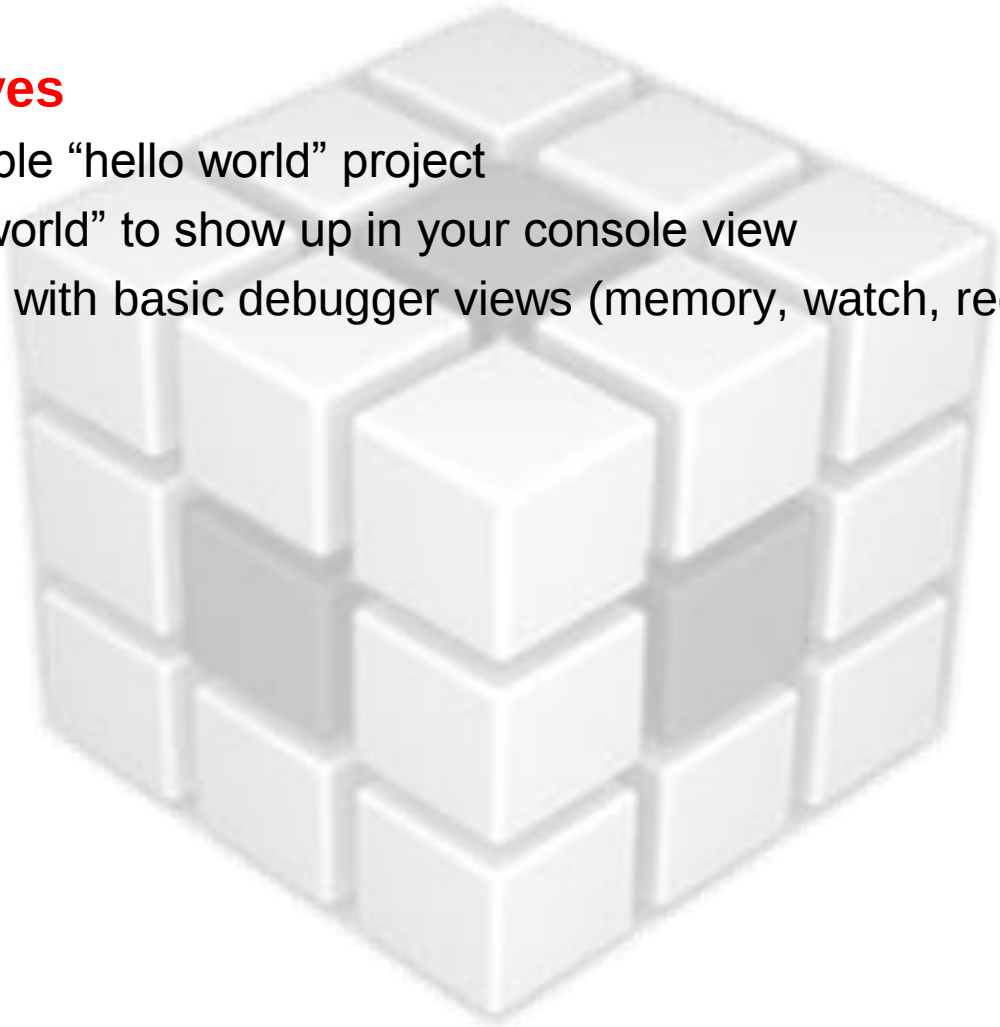
Getting Started with CCSv4
“Hello World”

JOHN STEVENSON

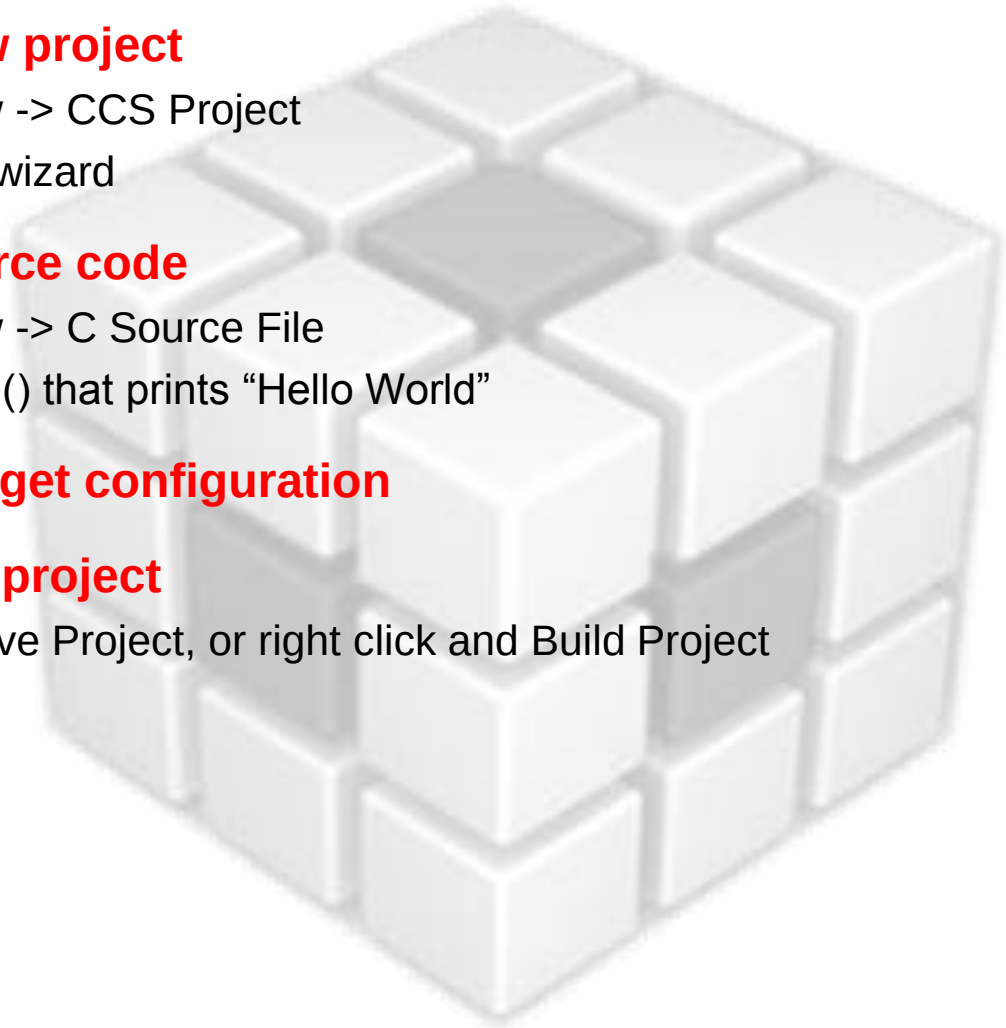
Hello World: Briefing



- **Key Objectives**
  - Build a simple “hello world” project
  - Get “hello world” to show up in your console view
  - Get familiar with basic debugger views (memory, watch, register, etc)

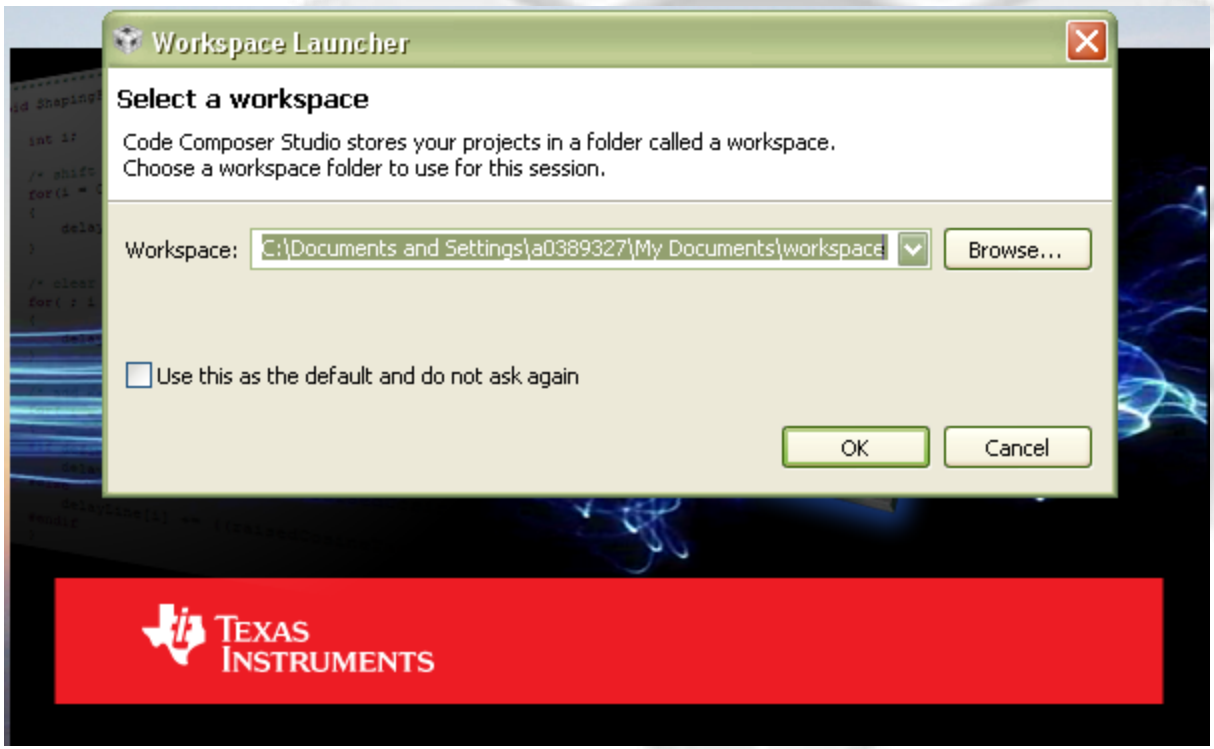
Hello World: Project Summary

- **Create a new project**
  - File -> New -> CCS Project
  - Follow the wizard
- **Add the source code**
  - File -> New -> C Source File
  - Add a main() that prints “Hello World”
- **Specify a target configuration**
- **Compile the project**
  - Debug Active Project, or right click and Build Project



Workspace

- **Launch CCS and select a workspace folder**

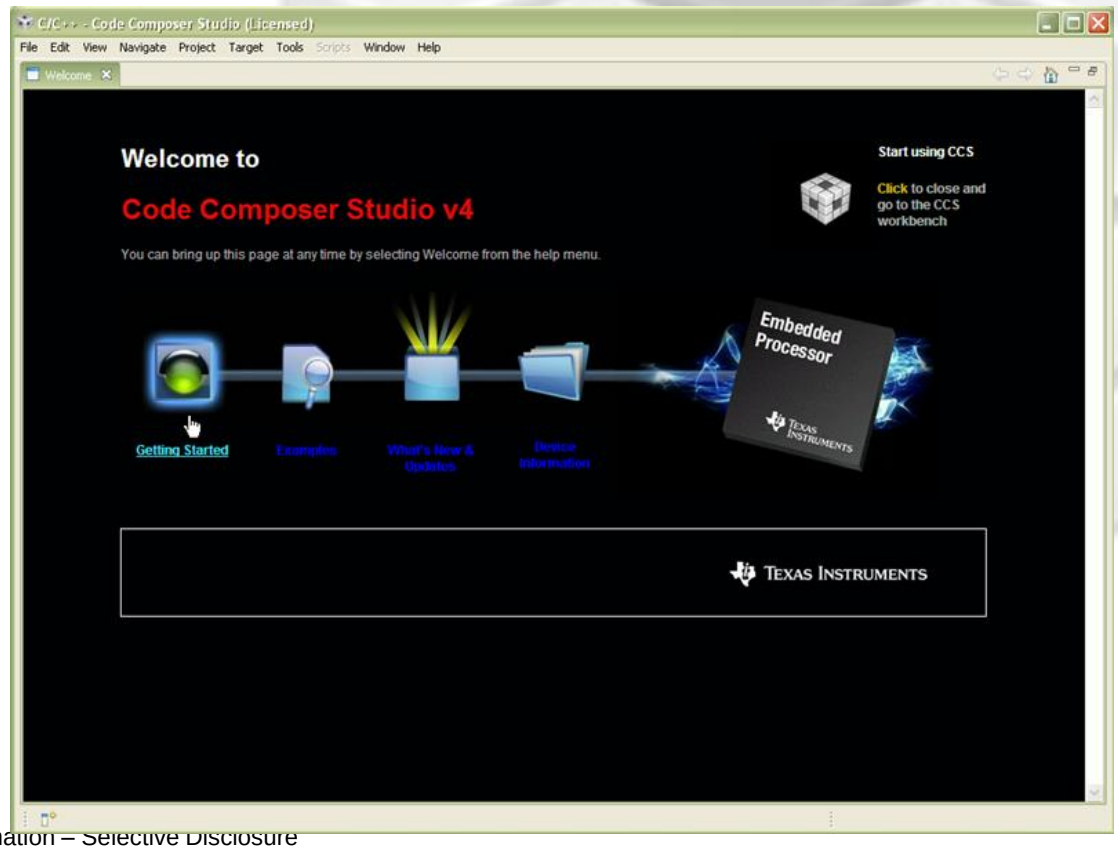


Eclipse Concept: Workspaces

- **Main working folder for CCSv4**
- **Contains information to manage all the projects defined to it**
  - The default location of any new projects created
- **User preferences, custom perspectives, cached data for plug-ins, etc all stored in the workspace**
- **Workspaces are not to be confused with CCSv3 workspace files (*.wks)**
- **Multiple workspaces can be maintained**
  - Only one can be active within each CCS instance
  - The same workspace cannot be shared by multiple running instances of CCS

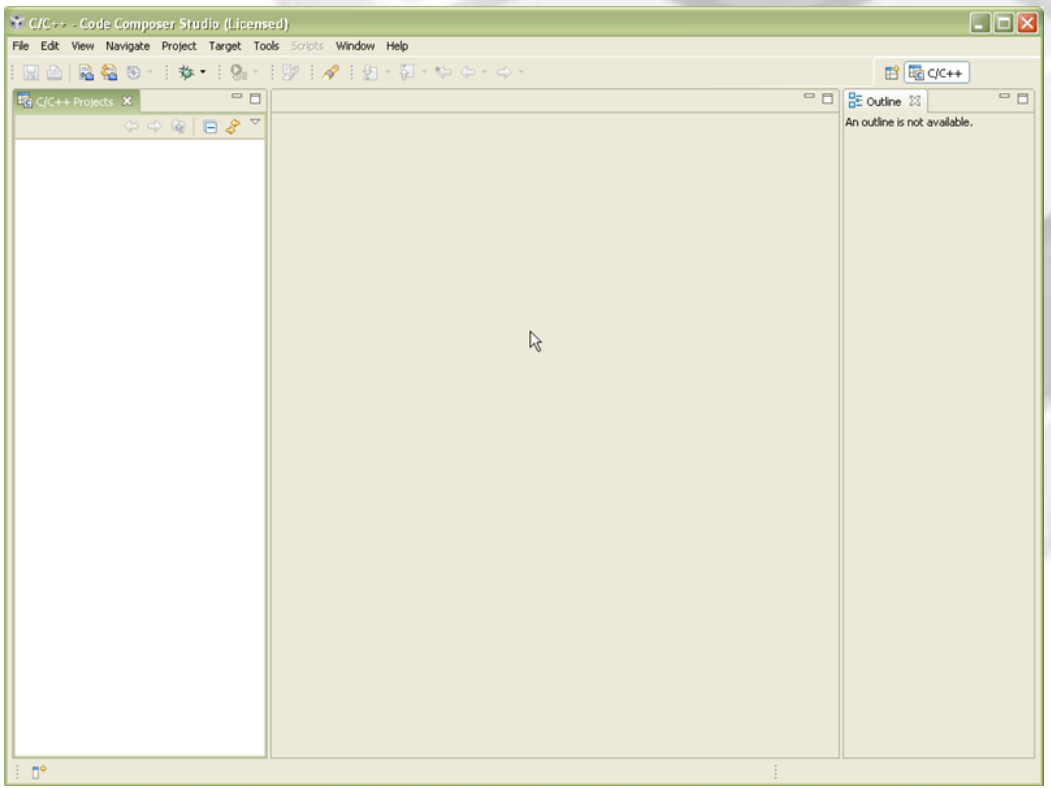
Welcome Page

- The ‘Welcome Page’ will be displayed the first time CCS is used with a new workspace
- Contains links to documentation, examples, support resources
- ‘Help->Welcome’ to return to the Welcome Page



Eclipse Concept: Workbench

- **'Workbench' refers to the main CCSv4 GUI window**
  - Equivalent to the 'Control Window' in CCS 3.x
- The Workbench contains all the various views and resources used for development and debug

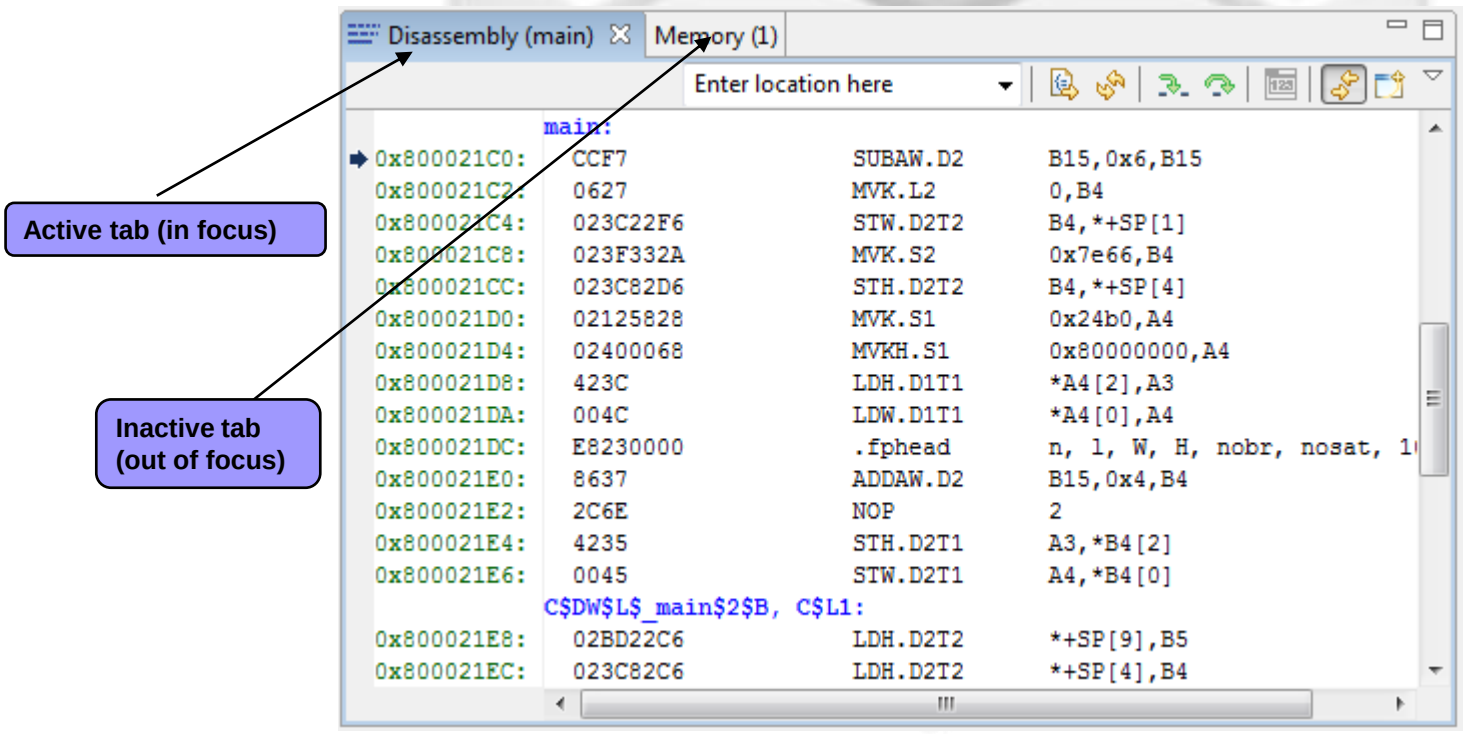


Workbench

- **CCS 3.x**
  - Only one Control Window can be opened for each debuggable CPU
  - Information is not shared between each Control Window
- **CCS 4.x**
  - Multiple Workbench windows can be opened ('Window->New Window')
  - Each Workbench window can differ visually (arrangement of views, toolbars and such), but refer to the same workspace and the same running instance of CCSv4
    - A project is opened from one Workbench will appear to be open in all the Workbench windows

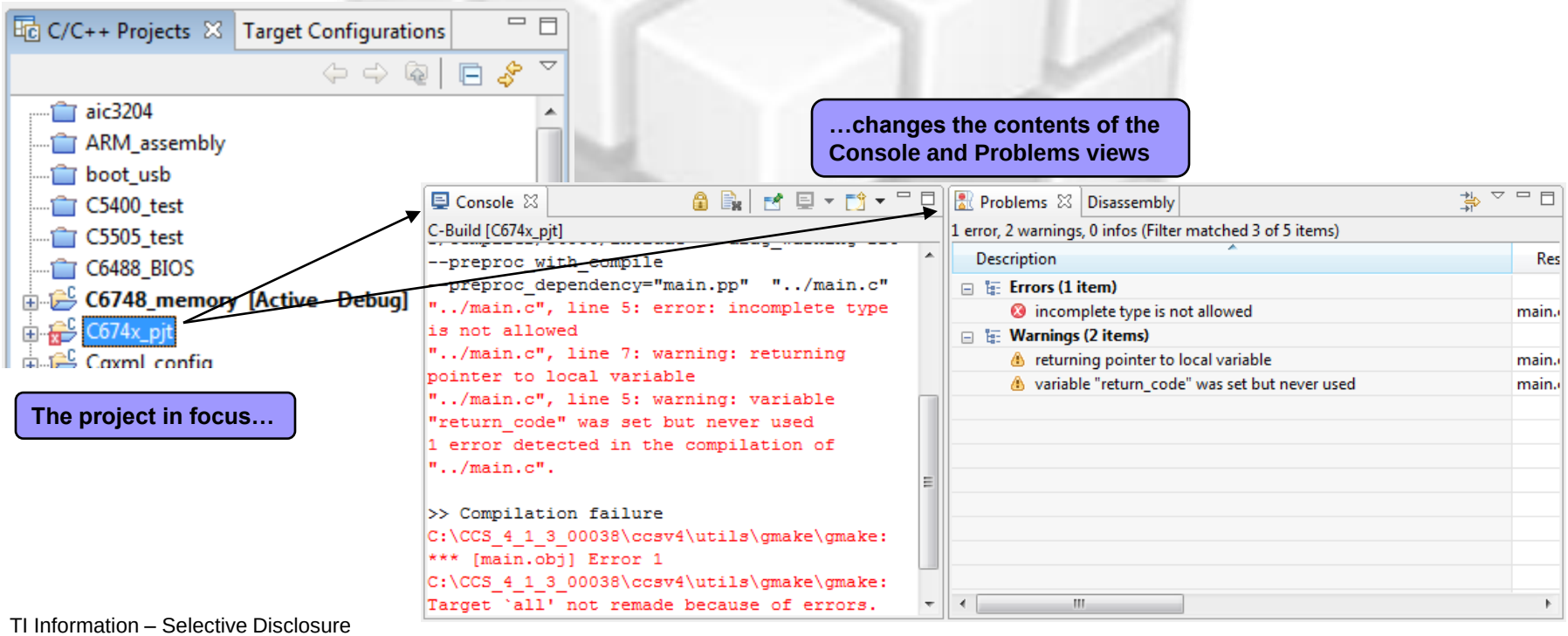
Eclipse Concept: Views

- **Views are windows within the main Workbench window that provide visual representation of some specific information.**
  - Most views can be accessed in the menu View
  - Can be identified by the organization in tabs



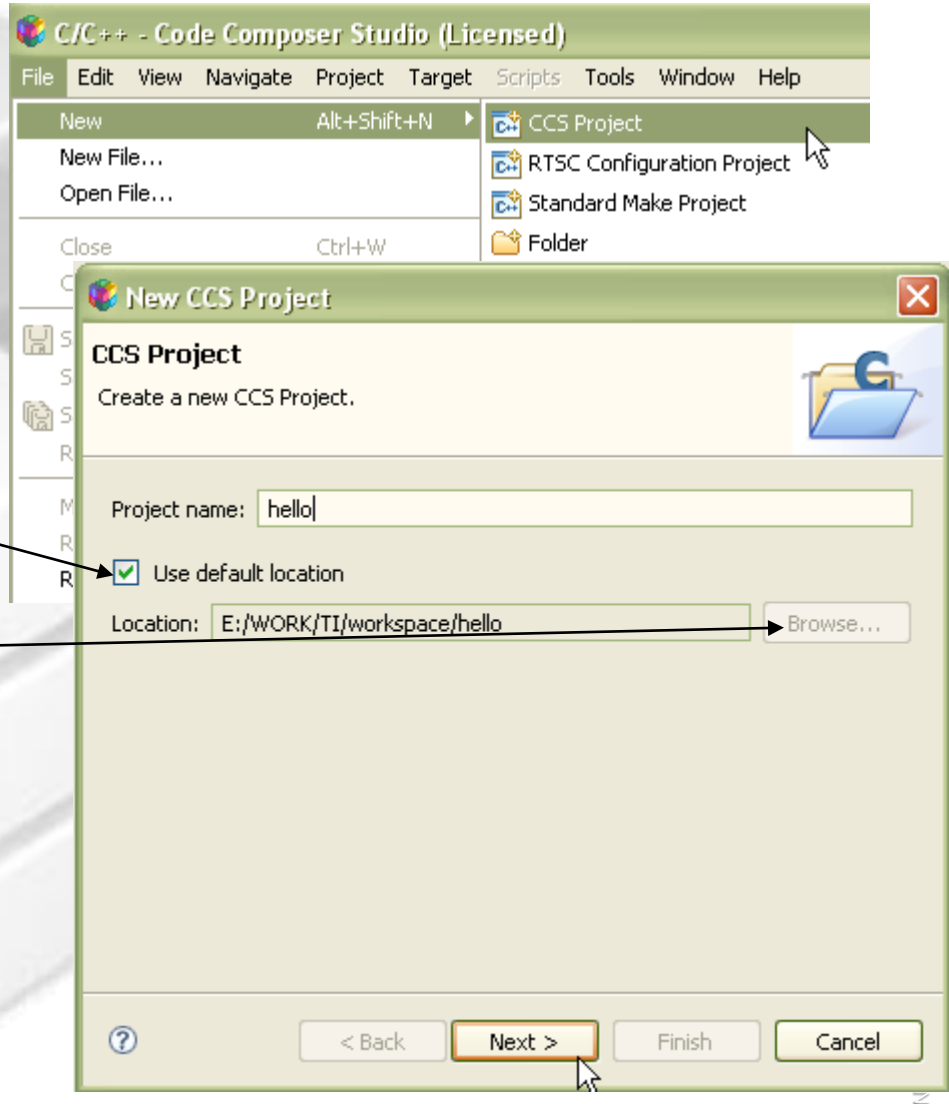
Eclipse Concept: Focus

- **Focus refers to the highlighted portion of the workbench. Can be an editor, a view, a project, etc.**
- **This concept is important since several operations inside Eclipse are tied to the element in focus**
  - Project build errors, console, menu and toolbar options, etc.



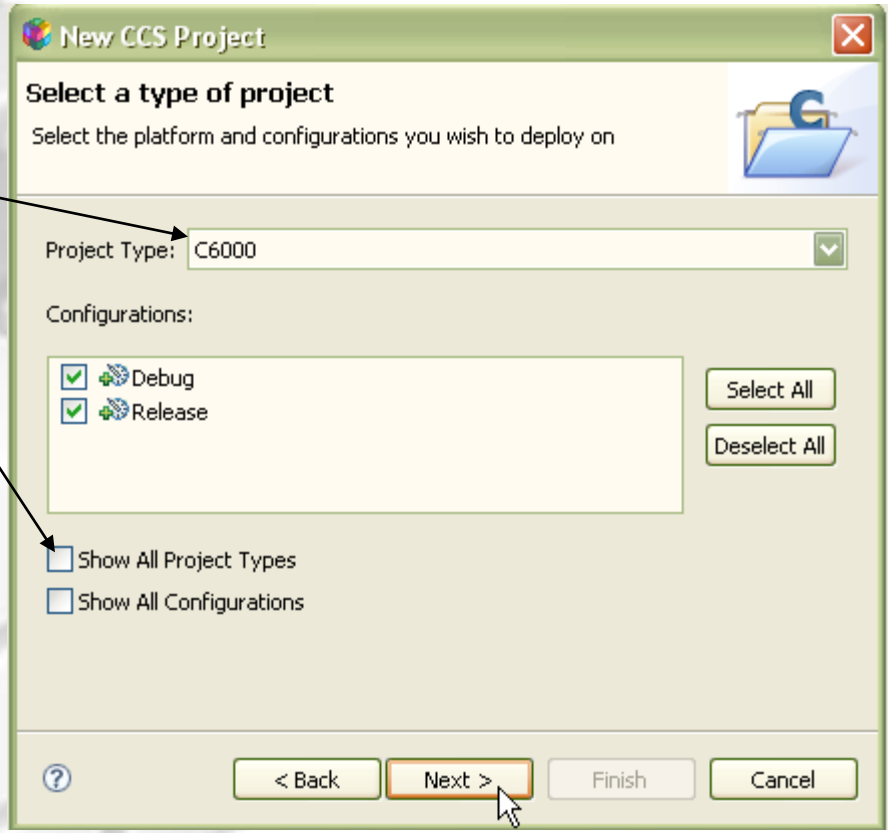
Project Wizard: Name/Location

- **Launch New Project Wizard**
  - File -> New -> CCS Project
- **Enter a name for the project**
  - This will create a folder of the project name in the workspace if the default location is used
  - Uncheck the ‘Use default location’ checkbox’ and then use the ‘Browse…’ button to place the project folder in a custom location
- **Select ‘Next’ when done**



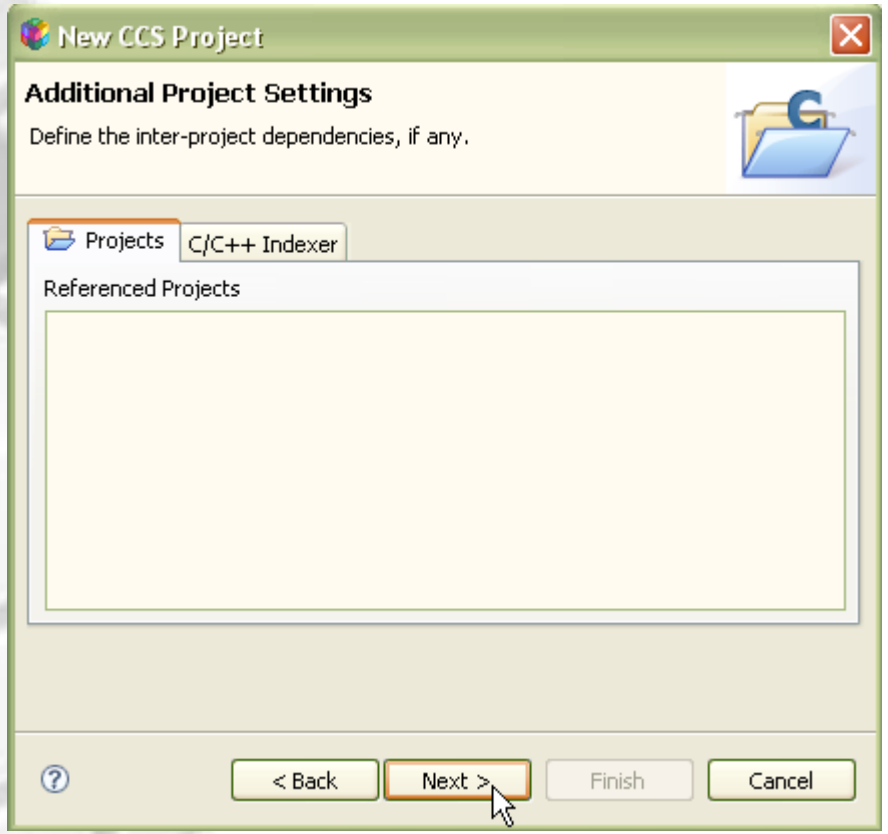
Project Wizard: Platform/Configuration

- **Select the Project Type**
  - Use the drop-down menu to specify platform to build for
  - Enabling the ‘Show All Project Types’ will display more options supported by Eclipse (not recommended)
- **Select build configurations to support**
  - Debug and Release are supported by default
  - Enabling ‘Show All Configurations’ will add a ‘Default’ configuration option
- **Select ‘Next’ when done**



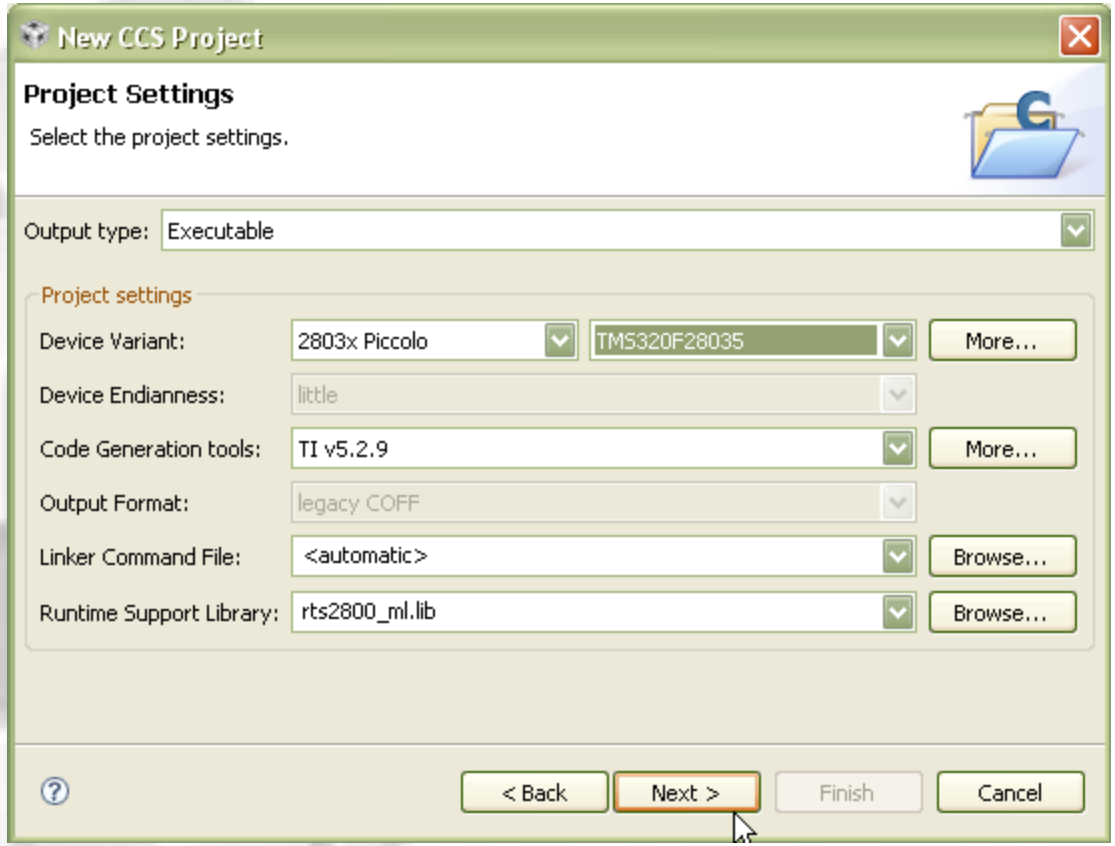
Project Wizard: Project Dependencies

- Other existing projects in the workspace will appear under ‘Referenced Projects’ and can be selected to create project dependencies
- C/C++ Indexer tab allows for changing the default indexer option (recommended to leave as default)
- Select ‘Next’ when done



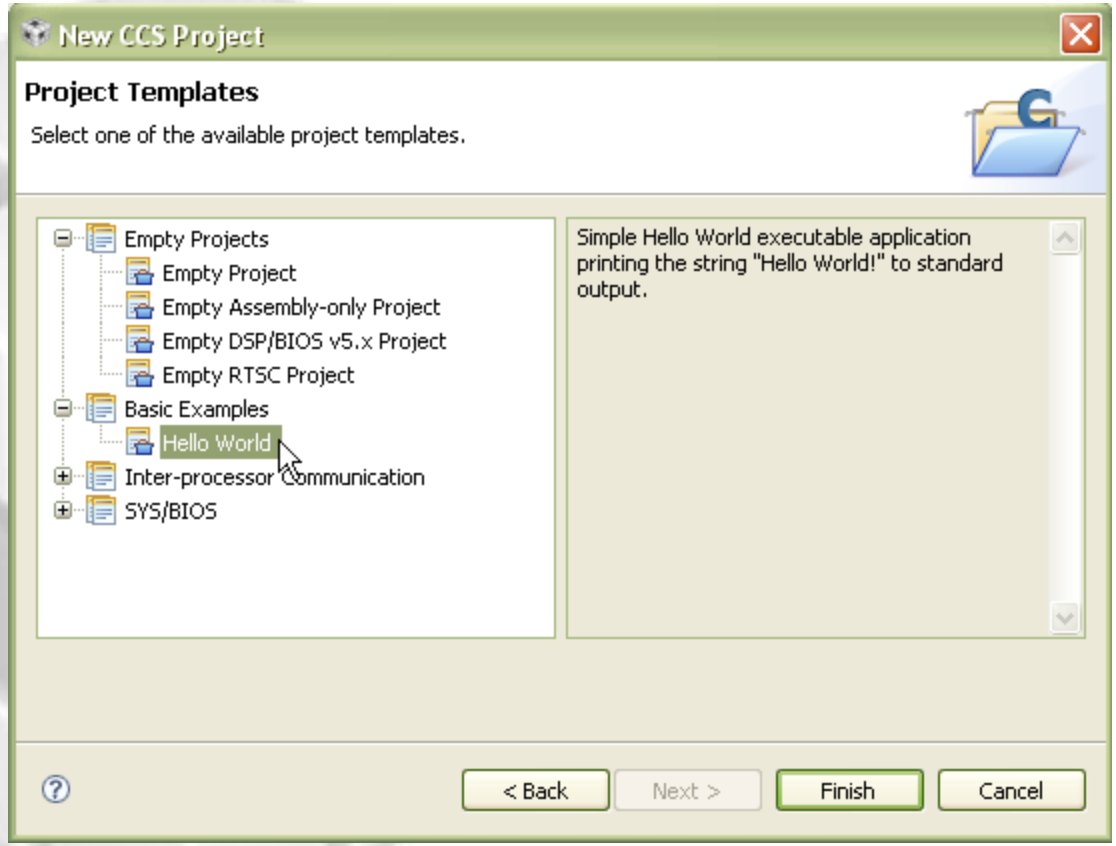
Project Wizard: Project Settings

- **Use menus to select:**
  - Project Kind (executable, library)
  - Device Variant
  - Endianness
  - CGT Version (configurable if CCS has been configured to use standalone CGT installs)
  - Linker *.cmd File (Linker will use default linker placement settings if left blank)
  - RTS Library
- **Select 'Next' when done**



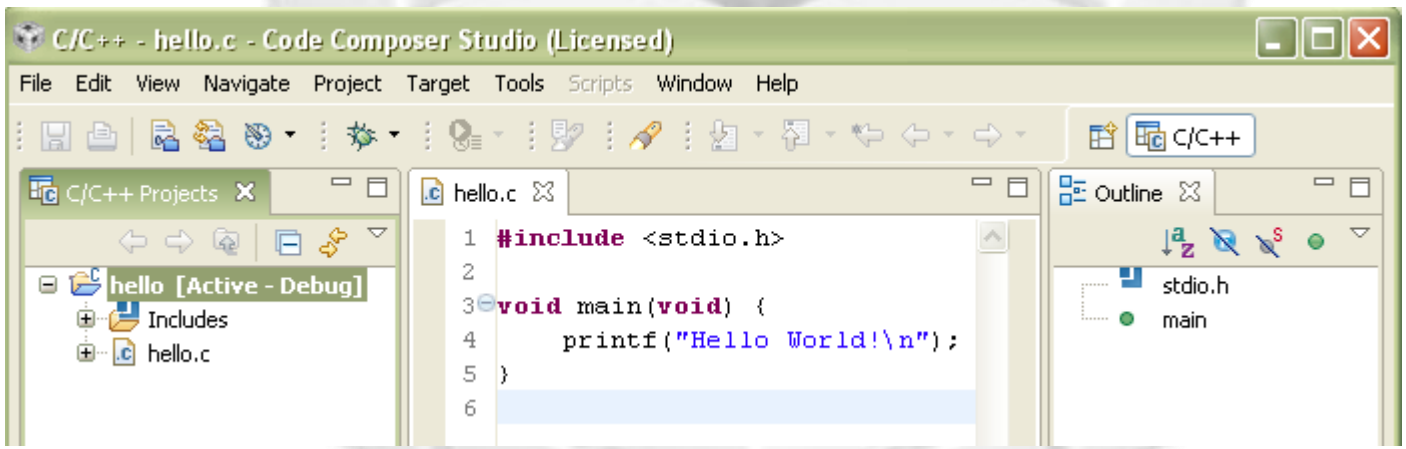
Project Wizard: Project Templates

- **Various project templates are available to help get started**
  - Hello World
  - SYS/BIOS examples
  - etc
- **Select 'Finish' when done**



Hello World: Projects

- Once the project is created, a reference to it will be made in the workspace and the project is now available for use within the Workbench and visible from the 'C/C++ Projects' view

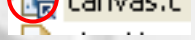


Eclipse Concept: Projects

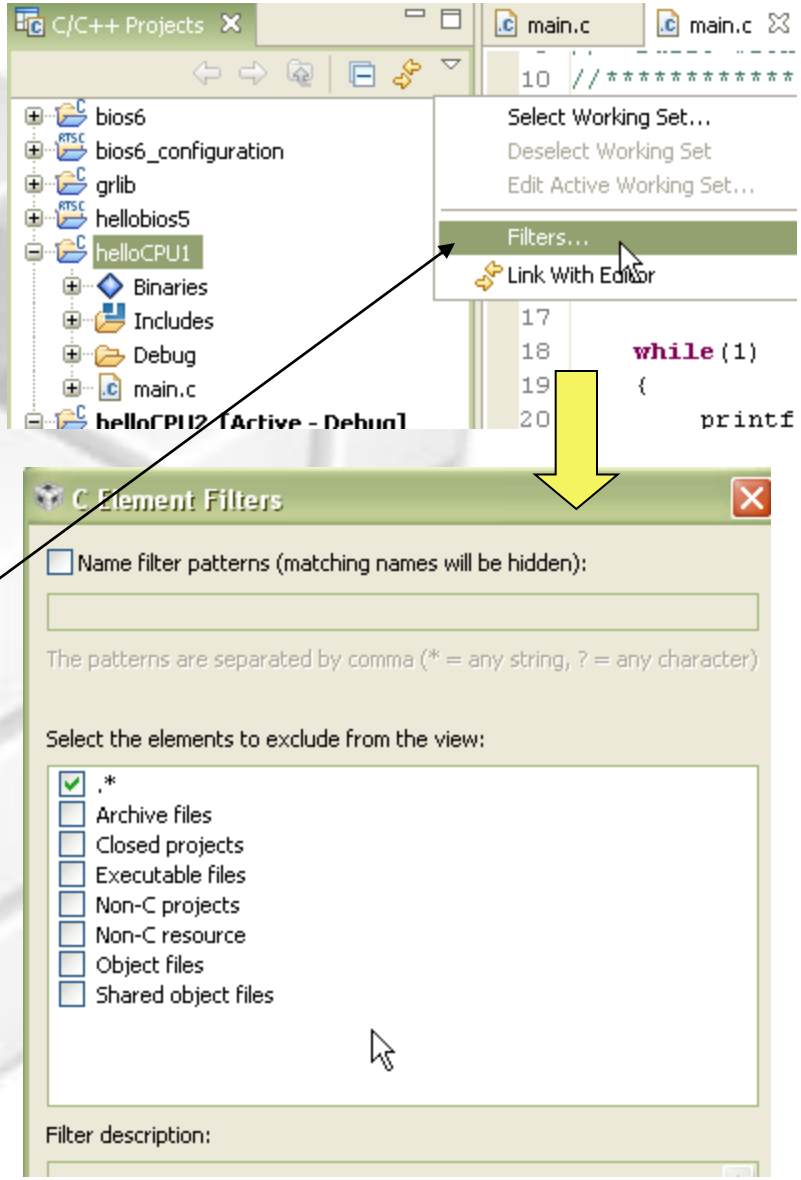
- **Projects map to directories in the file system**
- **Files can be added or linked to the project**
  - Adding file to project (Eclipse default behavior)
    - Copies the file into your project folder
  - Linking file to project
    - Makes a reference to the file in your project
    - File stays in its original location
    - CCS 3.x project files used this concept exclusively
- **Projects are either open or closed**
  - Closed Projects:
    - Still defined to the workspace, but it cannot be modified by the Workbench
    - The resources of a closed project will not appear in the Workbench, but the resources still reside on the local file system
    - Closed projects require less memory and are not scanned during routine activity
    - This differs from CCS 3.x, where closed projects do not appear at all in the CCS 3.x project view window.
- **Projects that have not been defined to the workspace must be imported into the active workspace before they can be opened**
  - Both CCSv4/CCE projects and legacy CCSv3 projects can be imported into CCSv4

View: C/C++ Projects

- **Displays all projects defined in the active workspace**
- **The view is mostly a representation of the file system of the project folder**
  - Linked files are marked with a special arrow graphic in the icon



- **Use filters to hide various file types to reduce clutter in the view**
  - Default is to filter CCS generated project files (.*)



Hello World: Launching Summary

- **Add a new target configuration to the project**
  - Right click, select New->Target Configuration File
  - Configure for the target you like
    - Set-up a simulator of the correct ISA and endianness
- **Launch a debug session**
  - Click the debug button at the top
- **The debug perspective should open**
  - It can be switched back in the upper right corner
- **The program should be halted at ‘main’**

Target Configuration Files

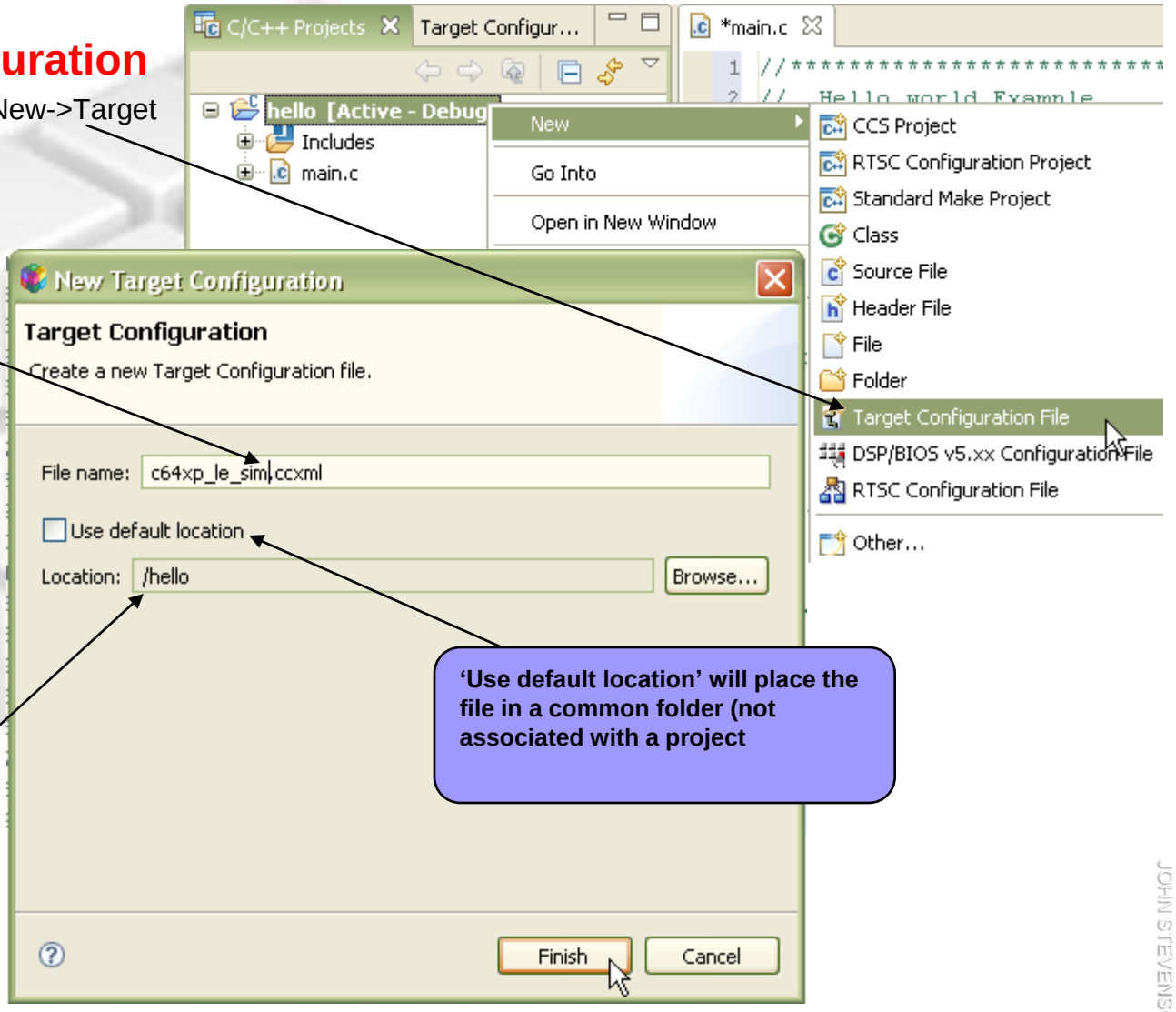
- **Target Configuration files are xml files that define the connection and device (have a file extension *.ccxml)**
  - Equivalent of the CCS 3.x configuration file (*.ccs extension)
- **'Target Configurations' view is used to manage and work with target configuration files**
- **Target configuration editor is used to create/modify target configuration files**
- **Basic tab is intended to be used by majority of end users**
- **Advanced tab is intended to be used for adjusting default properties, initialization scripts or creating target configurations for new boards/devices**

Target Configuration Files

- **CCS 3.x**
  - Use separate tool called ‘CCSetup’ to create configuration files and configure CCS to use a specific file
  - Can only specify a single configuration file at a global level
  - Must close CCS and relaunch CCSetup tool to configure CCS for a new configuration
- **CCS 4.x**
  - Tool for creating configuration files are integrated with CCS (no separate tool to launch)
  - Multiple configurations can exist and can start a CCS debug session by choosing one from a list
  - Can add a target configuration file to a specific project (‘Debug Active Project’ will automatically configure CCS for the target specified in the configuration file and launch a debug session for it

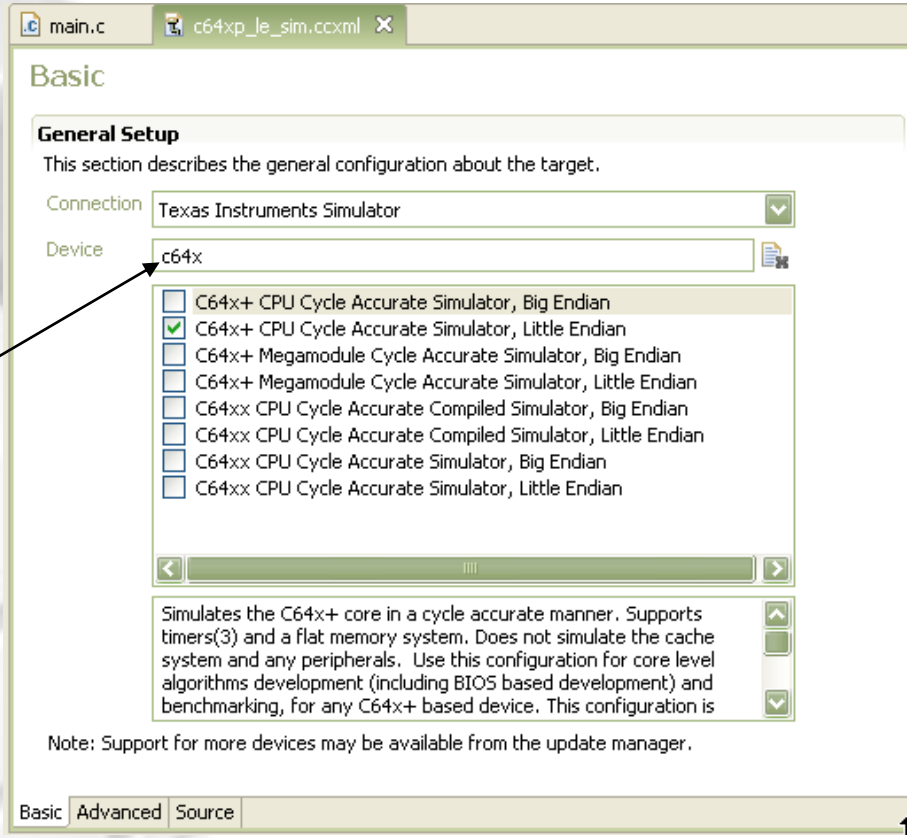
Create a Target Configuration

- **Add a new target configuration**
  - Right click on project and select New->Target Configuration File
- **Provide a name for the new target configuration file**
  - Multiple target configuration files can be created hence it is recommended to use a descriptive name
- **Specify the location of the target configuration file**
  - The location will be the project folder if the Project view is active



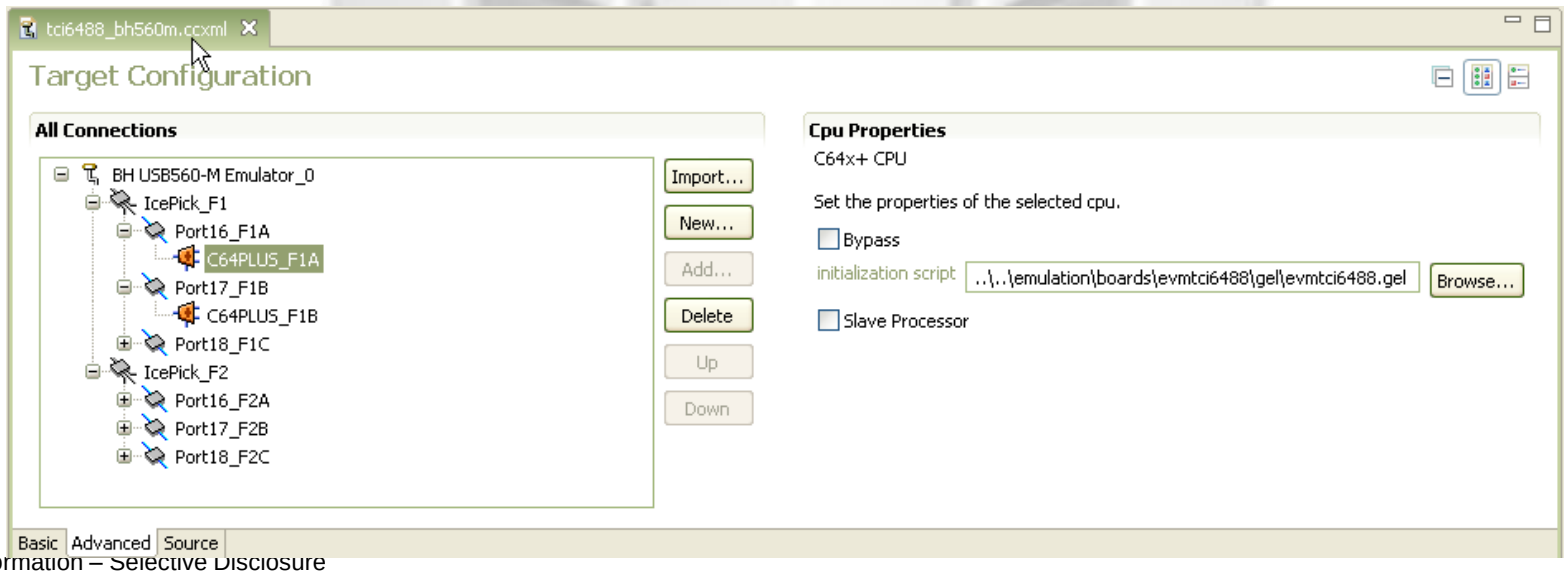
Create a Target Configuration

- **The new target configuration file will be opened in the editor**
- **Use drop down menu to select the connection type (simulator, emulator, etc)**
- **Select the device**
  - Enter characters in the filter field to narrow down the list of devices
- **Save the file when done**



Target Configurations - Advanced

- **Adjust default properties of the target configuration:**
  - Specify initialization files (GEL startup files)
  - Specify IcePick subpath port numbers
  - Bypass CPU
  - Set JTAG TCLK Frequency
  - etc…
- **Similar to the CCS 3.x CCS Setup utility**

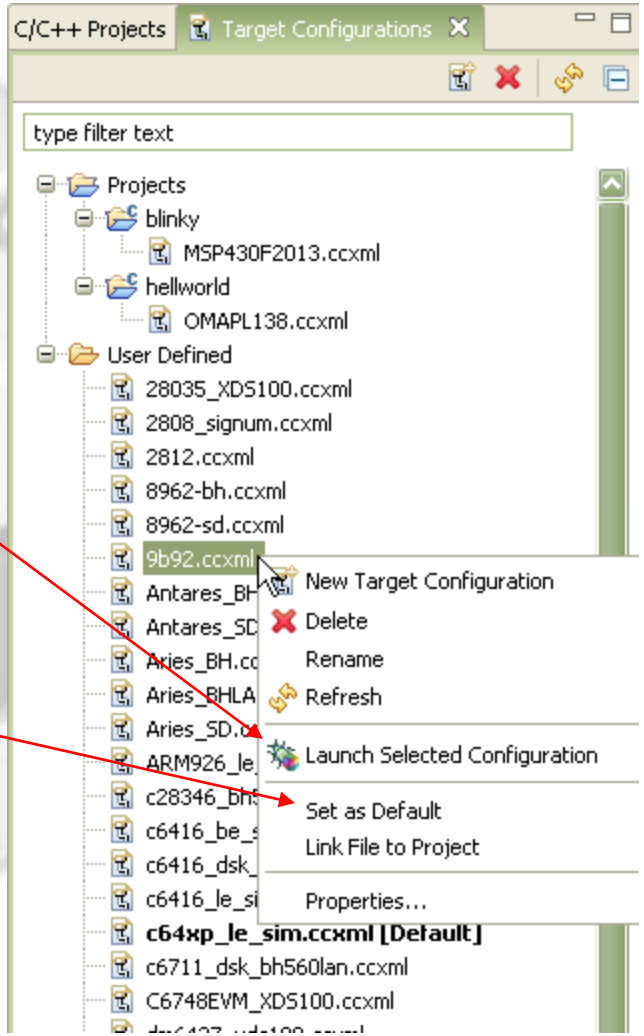


Target Configurations - Advanced

- **Use ‘Advanced Setup’ utility if option for your emulator/target is missing from the ‘Basic’ tab**
  - Select from a list of available 'Connections' to specify the connection type
  - Then select from the list of components ('Devices', 'CPUs', 'Routers') to add to the connection
- **Use the ‘Advanced Setup’ to create a single target configuration using two emulators**
- **Must have good knowledge of the device to correctly use 'Advanced Setup' utility to build a configuration**

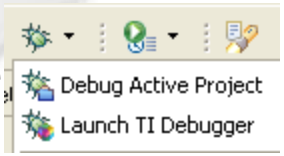
Target Configurations View

- **Target configurations easily deleted, copied and opened for inspection (XML files)**
- **Launch a debug session quick: right-click on target configuration and select 'Launch Selected Configuration'**
- **'Launch TI Debugger' will use target configuration that is identified with** [Default] **tag in Target Configurations View**
  - Right click on a file and select 'Set as Default' to make it the default
  - 'Debug Active Project' will also use the 'Default' if there is no target configuration file in the project



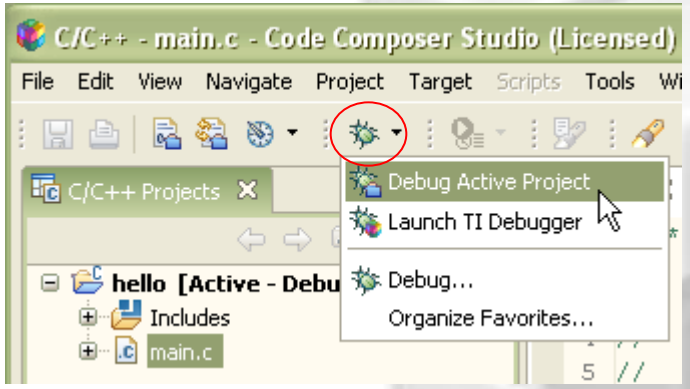
Launching the Debugger

- **Debug Active Project**
  - Automated debugger launch:
    - Incrementally builds project
    - Launches a debug session for associated target configuration
    - Connects to appropriate target in your system setup
    - Loads program
    - Runs to main (optional)
- **Launch TI Debugger**
  - Manual debugger launch (no other actions performed) for the default configuration
  - Also launched from the Target Configuration view context menu: 'Launch Selected Configuration'
  - Debug session has no project association
    - This is similar to the behavior of CCS 3.x
- **Automatic perspective switch**
  - CCS switches to the debug perspective when the debugge (configurable)

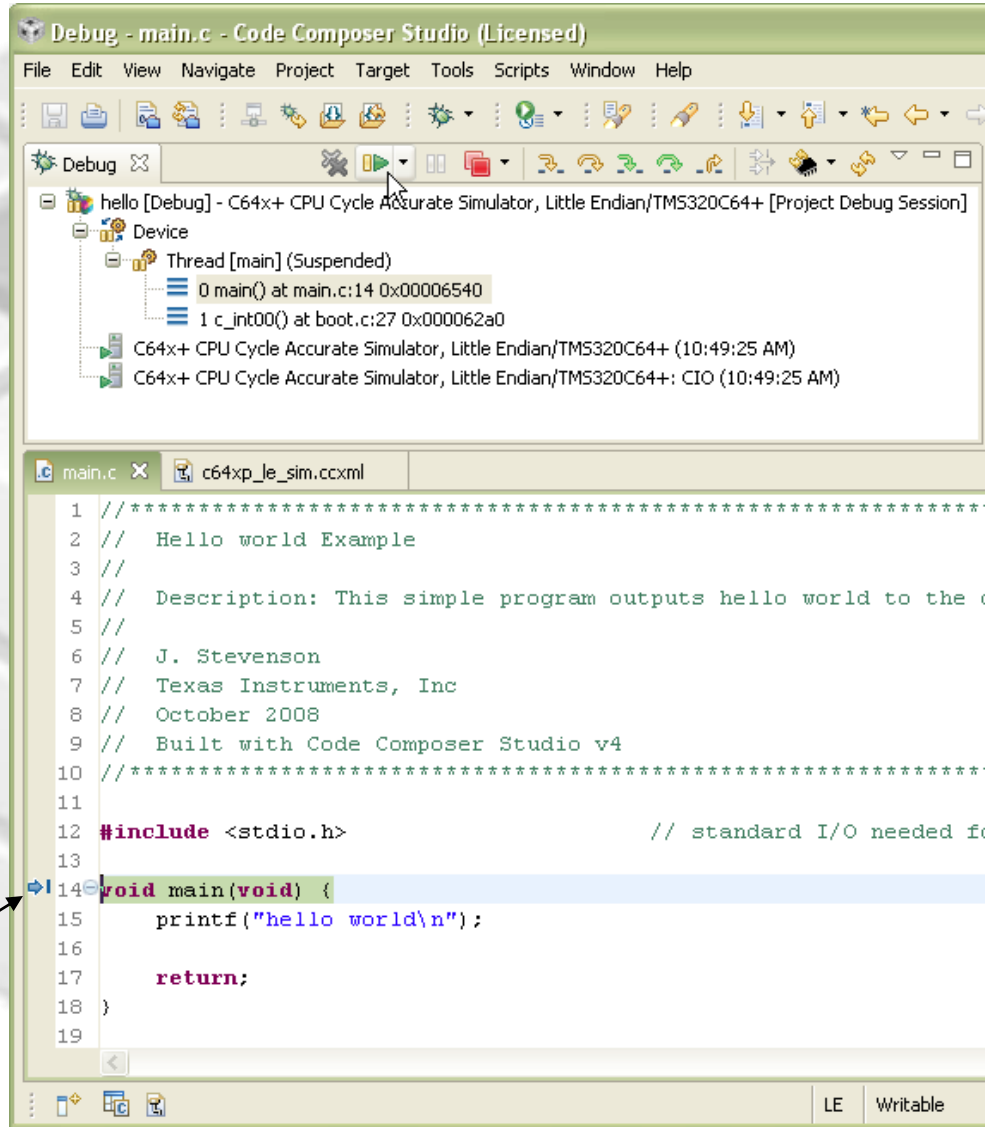


Build and Launch a Debug Session

- **Select 'Debug Active Project' menu to build the project and launch a debug session**

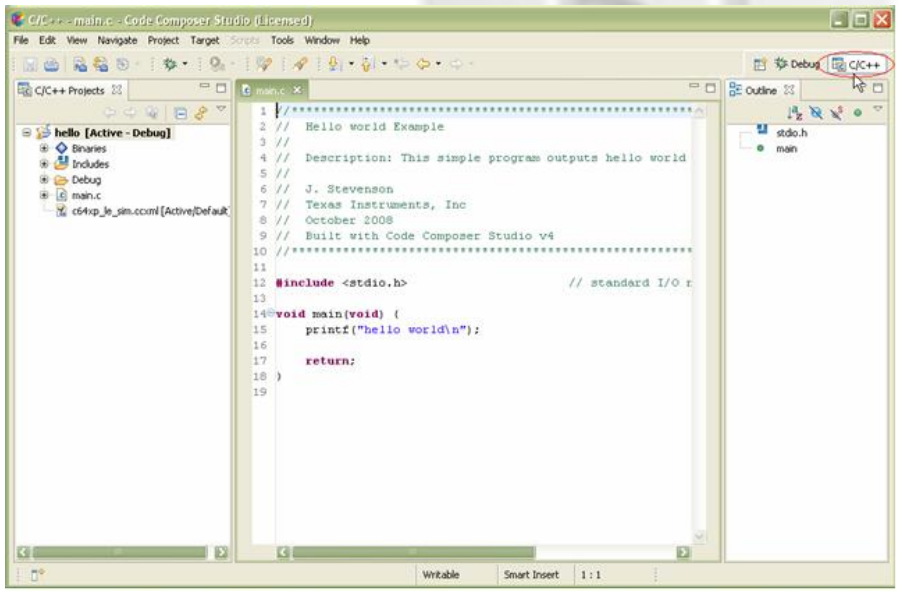


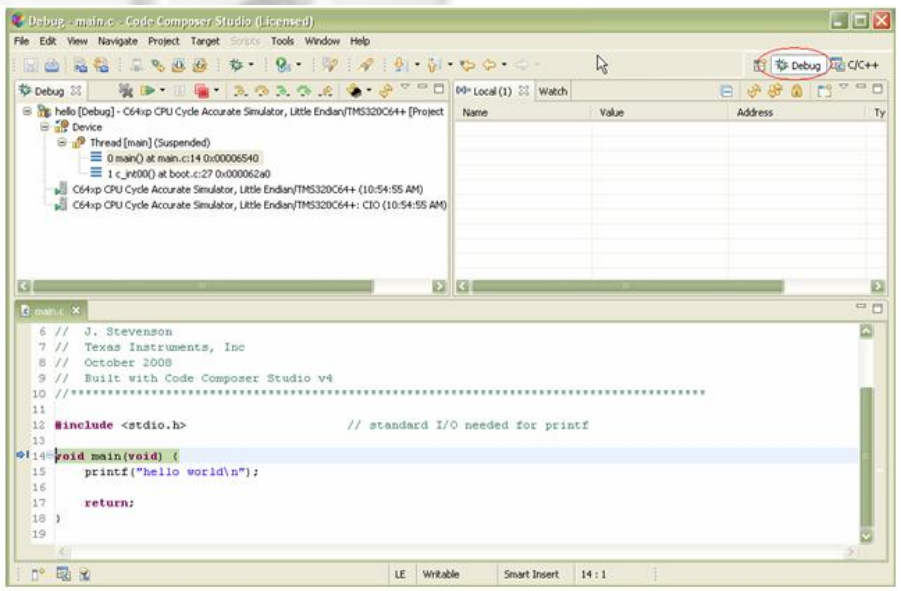
- **CCS will switch to the Debug Perspective**
- **The program should be halted at main()**



Eclipse Concept: Perspectives

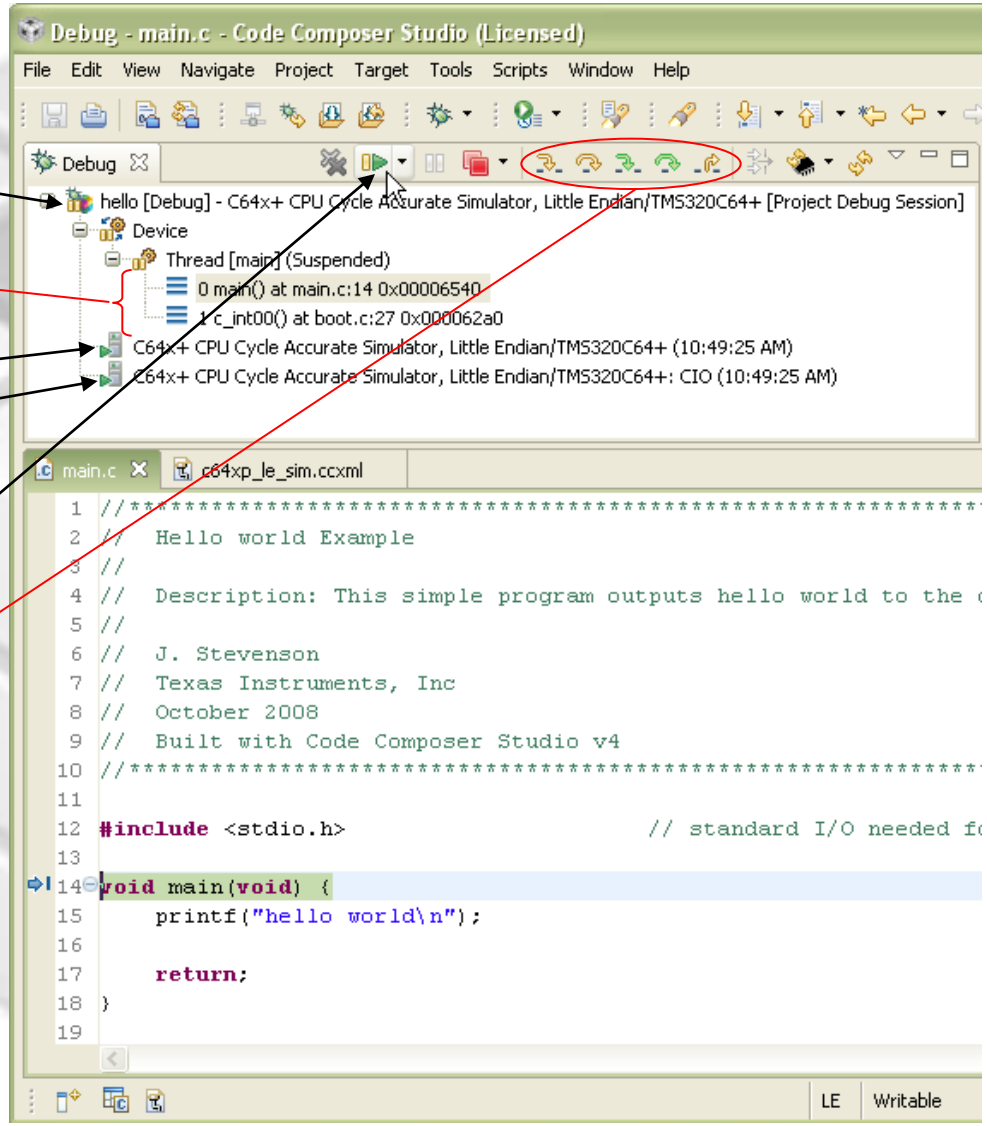
- Defines the initial set and layout of views in the Workbench window
- Similar in concept to CCSv3 'workspaces' (*.wks) except that multiple perspectives are available from the Workbench window (though only one can remain active at a time)
- Each perspective provides a set of functionality aimed at accomplishing a specific type of task (‘C/C++’ for project development, ‘Debug’ for debugging, etc)





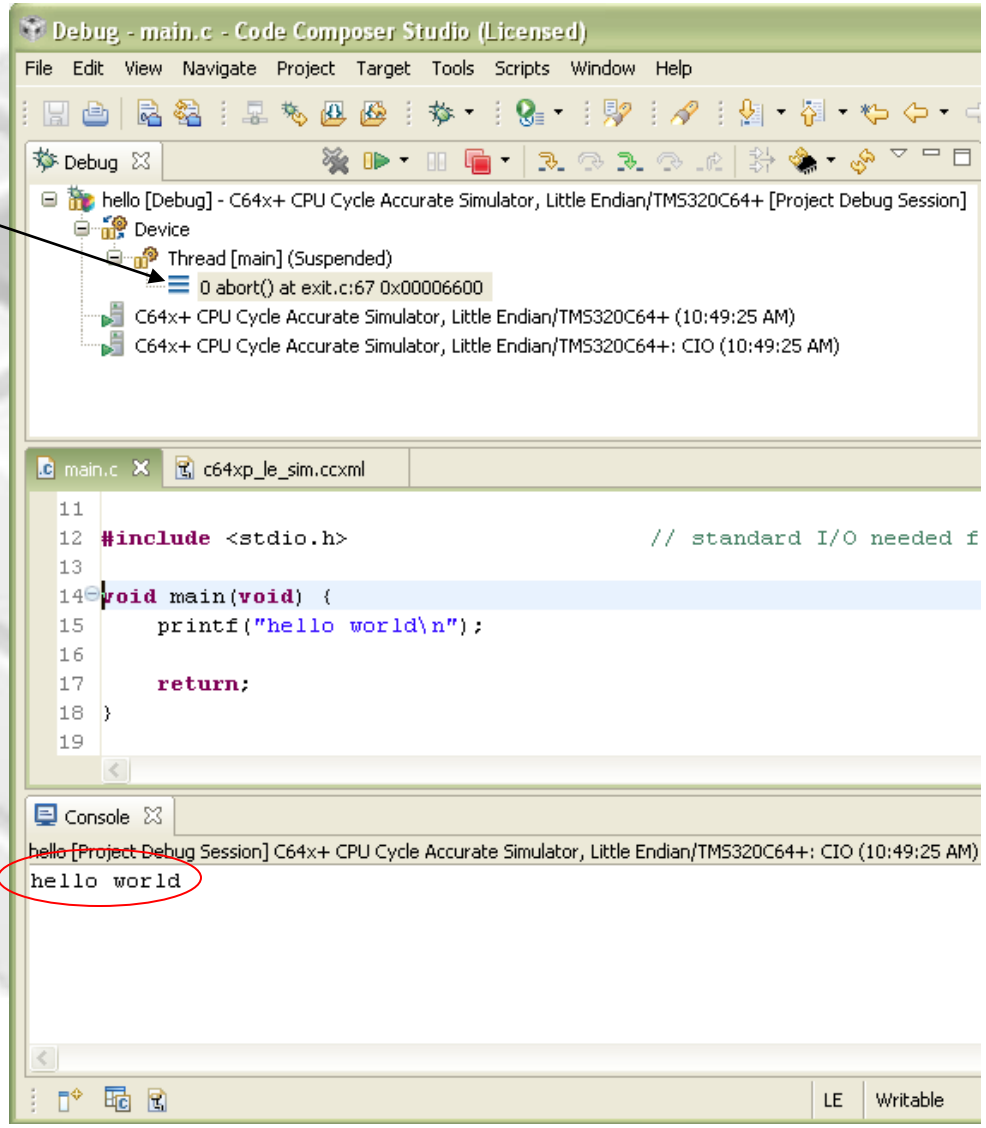
Debug Perspective

- **Debug view displays:**
  - Target configuration
  - Call stack
  - Associated console views
    - Standard console
    - CIO output
- **Select the 'run' button to execute application or use single stepping buttons to step through execution**



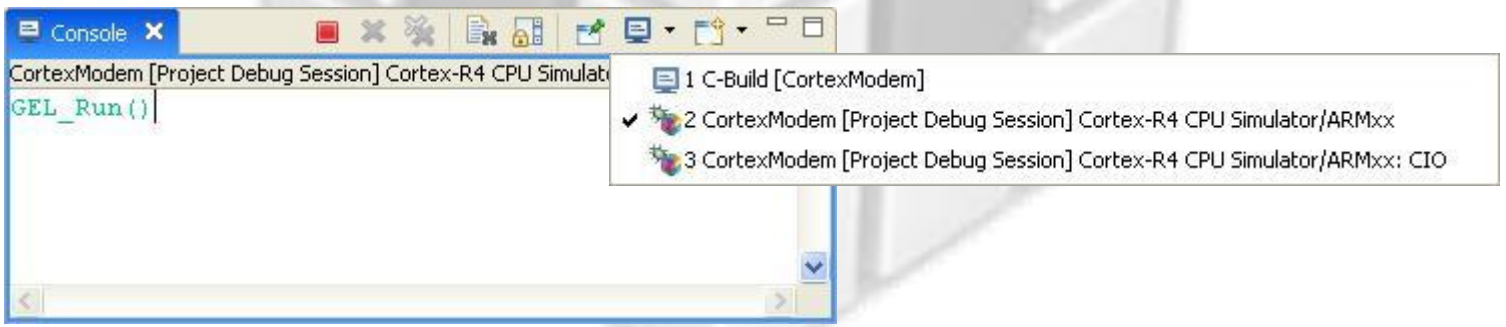
Debug Perspective

- Program halted at exit point
- “hello world” printed to CIO console (Console will automatically open)



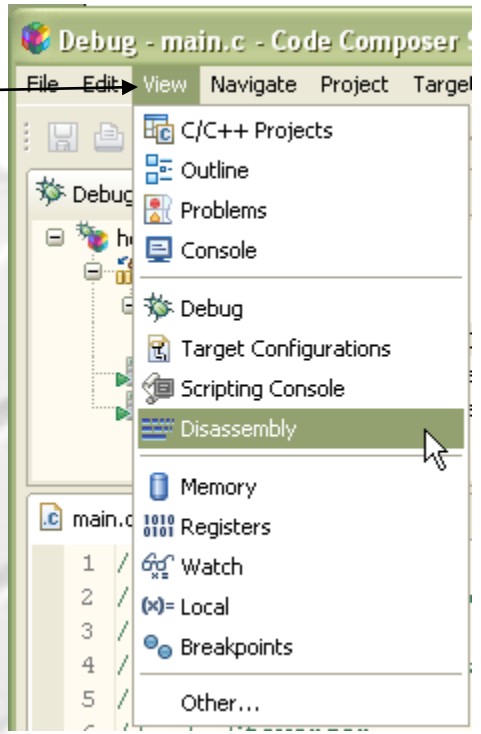
View: Console

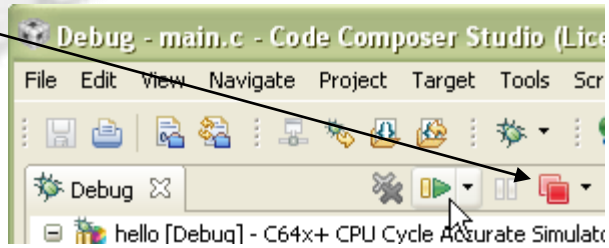
- **Multiple contexts**
  - When the CPU is selected it operates as a GEL command interface to the debugger
  - When CIO is selected it shows CIO output
  - Automatically switches contexts
    - Can use the “pin” option to prevent this
  - CCS 3.x had separate, dedicated views for build output, CIO output and GEL output and a separate GEL Toolbar for GEL command interface
- **You can open multiple console windows**
  - Printfs in one and command interface in another



More Debugging

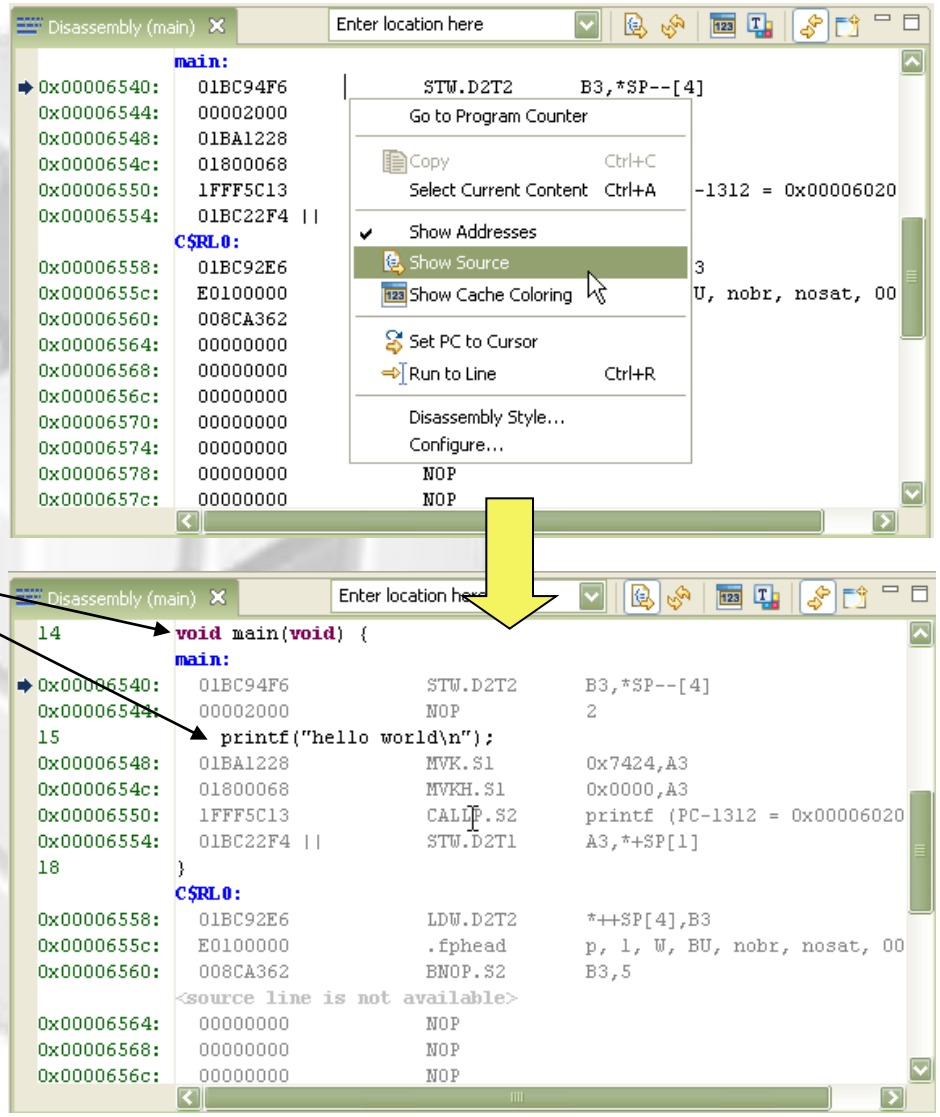
- **Investigate other debugging views (Open via 'View' menu)**
  - Memory View
  - Locals and Watch views
  - Register view
  - Disassembly (see next slide)
- **Set breakpoints**
  - Double click on a source line to set/clear
- **Terminate the session**
  - Red "stop" button in the debug view
    - This will end the debug session
  - The perspective will switch back to the 'C/C++ Perspective'





View: Disassembly

- Reload the program (Target->Reload Program) and open the disassembly view (View->Disassembly)
- Right click in the Disassembly view and select show source
- Note the interleaved source with the disassembly



View: Breakpoints

- **View all available breakpoints**
- **Can group breakpoints by CPU (multicore device)**
- **Specify various actions when the breakpoint is triggered**
  - Refresh All Windows or update a specific view (replaces “Animate” in CCS 3.3)
  - Control Profiling (set profile halt/resume points)
  - File I/O (Probe Points in CCS 3.3)
  - Run a GEL expression