Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates.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. 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 ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. 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.

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

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable.”

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Trademarks

The Microchip name and logo, the Microchip logo, dsPIC, FlashFlex, KEELoQ, KEELoQ logo, MPLAB, PIC, PICmicro, PICSTART, PIC32 logo, rPIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Hampshire, HI-TECH C, Linear Active Thermistor, MTP, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

Analog-for-the-Digital Age, Application Maestro, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, Hi-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MIWI, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Omniscient Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICkit, PICtail, REAL ICE, rLAB, Select Mode, SQI, Serial Quad I/O, Total Endurance, TSHARC, UniWinDriver, WiperLock, ZENA and Z-Scale 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.

GestiC and ULPP are registered trademarks 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.

© 2012, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

Printed on recycled paper.
ISBN: 978-1-62076-673-6

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV

ISO/TS 16949

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company’s quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KeElOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip’s quality system for the design and manufacture of development systems is ISO 9001:2000 certified.
Table of Contents

Preface ....................................................................................................................... 5
How to Use MDB ..................................................................................................... 7
Index ....................................................................................................................... 17
Worldwide Sales and Service ............................................................................... 18
Preface

NOTICE TO CUSTOMERS

All documentation becomes dated, and this manual is no exception. Microchip 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 web site
(www.microchip.com) to obtain the latest documentation available.

Documents are identified with a “DS” number. This number is located on the bottom of each
page, in front of the page number. The numbering convention for the DS number is
“DSXXXXXA”, where “XXXXX” is the document number and “A” is the revision level of the
document.

For the most up-to-date information on development tools, see the MPLAB® X IDE help. Select
the Help menu, and then Topics to open a list of available help files.

INTRODUCTION

This chapter contains general information that will be helpful to know before using the
MDB. Items that are discussed include:
• Conventions Used in This Guide
• Recommended Reading

CONVENTIONS USED IN THIS GUIDE

The following conventions may appear in this documentation:

TABLE 1: DOCUMENTATION CONVENTIONS

<table>
<thead>
<tr>
<th>Description</th>
<th>Represents</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arial font:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Italic</td>
<td>Referenced books</td>
<td>MPLAB® IDE User’s Guide</td>
</tr>
<tr>
<td></td>
<td>Emphasized text</td>
<td>…is the only compiler…</td>
</tr>
<tr>
<td>Initial caps</td>
<td>A window</td>
<td>the Output window</td>
</tr>
<tr>
<td></td>
<td>A dialog</td>
<td>the Settings dialog</td>
</tr>
<tr>
<td></td>
<td>A menu selection</td>
<td>select Enable Programmer</td>
</tr>
<tr>
<td>Quotes</td>
<td>A field name in a window or dialog</td>
<td>“Save project before build”</td>
</tr>
<tr>
<td>Underlined, italic text with</td>
<td>A menu path</td>
<td>File&gt;Save</td>
</tr>
<tr>
<td>right angle bracket</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bold</td>
<td>A dialog button</td>
<td>Click OK</td>
</tr>
<tr>
<td></td>
<td>A tab</td>
<td>Click the Power tab</td>
</tr>
<tr>
<td>Text in angle brackets &lt; &gt;</td>
<td>A key on the keyboard</td>
<td>Press &lt;Enter&gt;, &lt;F1&gt;</td>
</tr>
</tbody>
</table>

Courier font:
TABLE 1: DOCUMENTATION CONVENTIONS (CONTINUED)

<table>
<thead>
<tr>
<th>Description</th>
<th>Represents</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plain</td>
<td>Sample source code</td>
<td>#define START</td>
</tr>
<tr>
<td>Filenames</td>
<td>autoexec.bat</td>
<td>c:\mcc18\h</td>
</tr>
<tr>
<td>File paths</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Keywords</td>
<td>_asm, _endasm, static</td>
<td></td>
</tr>
<tr>
<td>Command-line options</td>
<td>-Opa+, -Opa-</td>
<td></td>
</tr>
<tr>
<td>Bit values</td>
<td>0, 1</td>
<td></td>
</tr>
<tr>
<td>Constants</td>
<td>$\text{0xFF, 'A'}$</td>
<td></td>
</tr>
<tr>
<td>Italic</td>
<td>A variable argument</td>
<td>file.o, where file can be any valid filename</td>
</tr>
<tr>
<td>Square brackets [ ]</td>
<td>Optional arguments</td>
<td>mpasmwin [options]</td>
</tr>
<tr>
<td>Curly brackets and pipe character: {}</td>
<td>Choice of mutually exclusive arguments; an OR selection</td>
<td>errorlevel {0</td>
</tr>
<tr>
<td>Ellipses...</td>
<td>Replaces repeated text</td>
<td>var_name [, var_name...]</td>
</tr>
<tr>
<td></td>
<td>Represents code supplied by user</td>
<td>void main (void) { ... }</td>
</tr>
</tbody>
</table>

RECOMMENDED READING

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

Release Notes for MDB

For the latest information on using the MDB, read the release notes under “Release Notes and Support Documentation” on the MPLAB X IDE Start page. The release notes contain update information and known issues that may not be included in this user’s guide.

Debugger Design Advisory (DS51764)

A small document on guidelines and implementation considerations to ensure proper interfacing to the various development tools.

Processor Extension Pak and Header Specification (DS51292)

This booklet describes how to install and use Processor Extension Packs (PEPs) and related debug headers 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.
INTRODUCTION

MDB is the Microchip Debugger. It facilitates programming and debugging devices through the Command Prompt interface, instead of using the Microchip MPLAB® X IDE (Integrated Development Environment). MDB was designed for engineers who prefer to use the Command Prompt.

GETTING STARTED

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

Generate a cof/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 a cof/elf file is generated. The cof/elf file is a linked executable file that contains symbolic debugging information.

INVOKING THE MDB

Use the Command Prompt to invoke MDB. On Windows 7, the Command Prompt must be opened in Administrator mode: Start>All Programs>Accessories>Command Prompt, right click and select “Run as Administrator.” This opens the Administrator: Command Prompt.

The path to the MDB may be vary, depending on where the MPLAB X IDE was installed, and which operating system is installed.

In Windows 32 bit:

c:\>cd "c:\Program Files\Microchip\MPLABX\mplab_ide\bin"
c:\Program Files\Microchip\MPLABX\mplab_ide\bin>mdb.bat

In Windows 64 bit:

C:\>cd "c:\Program Files (x86)\Microchip\MPLABX\mplab_ide\bin"
c:\Program Files (x86)\Microchip\MPLABX\mplab_ide\bin>mdb.bat

In Linux:

$ cd /opt/microchip/mplabx/mplab_ide/bin
$ ./mdb.sh

In Mac OSX:

$ cd
/Applications/microchip/mplabx/mplab_ide.app/Contents/Resources/mplab_ide/bin
$ ./mdb.sh
You can run a test using either of these 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 “Entering Commands Method”.

Running a command file method cannot be used after invoking the MDB. It is included as a parameter in the command line when invoking the MDB. See “Running a Command File Method”.

ENTERING COMMANDS METHOD

Entering commands is a step-by-step method to run a test with MDB. The following sections describe:

- Classes of Commands
- List of Commands
- Programming a Device
- Debugging a Device

Classes of Commands

Once the MDB is running, you can start entering commands. Please note that the MDB commands are not case-sensitive. Type help for a list of classes of commands in MDB.

<table>
<thead>
<tr>
<th>TABLE 1-1: MDB CLASSES OF COMMANDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>breakpoints</td>
</tr>
<tr>
<td>data</td>
</tr>
<tr>
<td>deviceandtool</td>
</tr>
<tr>
<td>others</td>
</tr>
<tr>
<td>programming</td>
</tr>
<tr>
<td>running</td>
</tr>
<tr>
<td>stack</td>
</tr>
</tbody>
</table>
List of Commands

For a list of all commands within a particular class, type `help` followed by the class name. See the following tables for information about each list of commands.

For documentation on a particular command, type `help` followed by the command name.

### TABLE 1-2: BREAKPOINTS - LIST OF COMMANDS

<table>
<thead>
<tr>
<th>Break</th>
<th>Set a breakpoint at the specified source line number: break filename:linenumber [passCount]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Set a breakpoint at an absolute address: break *address [passCount]</td>
</tr>
<tr>
<td></td>
<td>• address - The address of the data memory to be watch. Use command <code>print /a</code> to get a variable address.</td>
</tr>
<tr>
<td></td>
<td>• passCount - The parameter is optional. The number of times the breakon condition is met before the program breaks.</td>
</tr>
<tr>
<td></td>
<td>Example: break newmain.c:16</td>
</tr>
<tr>
<td></td>
<td>MDB assigns a breakpoint number and returns: Breakpoint 0 at 0x9d0000cc: file newmain.c, line 16.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Watch</th>
<th>Set a data breakpoint at the specified memory address: Watch address breakonType[:value] [passCount]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• address - The address of the data memory to be watched. Use command <code>print /a</code> to get a variable address.</td>
</tr>
<tr>
<td></td>
<td>• BreakonType:</td>
</tr>
<tr>
<td></td>
<td>R -- Read.</td>
</tr>
<tr>
<td></td>
<td>W -- Write.</td>
</tr>
<tr>
<td></td>
<td>RW -- Read or Write.</td>
</tr>
<tr>
<td></td>
<td>• value - The parameter is optional. If it is specified, the program will break only when the value held in the data memory matches the specified value.</td>
</tr>
<tr>
<td></td>
<td>• passCount - The parameter is optional. The number of times the breakon condition is met before the program breaks.</td>
</tr>
<tr>
<td></td>
<td>Examples: watch 0xfffff W</td>
</tr>
<tr>
<td></td>
<td>MDB assigns a watchpoint number and returns: Watchpoint 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Delete</th>
<th>Delete a breakpoint - If no argument is specified, delete all breakpoints. You can abbreviate this command as d. Delete [breakpoint number]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• breakpoint number - Generated by MDB for break and/or watch command.</td>
</tr>
<tr>
<td></td>
<td>Example: Delete 1</td>
</tr>
</tbody>
</table>

| Halt        | Stop the debugged program.                                                                                   |
### TABLE 1-3: DATA - LIST OF COMMANDS

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Print</strong></td>
<td>Print a variable with optional formatting. <code>&lt;print [/f] [/datasize:value] variable&gt;</code></td>
</tr>
<tr>
<td></td>
<td>• <code>/f</code> - 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.</td>
</tr>
<tr>
<td></td>
<td>• <code>/datasize:value</code> - Optional data size. Variable in assembly code might not have data size information. User can specify the data size if the .cof or .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.</td>
</tr>
<tr>
<td><strong>Stim</strong></td>
<td>Specify a simulator SCL stimulus file to use. 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.) <code>&lt;Stim &quot;[path to file]&quot;</code></td>
</tr>
<tr>
<td><strong>x</strong></td>
<td>Examine memory. You can use the command <code>x</code> (for examine) to examine memory in any of several formats, independently of your program’s data types. <code>&lt;x [/tnfu] [addr]&gt;</code></td>
</tr>
<tr>
<td></td>
<td>• <code>t</code> - the type of memory. 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. e - EE Data memory.</td>
</tr>
<tr>
<td></td>
<td>• <code>n</code> - the repeat count. The repeat count is a decimal integer; the default is 1. It specifies how much memory (counting by units u) to display.</td>
</tr>
<tr>
<td></td>
<td>• <code>f</code> - the display format. The display format is one of the formats used by print (x, d, o, f, s), and in addition “i” (for machine instructions). The default is ‘x’ (hexadecimal) initially. The default changes each time you use x.</td>
</tr>
<tr>
<td></td>
<td>• <code>u</code> - 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: b - Bytes. h - Halfwords (two bytes). w - Words (four bytes). This is the initial default.</td>
</tr>
<tr>
<td></td>
<td>• <code>addr</code> - starting display address. <code>addr</code> is the address where you want MDB to begin displaying memory. The expression need not have a pointer value (though it may); it is always interpreted as an integer address of a byte of memory. The default for <code>addr</code> is usually just after the last address examined, but 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).</td>
</tr>
</tbody>
</table>
### TABLE 1-4: DEVICE AND TOOL - LIST OF COMMANDS

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Device</strong></td>
<td>Set the name of the target device.</td>
</tr>
<tr>
<td>Device devicename</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Device PIC32MX795F512L</td>
</tr>
<tr>
<td><strong>Hwtool</strong></td>
<td>Set the debug tool.</td>
</tr>
<tr>
<td>Hardware toolname</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>Following are the supported tool names (not case-sensitive):</td>
</tr>
<tr>
<td></td>
<td>• ICD3 - MPLAB ICD 3 In-Circuit Debugger</td>
</tr>
<tr>
<td></td>
<td>• RealICE - MPLAB REALICE In-Circuit Emulator</td>
</tr>
<tr>
<td></td>
<td>• PICkit3 - PICkit 3 In-Circuit Debugger</td>
</tr>
<tr>
<td></td>
<td>• SIM - Simulator</td>
</tr>
<tr>
<td></td>
<td>• PM3 - MPLAB PM3 Programmer</td>
</tr>
<tr>
<td></td>
<td>• LicensedDebugger - third party debugger</td>
</tr>
<tr>
<td></td>
<td>• LicensedProgrammer - third party programmer</td>
</tr>
</tbody>
</table>

### TABLE 1-5: OTHERS - LIST OF COMMANDS

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Help</strong></td>
<td>Print list of commands.</td>
</tr>
<tr>
<td><strong>Quit</strong></td>
<td>Quit - exits the debugger.</td>
</tr>
<tr>
<td><strong>Set</strong></td>
<td>Set command. The tool property name and value are from the project properties</td>
</tr>
<tr>
<td></td>
<td>selected when creating the project in MPLAB X IDE.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>Set tool-property-name tool-property-value</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>Set programoptions.eraseb4program true</td>
</tr>
<tr>
<td><strong>Sleep</strong></td>
<td>Makes the current script processor sleep until specified milliseconds have elapsed.</td>
</tr>
<tr>
<td></td>
<td>Sleep milliseconds</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>Sleep 10</td>
</tr>
<tr>
<td><strong>Wait</strong></td>
<td>Wait command makes the current script processor wait until the debugger halts</td>
</tr>
<tr>
<td></td>
<td>before processing the next command.</td>
</tr>
<tr>
<td></td>
<td>Wait</td>
</tr>
<tr>
<td></td>
<td>Wait Milliseconds makes the processor process the next command if the debugger</td>
</tr>
<tr>
<td></td>
<td>does not halt and milliseconds have elapsed.</td>
</tr>
<tr>
<td></td>
<td>Wait [milliseconds]</td>
</tr>
</tbody>
</table>

### TABLE 1-6: PROGRAMMING - LIST OF COMMANDS

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Program</strong></td>
<td>Programs device memory with the image specified by the file. (Note, if the path or</td>
</tr>
<tr>
<td></td>
<td>filename has spaces in it, you must use the quotation marks. If there are no spaces</td>
</tr>
<tr>
<td></td>
<td>in the path of filename, the quotation marks are not needed, as shown below.)</td>
</tr>
<tr>
<td>Program executableImageFile</td>
<td></td>
</tr>
<tr>
<td><strong>Upload</strong></td>
<td>Uploads the executable image to MDB memory.</td>
</tr>
<tr>
<td>Upload</td>
<td></td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>Continue</td>
<td>Resume program being debugged, after breakpoint.</td>
</tr>
<tr>
<td>Halt</td>
<td>Stops the debugged program.</td>
</tr>
<tr>
<td>Next</td>
<td>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.</td>
</tr>
<tr>
<td>Run</td>
<td>Start the debugged program.</td>
</tr>
<tr>
<td>Step</td>
<td>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.</td>
</tr>
</tbody>
</table>

**TABLE 1-8: STACK - LIST OF COMMANDS**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backtrace</td>
<td>Print a backtrace of the entire stack; one line per frame for all frames in the stack.</td>
</tr>
<tr>
<td>Backtrace [full] [&lt;n, -n&gt;]</td>
<td></td>
</tr>
<tr>
<td>• full - prints the values of local variables</td>
<td></td>
</tr>
<tr>
<td>• n - prints the innermost n frames</td>
<td></td>
</tr>
<tr>
<td>• -n - prints the outermost n frames</td>
<td></td>
</tr>
</tbody>
</table>
Programming a Device

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

1. Select the device by entering the command:
   Device [device name]
   For example: Device PIC18F66K22

2. Select the hardware tool. Currently, MDB only supports MPLAB ICD 3, MPLAB PM3, MPLAB REAL ICE, PICkit 3 and Simulator. To verify the supported tools, type:
   Help Hwtool
   The MPLAB ICD 3, MPLAB REAL ICE, PICkit 3 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

3. If the project was already built, a cof or elf file was generated. To program the device with the cof, elf or hex file, enter the command:
   Program “[location of the cof or elf or hex file]”.
   For example:
   Program "C:\MDBTestExample\Build\test\preprocess\files\dist\test_IO_Button.cof".
   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. An scl file is used to set the condition.

A “Program succeeded” message displays when programming is complete.
Debugging a Device

You can use the following commands to debug a device. See Figure 1-1 shows an example of using commands to debug a project.

- MCLR Reset - Refer to the device data sheet for reset information. If an MCLR reset is needed for debugging purposes, enter the command:
  
  Reset MCLR

- Target Device Reset - Refer to the device data sheet for reset information. If you need to reset the target device, first halt the target, then use 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 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]
  
  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.
FIGURE 1-1: EXAMPLE OF USING COMMANDS TO DEBUG A PROJECT

Administrator: Command Prompt - mdb
>Reset MCLR
>Break *0x108
Breakpoint 0 at 0x108.
>Run
Running

>Test_IO_Button.c:32:test_IO_Button_Initailize should initialize_the
.
.
.
-----------------------------
9 Tests 0 Failures 0 Ignored
OK
Stop at
   address:0x108
   file:C:/MDBTestExample/Build/test/preprocess/files/Test/su
   source line:53
>Print PROD
PROD=0
>
RUNNING A COMMAND FILE METHOD

If programming and debugging needs to be done frequently or multiple times, run the
test by running a command file. This will save time entering the commands repeatedly.
Put all the commands in a file on the C drive and run it using this command in
Command Prompt, for example:

```
C:\Program Files\Microchip\MPLABX\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. MDB will run all the commands in the command file
sequentially.

FIGURE 1-2: EXAMPLE OF RUNNING A COMMAND FILE

```
<table>
<thead>
<tr>
<th>File</th>
<th>Edit</th>
<th>Format</th>
<th>View</th>
<th>Help</th>
</tr>
</thead>
</table>
| Device PIC18F66K22
| Hwtool SIM
| Program "C:\MDBTestExample\Build\test\preprocess\files\dist\test_IO_Button.cof"
| Reset MCLR
| Sleep 1500
| # set breakpoint at 0x108
| #Break simulator.c:53
| Break "0x108"
| Run
| Wait 600000
| #Sleep 6000
| Print PROD
| Quit
```
## Index

### B
- Break ................................................................. 14
- breakpoints ...................................................... 8

### C
- Classes of Commands ........................................ 8
  - breakpoints .................................................. 8
  - data .............................................................. 8
  - deviceandtool .............................................. 8
  - others .......................................................... 8
  - programming ................................................. 8
  - running ....................................................... 8
  - stack ............................................................ 8
- cof file ............................................................. 13

### Command
- Delete .............................................................. 14
- Device ............................................................. 13
- Print ................................................................. 14
- Program ........................................................... 13
- Quit ................................................................. 14
- Reset MCLR ...................................................... 14
- Run ................................................................. 14
- Sleep ............................................................... 16
- Step ................................................................. 14
- Stim ............................................................... 13

### Command Prompt ........................................... 7

### D
- data ................................................................. 8
- deviceandtool .................................................. 8

### Documentation
- Conventions .................................................... 5

### E
- elf file ............................................................. 13
- Entering Commands Method .............................. 8

### H
- Header Specification .......................................... 6
- Help .................................................................... 9

### M
- MCLR Reset ....................................................... 14
- MDB ............................................................... 7
- Microchip Debugger .......................................... 7

### O
- others ............................................................. 8

### P
- Processor Extension Pak Specification .................. 6
  - programming .................................................. 8

### R
- Reading, Recommended ...................................... 6
- Readme ........................................................... 6
- Reset ............................................................... 14
- running ........................................................... 8
- Running a Command File Method ........................ 16

### S
- scl file ............................................................. 13
- Set Breakpoint ............................................... 14
- Simulator ........................................................ 13
- stack ............................................................... 8
- Stimulus .......................................................... 13

### T
- Target Device Reset .......................................... 14
- Transition Socket
  - Specification ................................................ 6
**Worldwide Sales and Service**

### AMERICAS
- **Corporate Office**
  - 2355 West Chandler Blvd.
  - Chandler, AZ 85224-6199
  - Tel: 480-792-7200
  - Fax: 480-792-7277
  - Technical Support: [http://www.microchip.com/support](http://www.microchip.com/support)
- **Atlanta**
  - Duluth, GA
  - Tel: 678-957-9614
  - Fax: 678-957-1455
- **Boston**
  - Westborough, MA
  - Tel: 774-760-0087
  - Fax: 774-760-0088
- **Chicago**
  - Itasca, IL
  - Tel: 630-285-0071
  - Fax: 630-285-0075
- **Cleveland**
  - Independence, OH
  - Tel: 216-447-0464
  - Fax: 216-447-0643
- **Dallas**
  - Addison, TX
  - Tel: 972-818-7423
  - Fax: 972-818-2924
- **Detroit**
  - Farmington Hills, MI
  - Tel: 248-538-2250
  - Fax: 248-538-2260
- **Indianapolis**
  - Noblesville, IN
  - Tel: 317-773-8323
  - Fax: 317-773-5453
- **Los Angeles**
  - Mission Viejo, CA
  - Tel: 949-462-9523
  - Fax: 949-462-9608
- **Santa Clara**
  - Santa Clara, CA
  - Tel: 408-961-6444
  - Fax: 408-961-6445
- **Toronto**
  - Mississauga, Ontario, Canada
  - Tel: 905-673-0699
  - Fax: 905-673-6509

### ASIA/PACIFIC
- **Asia Pacific Office**
  - Suites 3707-14, 37th Floor
  - Tower 6, The Gateway
  - Harbour City, Kowloon
  - Hong Kong
  - Tel: 852-2401-1200
  - Fax: 852-2401-3431
- **Australia - Sydney**
  - Tel: 61-2-9868-6733
  - Fax: 61-2-9868-6755
- **China - Beijing**
  - Tel: 86-10-8569-7000
  - Fax: 86-10-8528-2104
- **China - Chengdu**
  - Tel: 86-28-8665-5511
  - Fax: 86-28-8665-7889
- **China - Chongqing**
  - Tel: 86-23-8980-9588
  - Fax: 86-23-8980-9500
- **China - Hangzhou**
  - Tel: 86-571-2819-3187
  - Fax: 86-571-2819-3189
- **China - Hong Kong SAR**
  - Tel: 852-2541-1200
  - Fax: 852-2401-3431
- **China - Nanjing**
  - Tel: 86-25-8473-2460
  - Fax: 86-25-8473-2470
- **China - Qingdao**
  - Tel: 86-532-8502-7355
  - Fax: 86-532-8502-7205
- **China - Shanghai**
  - Tel: 86-21-5407-5533
  - Fax: 86-21-5407-5066
- **China - Shenyang**
  - Tel: 86-24-2334-2829
  - Fax: 86-24-2334-2393
- **China - Shenzhen**
  - Tel: 86-755-8203-2660
  - Fax: 86-755-8203-1760
- **China - Wuhan**
  - Tel: 86-27-5980-5300
  - Fax: 86-27-5980-5118
- **China - Xian**
  - Tel: 86-29-8833-7252
  - Fax: 86-29-8833-7256
- **China - Xiamen**
  - Tel: 86-592-2388138
  - Fax: 86-592-2388130
- **China - Zhuhai**
  - Tel: 86-756-3210040
  - Fax: 86-756-3210049

### ASIA/PACIFIC
- **India - Bangalore**
  - Tel: 91-80-3090-4444
  - Fax: 91-80-3090-4123
- **India - New Delhi**
  - Tel: 91-11-4160-8631
  - Fax: 91-11-4160-8632
- **India - Pune**
  - Tel: 91-20-2566-1512
  - Fax: 91-20-2566-1513
- **Japan - Osaka**
  - Tel: 81-66-152-7160
  - Fax: 81-66-152-9310
- **Japan - Yokohama**
  - Tel: 81-45-471-6166
  - Fax: 81-45-471-6122
- **Korea - Daegu**
  - Tel: 82-53-744-4301
  - Fax: 82-53-744-4302
- **Korea - Seoul**
  - Tel: 82-2-554-7200
  - Fax: 82-2-558-5932 or 82-2-558-5934
- **Malaysia - Kuala Lumpur**
  - Tel: 60-3-6201-9857
  - Fax: 60-3-6201-9859
- **Malaysia - Penang**
  - Tel: 60-4-227-8870
  - Fax: 60-4-227-8850
- **Philippines - Manila**
  - Tel: 63-2-634-9065
  - Fax: 63-2-634-9069
- **Singapore**
  - Tel: 65-6334-8870
  - Fax: 65-6334-8850
- **Taiwan - Hsin Chu**
  - Tel: 886-3-5778-366
  - Fax: 886-3-5770-955
- **Taiwan - Kaohsiung**
  - Tel: 886-7-213-7828
  - Fax: 886-7-330-9305
- **Taiwan - Taipei**
  - Tel: 886-2-2508-8600
  - Fax: 886-2-2508-0102
- **Thailand - Bangkok**
  - Tel: 66-2-694-1351
  - Fax: 66-2-694-1350

### EUROPE
- **Austria - Wels**
  - Tel: 43-7242-2244-39
  - Fax: 43-7242-2244-393
- **Denmark - Copenhagen**
  - Tel: 45-4450-2828
  - Fax: 45-4485-2829
- **France - Paris**
  - Tel: 33-1-69-53-63-20
  - Fax: 33-1-69-30-90-79
- **Germany - Munich**
  - Tel: 49-89-627-144-0
  - Fax: 49-89-627-144-44
- **Italy - Milan**
  - Tel: 39-0331-742611
  - Fax: 39-0331-466781
- **Netherlands - Drunen**
  - Tel: 31-416-690399
  - Fax: 31-416-690340
- **Spain - Madrid**
  - Tel: 34-91-708-08-90
  - Fax: 34-91-708-08-91
- **UK - Wokingham**
  - Tel: 44-118-921-5869
  - Fax: 44-118-921-5820

10/26/12