

Visual Studio C++ Debugger Tutorial

Chris Pu CSCE-315

Why you need to learn how to use debugger?

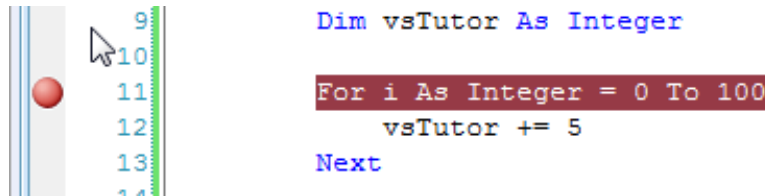
- What is debugger?
 - A tool allow you to monitor your code's variable value at runtime
 - Identify which code is incorrect?
 - **Quote:** “If there's a single feature of Visual Studio that **every** developer uses and is **essential** to the development process it is the built-in debugger”

Prior to Debugging

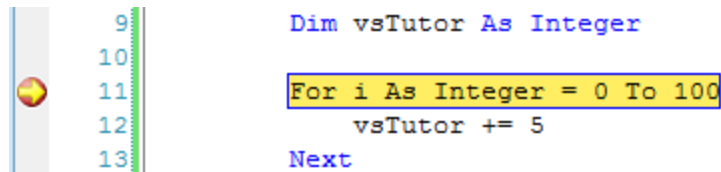
- Setup **breakpoint**

- Breakpoints **pause** the execution of the code and allow developers to examine controls and variables before allowing the program to continue to execute.

- Looks like:



- Execution paused at runtime:



Practice

- How to set up a **breakpoint** in a program?
 - Move your mouse to the left margin of a line you want to pause
 - Then, left click your mouse, and you will see a red circle appears. It means a **breakpoint** is set at this line.

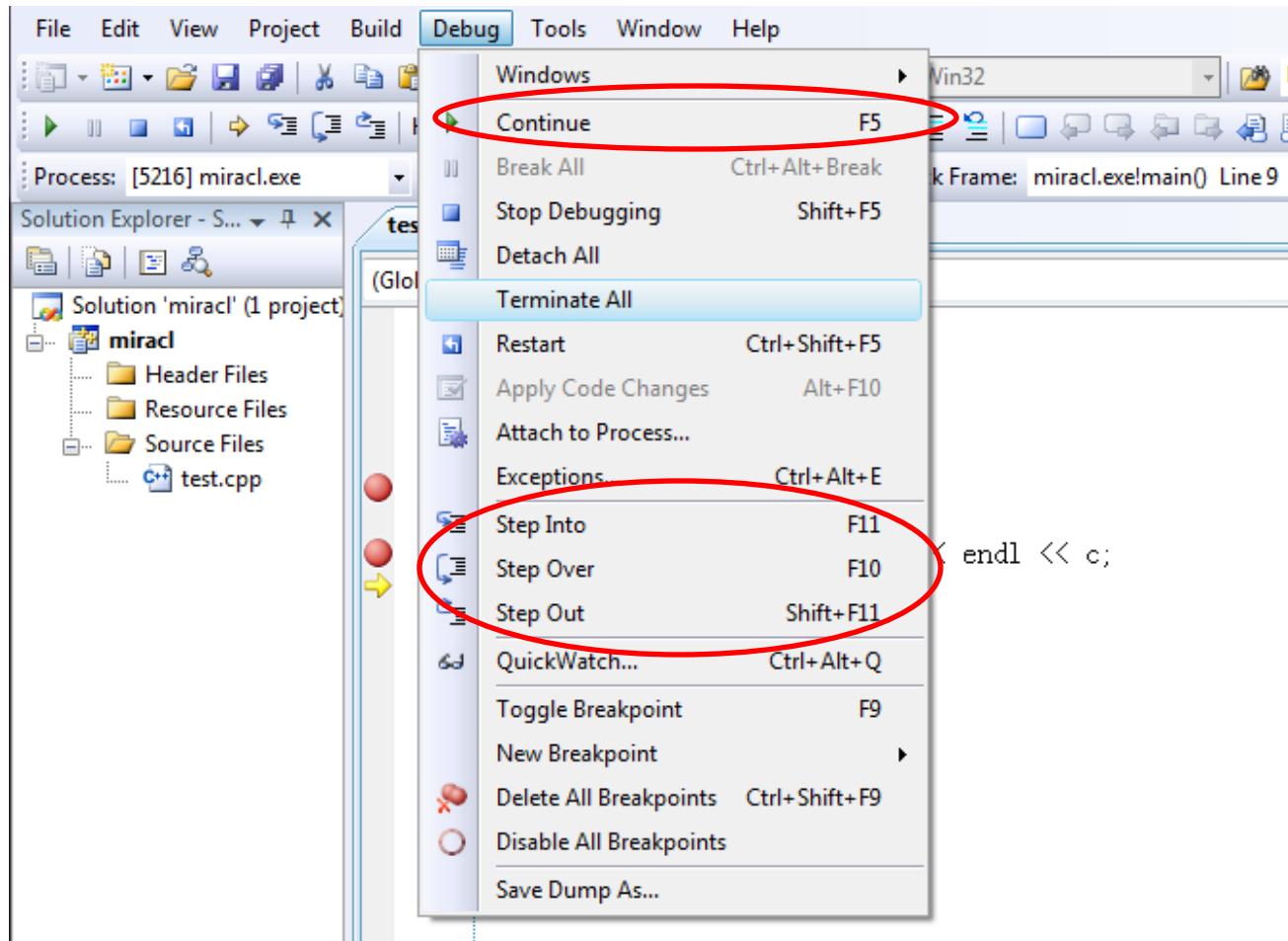
Debugging

- Important windows:
 - 1. **watch** window: it allows monitoring of any object which can simply be highlighted in the code window and dragged to the **Watch** window. The Watch window monitor objects regardless of whether they are in scope or now.
 - 2. **Locals** window: it cannot have objects dragged into it and shows all objects that are currently in scope.
 - A powerful feature of the locals window is that it allow the objects to by modified.

- Important windows:
 - **Autos** window: it shows the objects used in the execution of the current statement.

- At all times the yellow arrow shows the line of code that will be executed next.
 - Advanced feature: this arrow can be dragged down to execute subsequent lines of code without executing the code that it is dragged over (you may not need to use this feature in this class).

Runtime Control



- Continue (F5): resume the running process of a program, will be paused by the next breakpoint.
- Step Into(F11): executes the next line of code that the program would normally execute (if the current line is a function invocation, it will **jump into** the **function** execution)

- Step Over(F10): proceeds to the next line of code in the current procedure, this means that other routines (such as **functions**) are called it will **not proceed into** those routines but **simply execute** them and continue to the next line of the current code block.

- Step Out (shift+F11): will move to the line of code which called the current process or the next breakpoint if that comes first.
- Restart (ctrl+shift+F5): restart the program execution. It usually occurs:
 - when you change the code during debugging.

Practice & Question?

SVN

- SVN: online code repository
 - Purpose: code management
- Online Help:
 - <http://www.yolinux.com/TUTORIALS/Subversion.html>
 - Or google with key words “SVN command”
- Concepts:
 - Top of tree: your latest code modification in repository
 - Local code base: code in your local development environment

- svn checkout: get code from repository
- svn sync: get the Top of tree from repository to your local code base
- svn add/delete: create/remove a (new) file in your local code base, and add/remove it into/from repository
- svn merge: merge your latest changes of files to repository

- `svn commit`: make `svn add/delete/update` commands effective.
 - Give you a chance to regret 😊
- `svn revert`: if you change some file in repository by mistake, revert it to a earlier version