11019 MPA

MPLAB® In-Circuit Debuggers for Advanced Users

Class Objective

- **Convey a picture of the MPLAB® Debug Tools the way the developers see them**
  - With emphasis on MPLAB ICD 2
- **Cover the topics that the developers wished the users knew**

Class Objective

- **Compare and Contrast the Debug Tools**
  - MPLAB® REAL ICE™ In-Circuit Emulator
  - MPLAB ICD 2
  - PICkit™ 2 Starter Kit

University of Microchip UofM MASTERs 2007

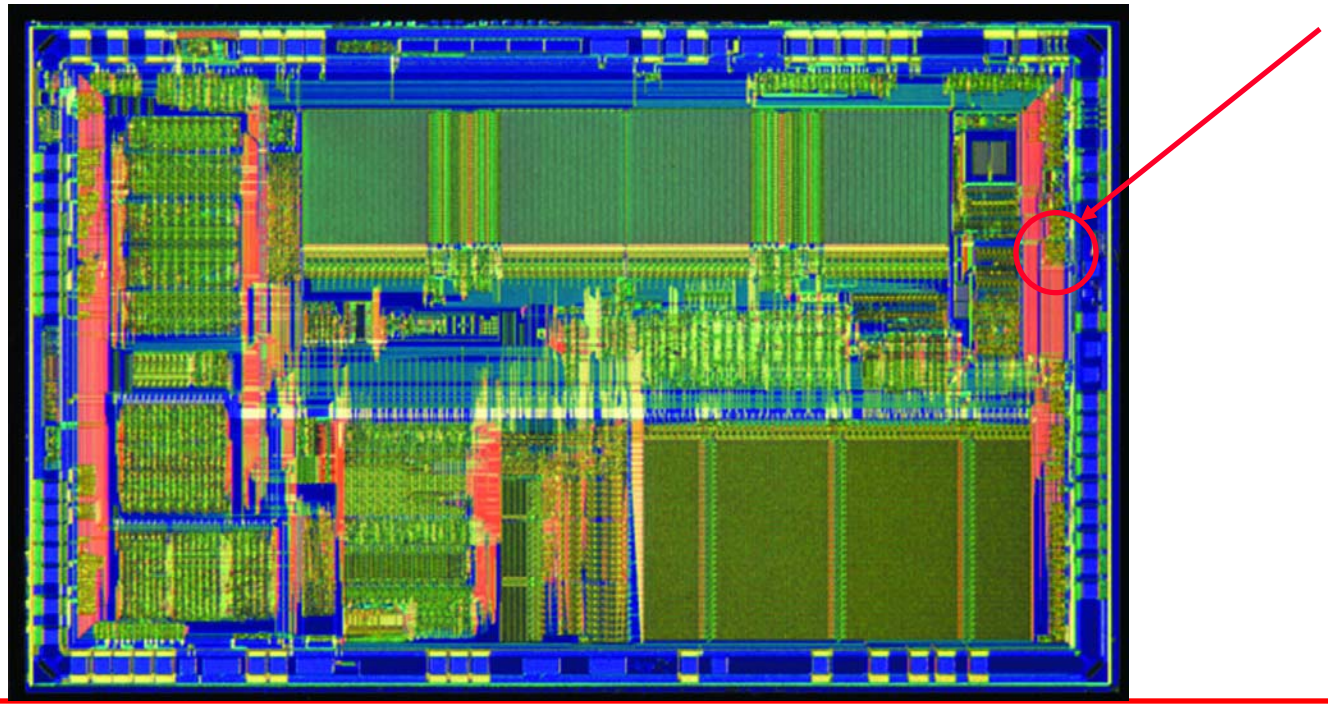
Agenda

- How the MPLAB® Debug Tools Work
- Consequences and Gotchas
- Advanced Breakpoints
- Comparing the Different Tools
- Odds and Ends

How the MPLAB® Debug Tools Work

How the MPLAB® Debug Tools Work

- It all starts with a special Background Debug Module (BDM) inserted into the silicon



How the MPLAB® Debug Tools Work

- The BDM provides a special Non-Maskable Interrupt (NMI) triggered on certain debugging events:
  - External Halt
  - Breakpoint Match
  - Single-Step Execution

- The chip runs in normal user mode until one of these debugging events occurs

How the MPLAB® Debug Tools Work

- When the debug event occurs, the silicon issues the debug NMI (called a Halt) and vectors to a known location
  - May or may not be in user's program memory space
  - The chip is said to be in the INBUG mode at this point
- The Debug Tool programs a special interrupt handler at this location known as a Debug Executive (DE)

How the MPLAB® Debug Tools Work

- The DE simply process commands from the Debug Tool until ordered to return to the user's program
  - Command set is simplistic:
    - Read/Write a register
    - Perform a single step
    - Return to user mode
- The DE often needs user resources to execute

How the MPLAB® Debug Tools Work

- Communications between the DE and Debug Tool are bit-banged through the ICDC and ICDD pins
  - Sometimes called PGC-PGD or RB6-RB7 or EMUC-EMUD
  - Usually the same pins as used for ICSP™ technology
  - The DE controls the bit-bang clock
  - Don’t use these pins in the user code
    - Will cause a HALT or skew the HALT handshake

How the MPLAB® Debug Tools Work

- While the silicon is in INBUG mode, certain SFRs become available which allow control of the BDM
  - These SFRs allow:
    - Specification of breakpoint conditions
    - Single step operation
    - Control of the ICDC and ICDD pins

How the MPLAB® Debug Tools Work

- The Debug Tool system is set up as a master/slave relationship
  - MPLAB® IDE initiates all communications with the Debug Tool
  - The Debug Tool initiates all communications with the DE
  - Simplifies system by not attempting to handle asynchronous bi-directional communications
    - More robust system would have used more user resources

How the MPLAB® Debug Tools Work

- In circuit debugging offers higher fidelity debugging than emulators and simulators
  - Use actual silicon in actual target environment
    - Actual power, clock and timing
  - Not necessarily the most powerful debugging
  - Some smaller parts break this paradigm

Consequences and Gotchas

Consequences and Gotchas

- INBUG, ICSP™ technology and User Modes
- Breakpoints
- Registers

Consequences and Gotchas

INBUG, ICSP™ technology and User Modes

Consequences and Gotchas

INBUG, ICSP™ technology and User Modes

- From the Debug Tool point of view, the target microcontroller can be in 1 of 4 states:
  - Reset : MCLR = V_{il} (boring)
  - ICSP : MCLR = V_{pp} (programming)
  - User : MCLR = V_{dd} (user code)
  - INBUG : MCLR = V_{dd} (debug executive)
- Reset is the default state; the Debug Tool will move the target to Reset whenever there is a problem
- MPLAB® REAL ICE™ In-Circuit Emulator defaults to a tri-state MCLR

Consequences and Gotchas

INBUG, ICSP™ technology and User Modes

- The Debug Tool can only read or write particular memories in specific modes
  - This is slowly changing in certain situations

	Program	File Reg	EEDATA	Config
Reset	-	-	-	-
ICSP	X	-	X	X
User	-	-	-	-
INBUG	-	X	-	-

Consequences and Gotchas

INBUG, ICSP™ technology and User Modes

- Occasionally the Debug Tool is asked to read or write memory in the wrong mode
  - We've eradicated most of these occurrences
  - ICD0157: Attempted target memory access using an invalid type and mode combination (Mem = %s) (Mode = %s) (Type = %s)
- DE could perform INBUG program memory access, but would grow in size to do so
  - For some devices (PIC24, dsPIC30, dsPIC33) we don't care
- *Switching modes causes a target reset*

Consequences and Gotchas

INBUG, ICSP™ technology and User Modes

- One of the most prevalent error messages is
  - ICD0083: Debug: Unable to enter debug mode.
- MCLR is at Vdd, but the Debug Tool is unable to establish communications with the DE

Consequences and Gotchas

INBUG, ICSP™ technology and User Modes

- Hardest part to using a Debug Tool
  - Application must be executable before DE can actually run
  - Is power, oscillator, connection etc. correct
  - Try switching to programmer mode and run a “blinky-light” program

Consequences and Gotchas

Breakpoints

Consequences and Gotchas

Breakpoints

- The Debug Tool sets breakpoints by writing the desired address to INBUG SFRs in the BDM
- While in user mode, the BDM compares each fetched instruction address with the desired break address and generates a HALT on a match
- But the instruction has already entered the microcontroller pipeline so the HALT NMI is delayed until after that instruction finishes execution

Consequences and Gotchas

Breakpoints

- The PC read by the Debug Tool after the HALT contains the address of the *next* instruction to be executed

- This is known as Skidding (demo)
  - Breaking on a 2 word instruction will result in a skid of 2 words rather than 1 word
  - dsPIC® DSC and PIC24F devices skid 2 instructions
  - Breaking on branch instructions can be confusing because of skid (demo)
  - NOPs can be used to ease this problem (demo)

Consequences and Gotchas

Breakpoints

- Breakpoints hit during Animate do *not* skid (demo)
  - Animate is simply automated single stepping
  - MPLAB® IDE compares the address *before* the instruction is executed

- Different microcontrollers offer different numbers of breakpoints
  - PIC10, PIC12 and PIC16 — 1 BP
  - PIC18 — 1 or 3 or 5 BPs
  - dsPIC30 — 1 or 2 BPs
  - PIC24 & dsPIC33 — 4 BPs

Consequences and Gotchas

Breakpoints

- The number of available breakpoints affects Step-Over performance
  - MPLAB® IDE will attempt to temporarily use available breakpoints for step-over target address (demo)
  - If insufficient breakpoints are available, MPLAB IDE will instead switch to *Animate* mode until the step-over target is reached (demo)
    - This may take a long time
  - Note that breakpoints and advanced breakpoints are really the same thing

Consequences and Gotchas

Registers

Consequences and Gotchas

Registers

- To update the value of a register displayed in MPLAB® IDE, the Debug Tool must perform a read operation through the DE
  - This transaction takes a finite amount of time
  - On slow targets, even more time is needed
  - So, the more data you display in MPLAB IDE, the longer it will take Debug Tool to halt or step (demo)
  - Microchip recommends the judicious use of the Watch Window instead of the File Register or SFR Windows

Consequences and Gotchas

Registers

- SFRs are viewed as just more file registers by the Debug Tool
- This presents some challenges for users
  - Reading or writing SFRs typically have side-effects
  - For example, reading the INDF flag has the side effect of incrementing the indirect address
  - Remember that displaying the INDF register in the watch, file register, or SFR windows will result in Debug Tool reading this register on each Halt

Consequences and Gotchas

Registers

- The INDF register is somewhat obvious and the Debug Tool knows to skip over requests to read this SFR (demo)

- Other SFRs can be just as troublesome though
  - For example, the RCREG register
  - We suggest reading the SFR into a GPR and working with the GPR in your software; and watch the GPR instead of the SFR (demo)

Advanced Breakpoints

UNIVERSITY OF MICROCHIP UOFM MASTERS 2007

Advanced Breakpoints

- Over the years, Microchip has enhanced the capabilities of the BDMs placed in its microcontrollers
- Newer devices have break capabilities beyond simply matching a program memory address
- Exactly what those capabilities are depends upon which device you are targeting
- Access to these Advanced Breakpoints is provided through the Debug menu

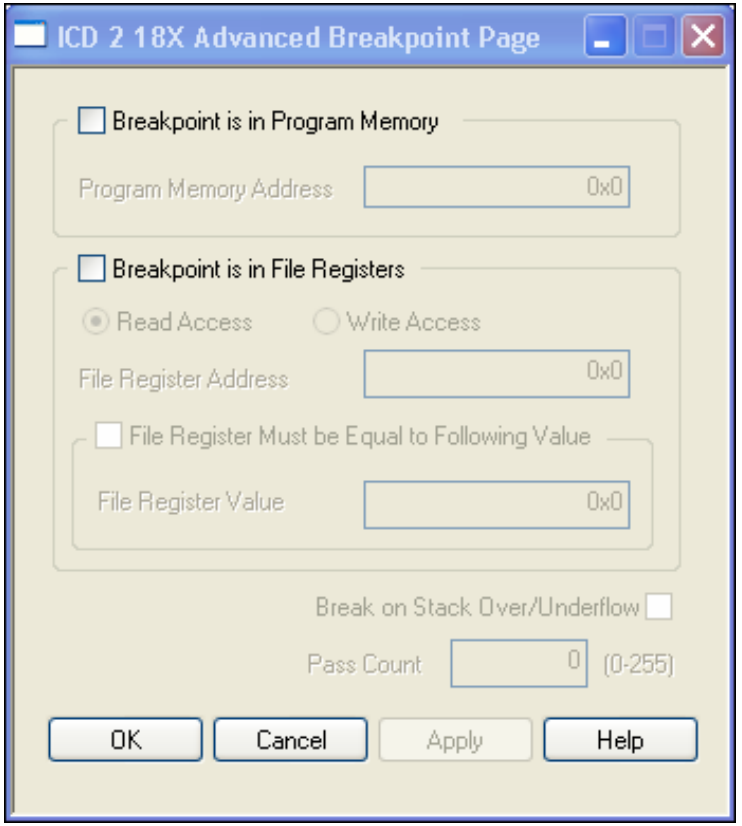
Advanced Breakpoints

- PIC10, PIC12 and PIC16 devices have 1 program memory only break point
  - No Advanced BP capability

University of Microchip UofM MASTERs 2007

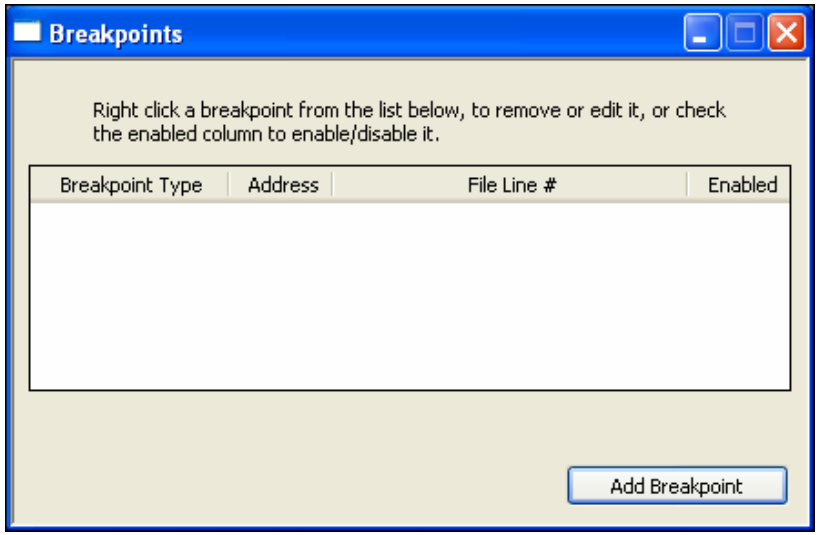
Advanced Breakpoints

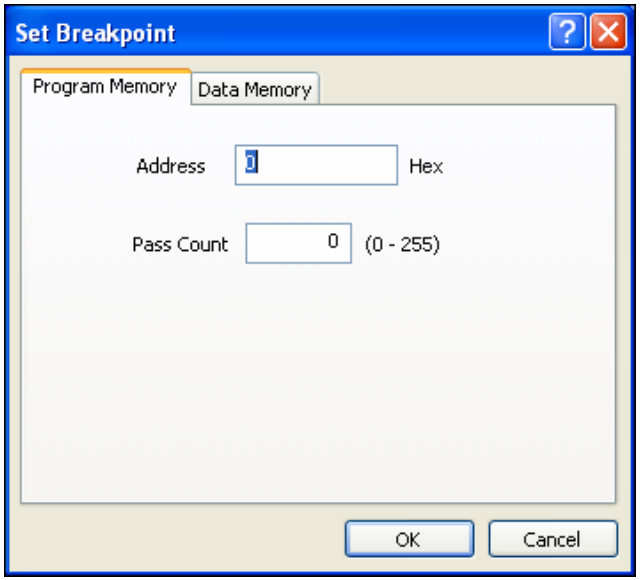
- PIC18F parts have more capabilities, but still only 1 breakpoint
  - Data match (demo)
  - Pass counting
  - *Break on Stack*

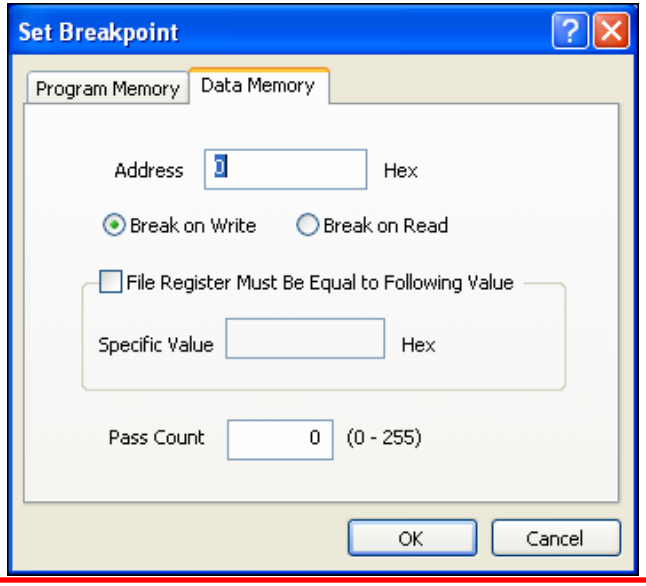


Advanced Breakpoints

- **MPLAB® REAL ICE™ In-Circuit Emulator**
  - Dialogs different
  - Fields the same

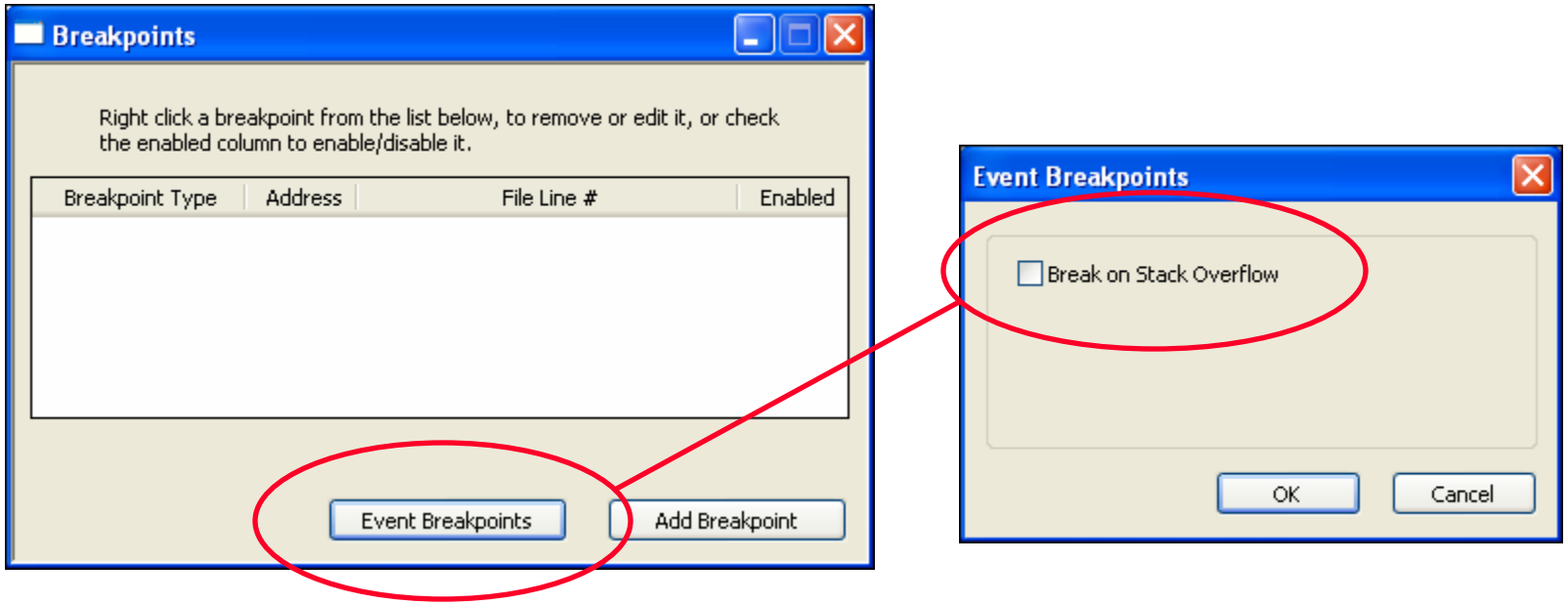






Advanced Breakpoints

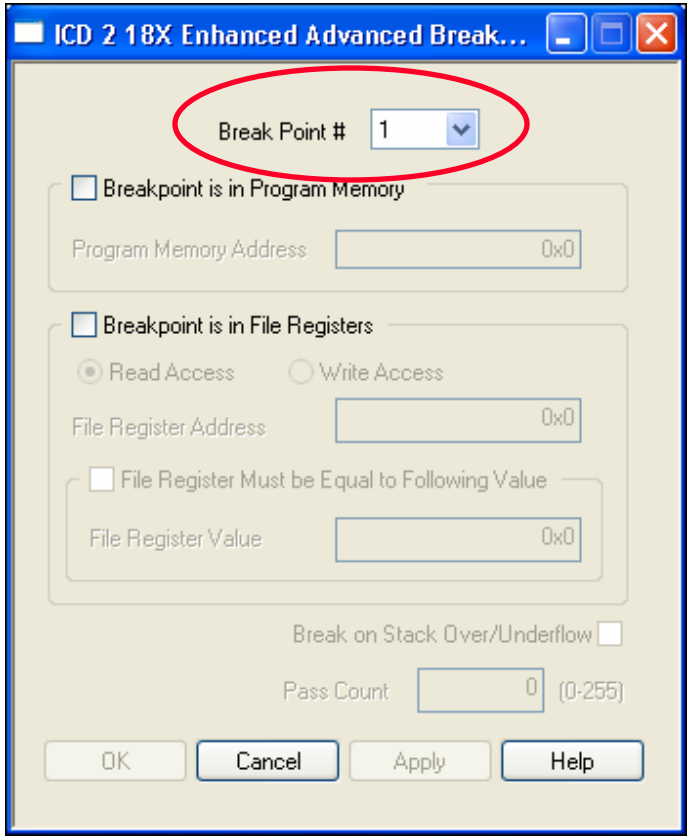
- **MPLAB® REAL ICE™ In-Circuit Emulator**
  - Break on Stack works



University of Microchip UofM MASTERs 2007

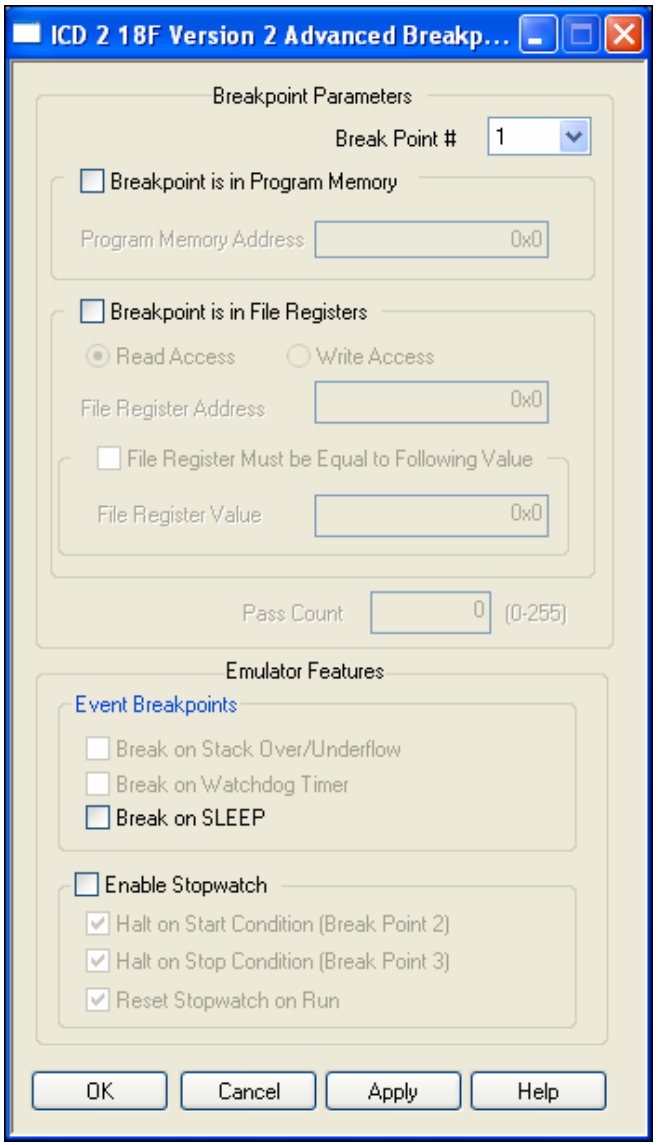
Advanced Breakpoints

- PIC18F Extended parts have the same breakpoint capabilities
  - 3 breakpoints



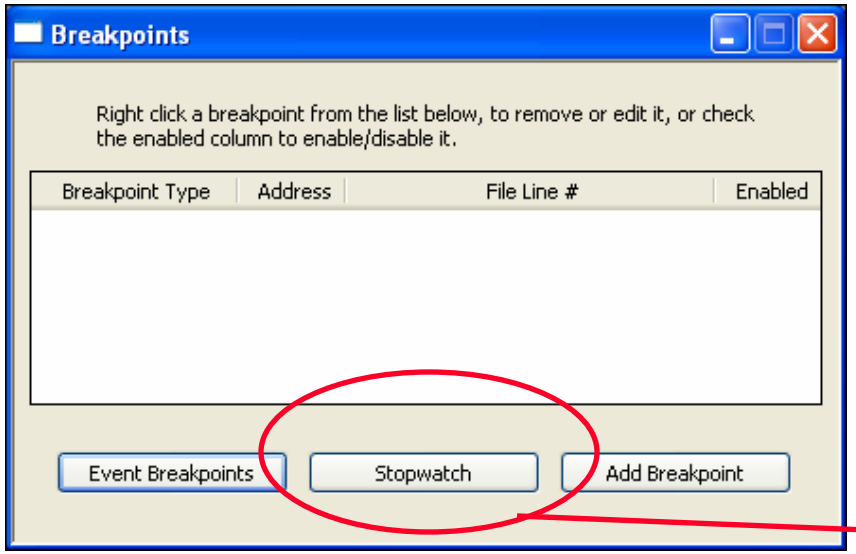
Advanced Breakpoints

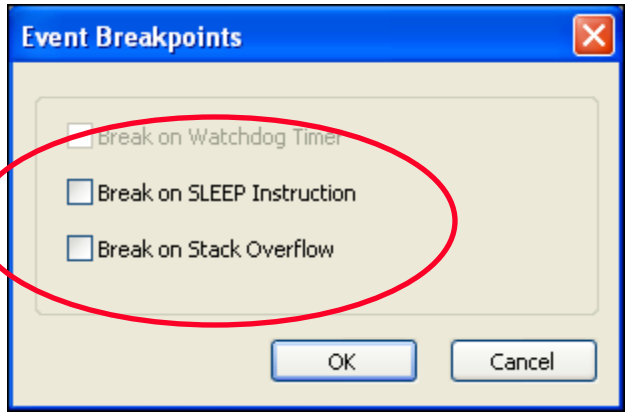
- **PIC18F "J" parts have even more features**
  - Break on Watchdog
  - Break on Sleep
  - Stopwatch
  - *Real-time data watch*

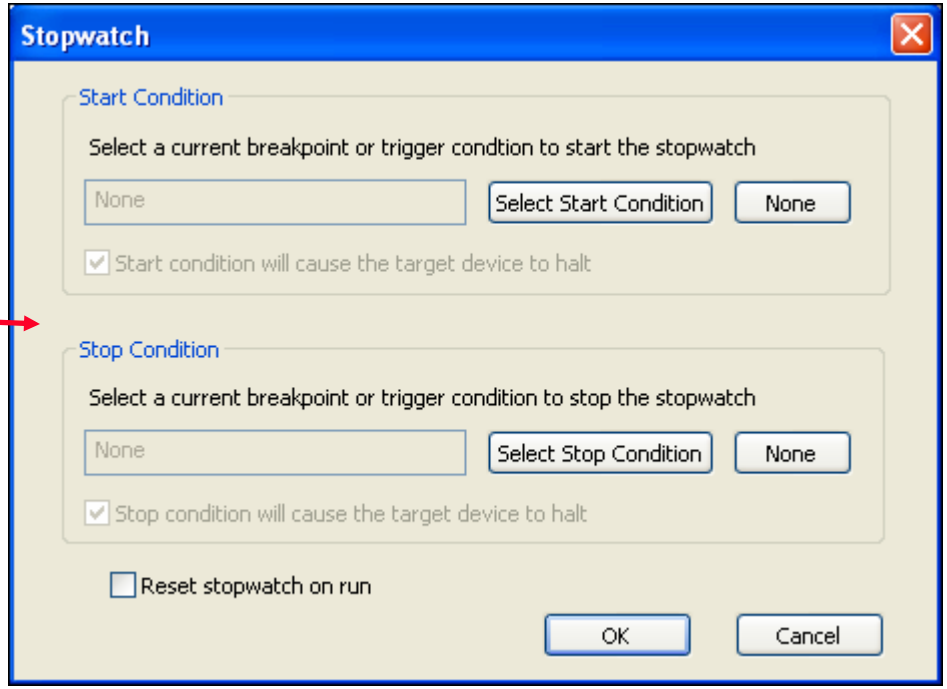


Advanced Breakpoints

- **MPLAB® REAL ICE™ In-Circuit Emulator**

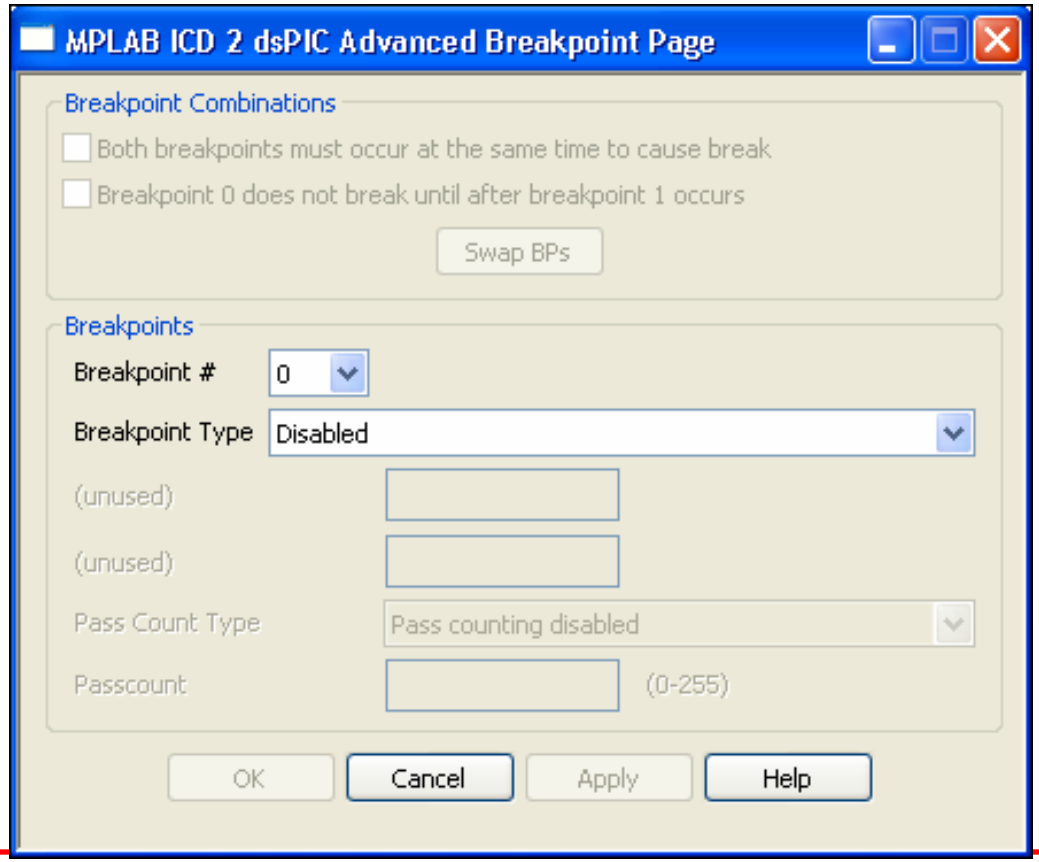






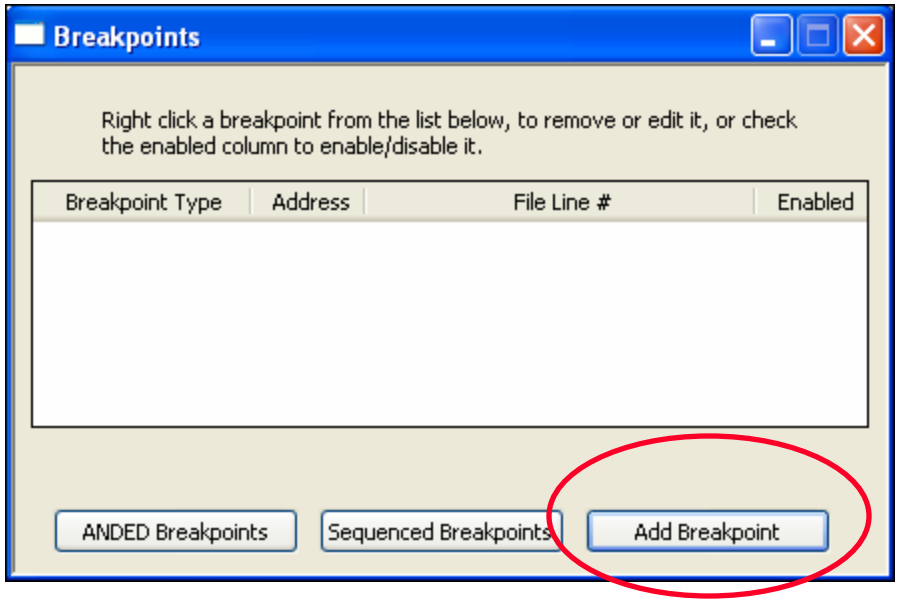
Advanced Breakpoints

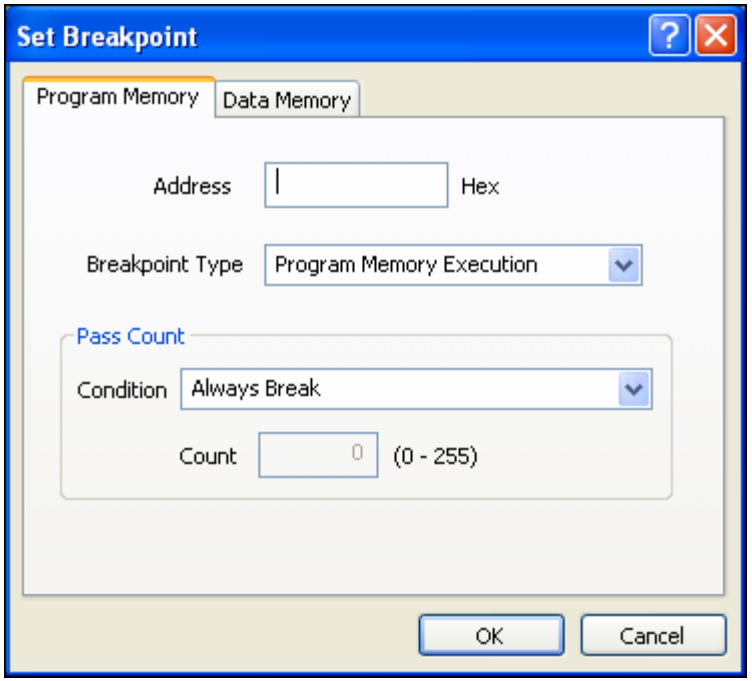
- **dsPIC® DSC has a somewhat different approach**
  - Combinations
  - Different busses
  - 2 passcounts



Advanced Breakpoints

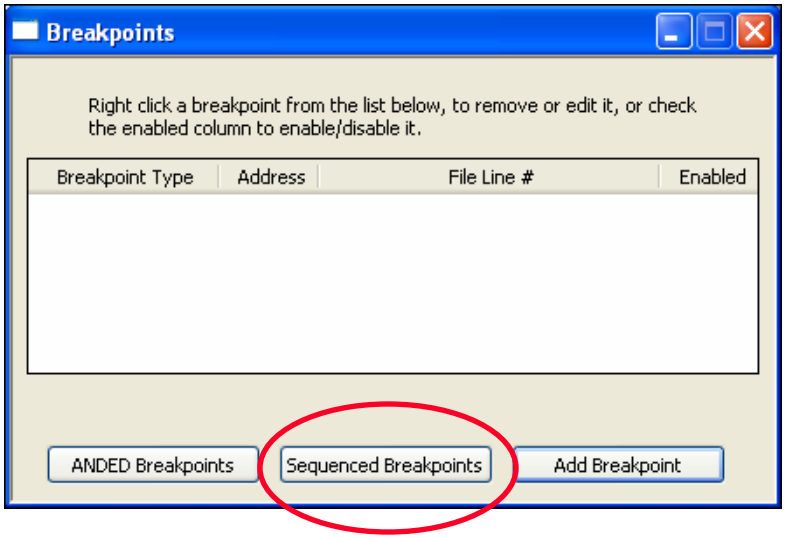
- **MPLAB® REAL ICE™ In-Circuit Emulator**

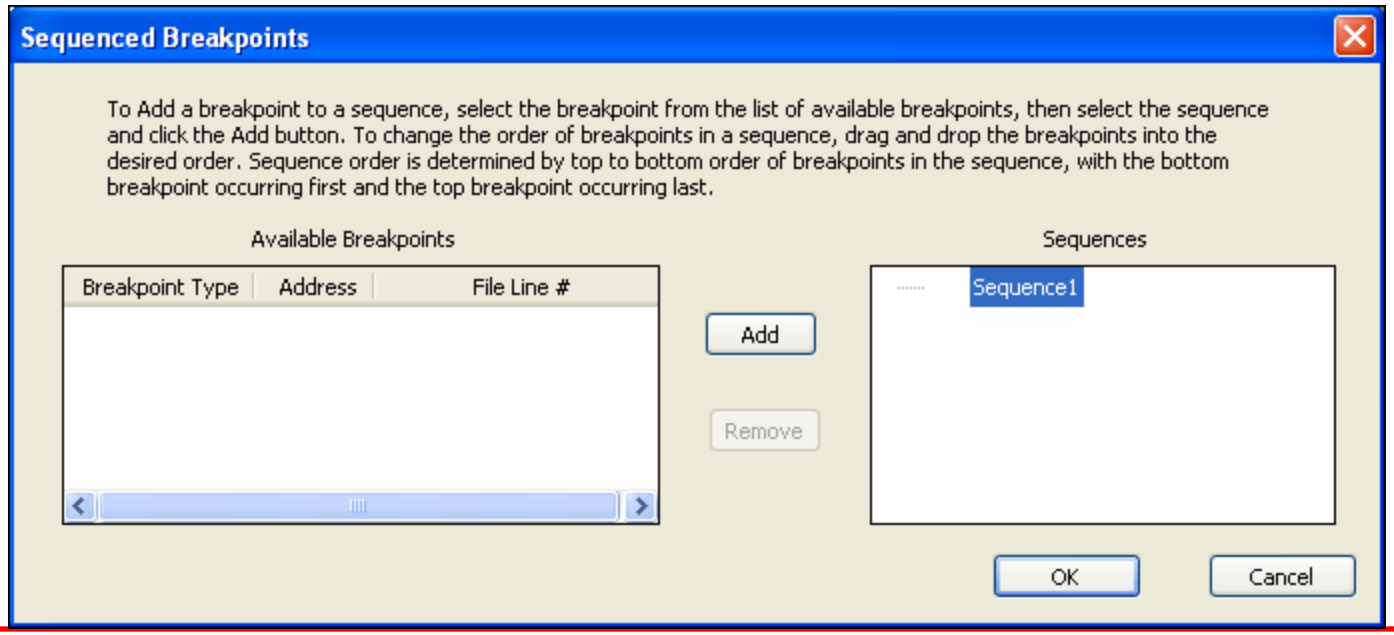




Advanced Breakpoints

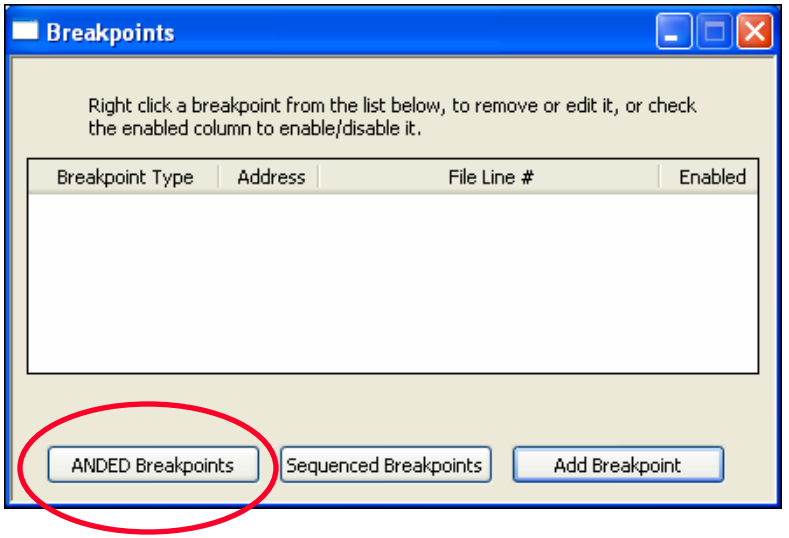
- **MPLAB® REAL ICE™ In-Circuit Emulator**

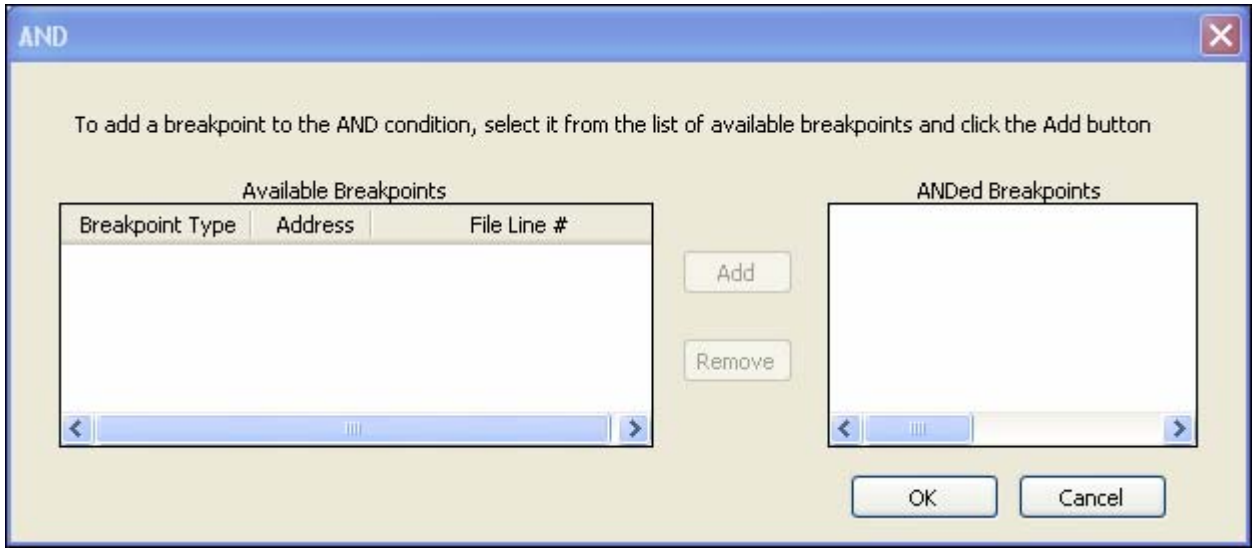




Advanced Breakpoints

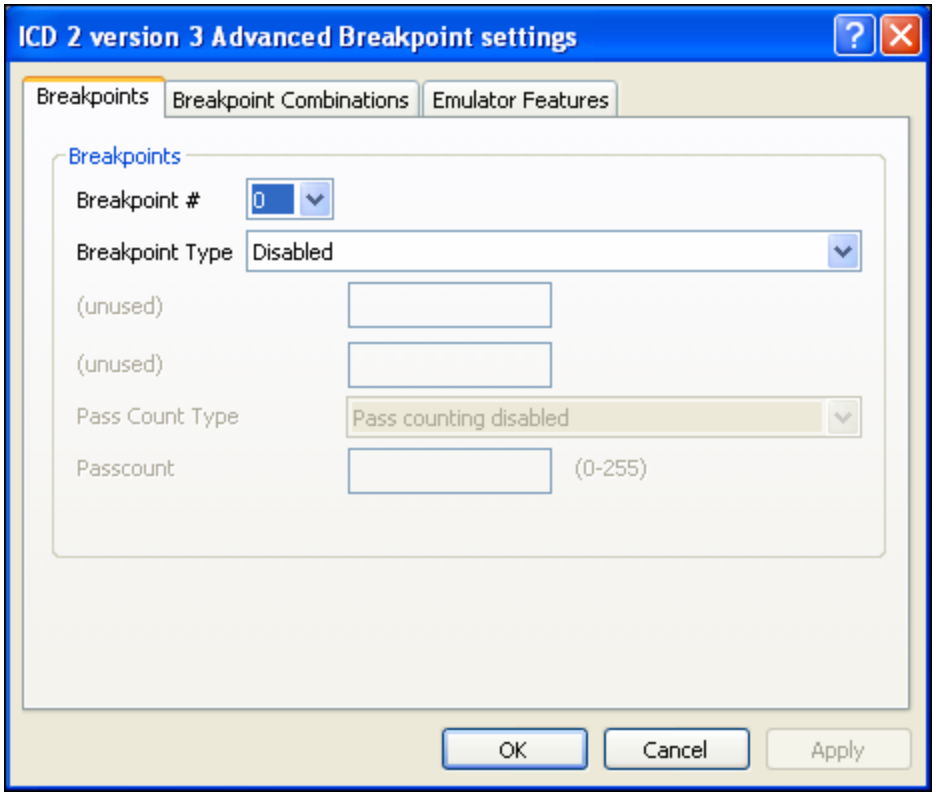
- **MPLAB® REAL ICE™ In-Circuit Emulator**





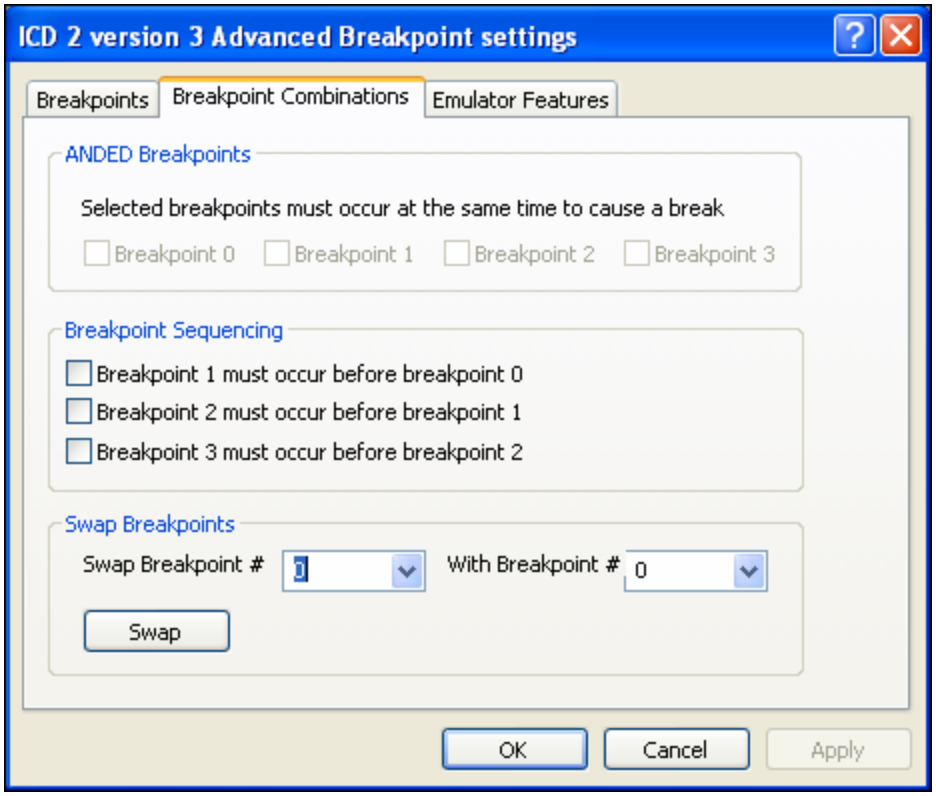
Advanced Breakpoints

- **PIC24 and dsPIC33 have all the features of both**
  - Tabbed presentation



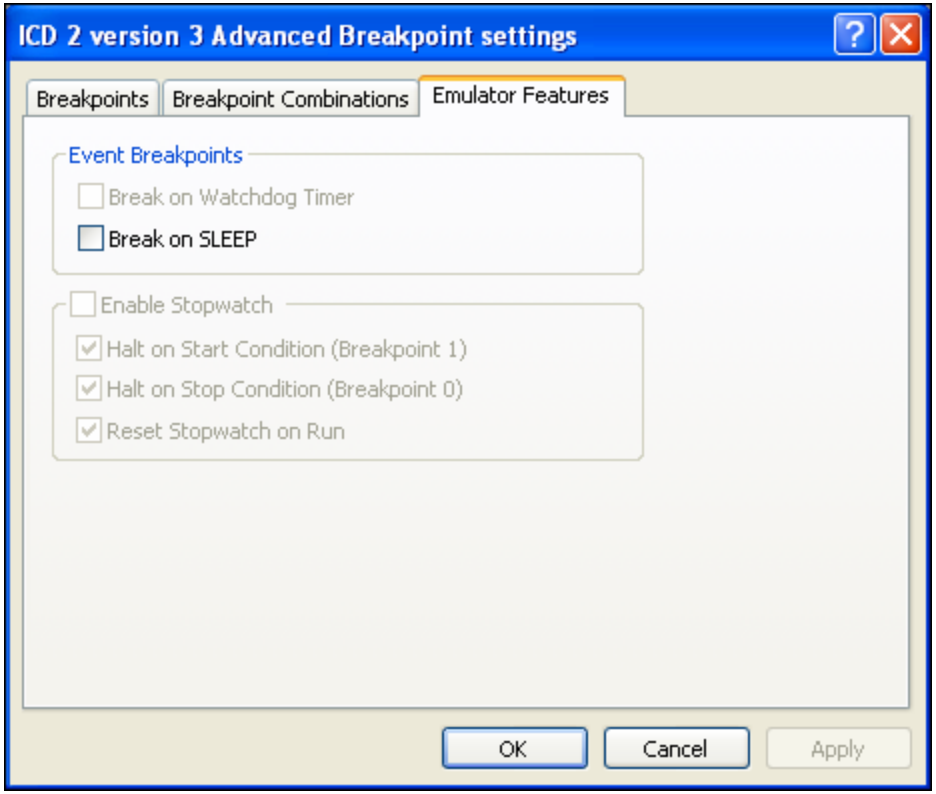
Advanced Breakpoints

- **PIC24 and dsPIC33 have all the features of both**
  - Tabbed presentation



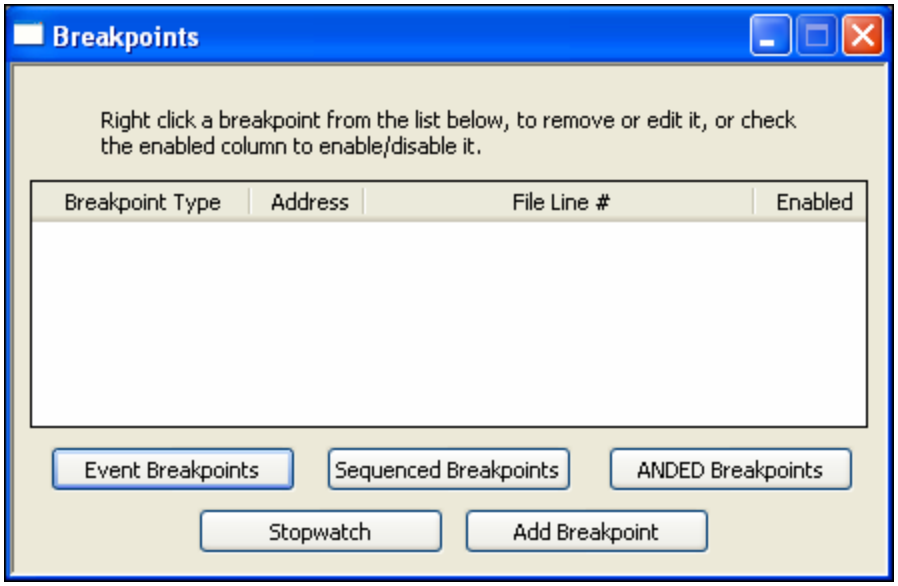
Advanced Breakpoints

- **PIC24 and dsPIC33 have all the features of both**
  - Tabbed presentation



Advanced Breakpoints

- **MPLAB® REAL ICE™ In-Circuit Emulator**



Comparing the Different Tools

Comparing the Different Tools

- **The BDM means that at some level all the Debug Tools have the same functionality**
- **So why choose one tool over any other?**
- **Price is usually the first discriminating criteria used!**
  - These prices were pulled from the Microchip Web page. They may be different today, and ARE different here at MASTERs!
    - **PICkit™ 2 Starter Kit** **$35**
    - **MPLAB® ICD 2** **$160**
    - **MPLAB REAL ICE™ In-Circuit Emulator** **$500**
- **There MUST be a reason for these price differences!**

Comparing the Different Tools

- **Let's start by looking at communications**
  - PICkit™ 2 Starter Kit
    - **Uses full speed USB, Human Interface Device**
      - No special driver needed
      - Packets limited to 64 bytes
  - MPLAB® ICD 2
    - **Uses full speed USB, Custom Bulk Device**
      - Special Microchip driver needed
      - Large packets allowed
  - MPLAB REAL ICE™ In-Circuit Emulator
    - **Uses high speed USB, Custom Bulk Device**
      - Special Microchip driver needed
      - Large packets allowed

Comparing the Different Tools

- **Next let's examine on board resources**
  - PICkit™ 2 Starter Kit
    - **PIC18F2550**
      - 32K Flash, 4K RAM, 48 MHz
  - MPLAB® ICD 2
    - **PIC16F877**
      - 8K Flash, .5K RAM, 20 MHz
  - MPLAB REAL ICE™ In-Circuit Emulator
    - **dsPIC30F6014A**
      - 48K Flash, 10K RAM, 120 MHz

Comparing the Different Tools

- **Supported Parts**
  - PICkit™ 2 Starter Kit
    - **About 40 selected PIC10, PIC12, PIC16 and PIC18 devices**
    - **More being added slowly**
  - MPLAB® ICD 2
    - **All PIC10F, PIC12F, PIC16F, PIC18F, PIC24F, dsPIC30Fand dsPIC33F devices**
  - MPLAB REAL ICE™ In-Circuit Emulator
    - **All PIC18F, PIC24F, dsPIC30F and dsPIC33F devices**
    - **PIC10F, PIC12F and PIC16F coming soon**

Comparing the Different Tools

- **PICkit™ 2 Starter Kit specifics**
  - Very inexpensive
  - Limited part support
  - Works with MPLAB® IDE or standalone
  - Slow
  - No protection on clock, data and MCLR lines
    - **Cross-talk interference, etc.**

Comparing the Different Tools

- **MPLAB® ICD 2 specifics**
  - Medium price
  - Supports everything
  - Tons of line protection
  - Tons of software protection
    - **Code protect, invalid config fields, power checks, etc.**
  - Mature tool, not many kinks left
  - Technology getting old; questionable debugging capability for newer, larger faster devices

University of Microchip UofM MASTERs 2007

Comparing the Different Tools

- **MPLAB® REAL ICE™ In-Circuit Emulator**
  - Pricey, but value is there
  - No support for PIC10, PIC12, or PIC16 yet but will be there soon
  - **Hot pluggable!**
  - **Real time data watch!**
    - Combines very nicely with Data Monitor and Control Interface plugging
  - **Instrumented trace!**
  - **Long cable option!**
  - Good line protection
  - A lot of the software protection still to come, but will be there eventually
  - Technology new so the tool will be useable for many years to come

Comparing the Different Tools

- **So which tool do you use?**
  - Depends on your needs and budget
  - Some customers will buy one or two MPLAB® REAL ICE™ In-Circuit Emulator tools for the lab and MPLAB ICD 2 tools for each engineer's desk
  - If you are debugging a high speed data intensive application, MPLAB REAL ICE In-Circuit Emulator is probably the way to go

Odds and Ends

Odds and Ends

- **None of the tools can perform a Power-On-Reset**
  - PICkit 2 Starter Kit can provide power up to 50mA from the USB bus
  - MPLAB® ICD 2 can provide power up to 250mA, but not from the USB bus
  - MPLAB REAL ICE™ In-Circuit Emulator will not provide target power
  - Microchip recommends that the target provide its own power

University of Microchip UofM MASTERs 2007

Odds and Ends

- Forums are a great place to look for help
  - http://forum.microchip.com/

UNIVERSITY OF MICROCHIP UOFM MASTERS 2007

Trademarks

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KeeLoq, KeeLoq logo, microID, MPLAB, PIC, PICmicro, PICSTART, PRO MATE, rfPIC and SmartShunt are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AmpLab, FilterLab, Linear Active Thermistor, Migratable Memory, MXDEV, MXLAB, SEEVAL, SmartSensor and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, ECAN, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, PICkit, PICDEM, PICDEM.net, PICLAB, PICtail, PowerCal, PowerInfo, PowerMate, PowerTool, REAL ICE, rfLAB, Select Mode, Smart Serial, SmartTel, Total Endurance, UNI/O, WiperLock 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.

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