905-695-1980			
Fax: 905-695-2078			

© 2023 Microchip Technology Inc. and its subsidiaries