

# Getting Started with CCS and C2000Ware

## Software Development for C2000 Devices

Using C2000Ware with TI's Code Composer Studio (CCS)

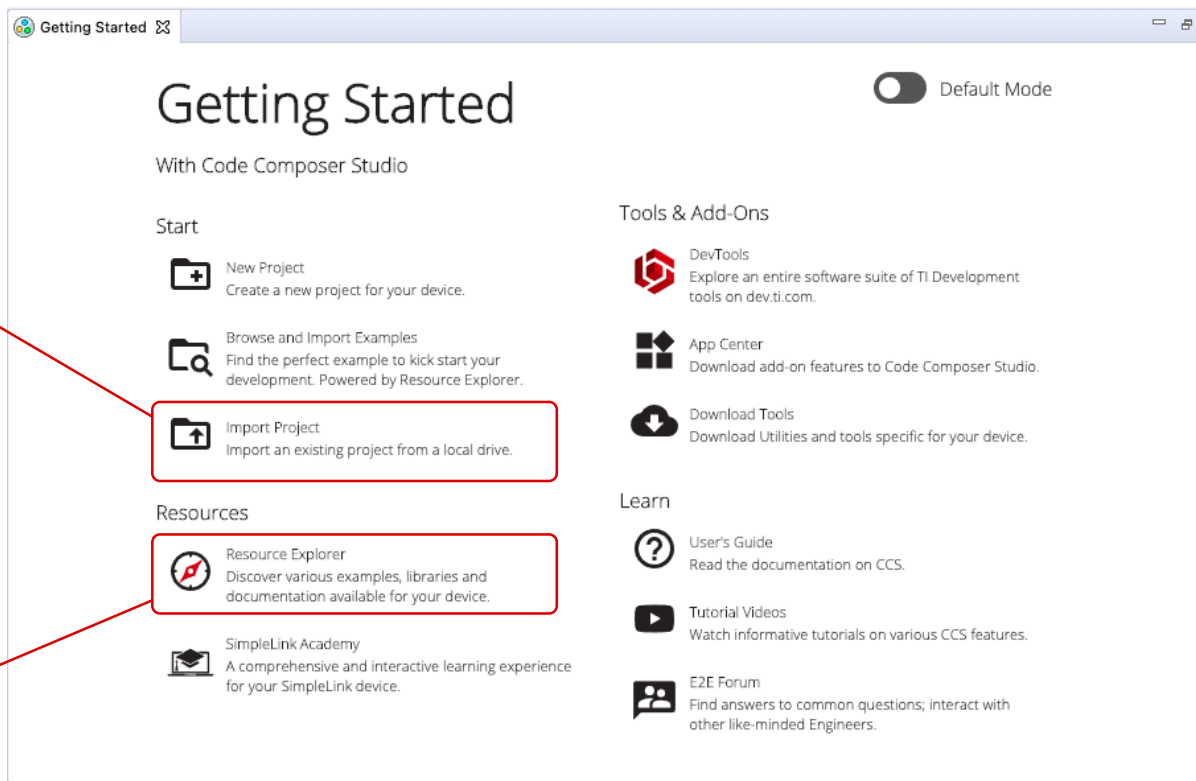
# What is Code Composer Studio?

- **Integrated Development Environment (IDE) for TI embedded processors**
  - ❖ IDE built on existing system frameworks
  - ❖ Extended by TI to support TI device capabilities
  - ❖ Compatible with Windows, Linux, and macOS operating systems
- **CCS Is an All-in-One Development Tool**
  - ❖ Includes debugger, compiler, editor, profiling tools, etc.
  - ❖ Includes features like customizable syntax highlighting, code completion, and local history
  - ❖ Integrated with TIREX for easy access to examples, libraries, datasheets, and more
- **Integrate Additional Tools**
  - ❖ OS level application like TI RTOS (Real-Time Operating Systems)
  - ❖ Code analysis and source control

# What is C2000Ware?

- **Software Development Kit (SDK) for TI C2000 real-time MCUs**
  - ❖ Contains libraries and device support for C2000 devices
  - ❖ Repository of example code for working with C2000 peripherals
  - ❖ Board files and schematics for C2000 evaluation modules (EVMs)
- **Compatible with other TI tools, like CCS and SysConfig**
  - ❖ Couples with other programs and tools for improved development experience
  - ❖ Continually updated alongside other TI tools for up-to-date support
- **MotorControl SDK and DigitalPower SDK build on top of C2000Ware**
  - ❖ Separate software repositories for specific end applications
  - ❖ Updated continually to maintain device support and added capabilities
  - ❖ Interface with CCS similarly to C2000Ware

# CCS Getting Started View

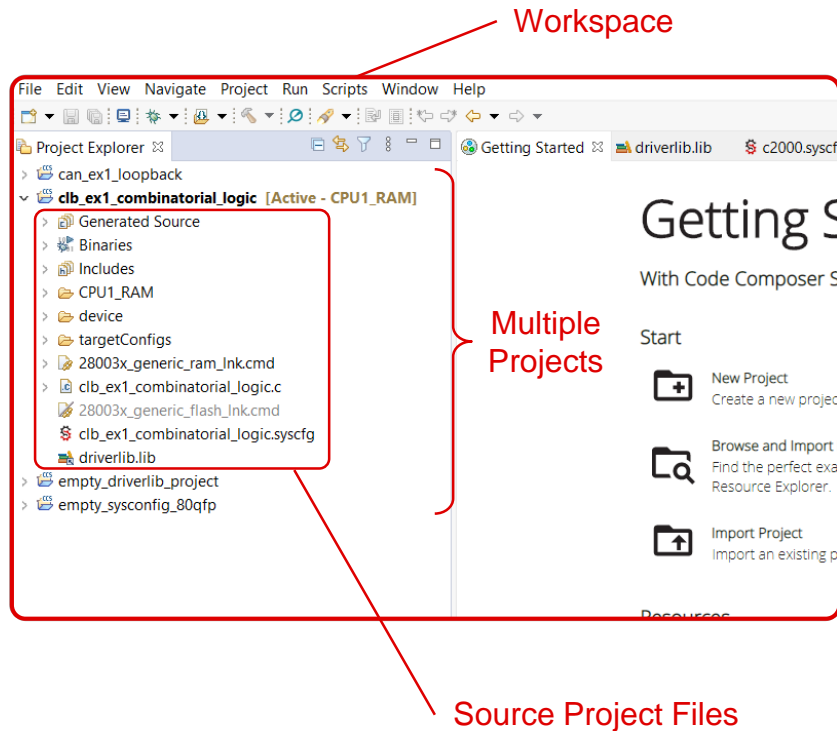


Import Example  
Using C2000Ware  
Download

Import Example  
Using TI Resource  
Explorer

# CCS Workspace and Project

- Workspace
  - A single space containing a single or multiple projects
  - Stores settings and preferences
  - Everything packaged into a single folder within user's CCS file directory
- Project
  - Houses all source files, header files, linker command files, etc.
  - Contains build and tool settings



# Target Configuration File (.ccxml)

- Describes aspects of debugging environment
  - Type of Debug Probe
  - Target Board or Device
  - GEL (General Extension Language) script
    - Performs device and hardware initialization
- Composed of connection files, board XML file, device XML file, and GEL file
  - Automatically created when creating a new project
  - Can be manually created and configured for custom device configuration
- Advanced configuration of JTAG settings such as clock speed & JTAG mode

# General Extension Language (GEL) File

- Allows for the automation of processes in CCS
  - Primarily used to initialize device or boards
    - OnTargetConnect()
    - OnReset()
    - OnRestart()
- GEL file needs to be included into CCS
  - Manually load and unload GEL file
  - Automatically loaded via target configuration file (during debug)
- Builds upon CCS by adding executable functions to the IDE's menu bar

# Connecting C2000 Devices

## Connecting Target with Host Machine

- LaunchPad with onboard debugger
- controlCARD
  - Onboard debugger
  - External emulator (XDS110, XDS200, etc.)

**Host**  
(Code Composer Studio)



**Target**  
(C2000 controlCARD)



USB cables

Onboard Debugger  
XDS100v2 or XDS110

**Host**  
(Code Composer Studio)



**Target**  
(C2000 Launchpad)



USB cable

Onboard Debugger  
XDS100v2 or XDS110

**Host**  
(Code Composer Studio)



**Target**  
(C2000 controlCARD)



5V


External Debugger  
XDS100v2 or XDS110 or XDS200

USB cable





# Debug Session

- Execute and debug project using CCS Debug Session
  - Launch automatically using  button
  - Launch manually by launching target configuration file and loading in project
- Run program, set breakpoints, and step through code
  - **Variables** – Local variables can be added to debug view and manually changed
  - **Watch Expressions** – Variables, expressions, and registers can be monitored
  - **Registers** – View and edit contents of core and peripheral registers
- Debug view contains target configuration and call stack for each core
  - **Disassembly** and **Memory Browser** panes used to examine device data during execution

# Additional CCS and C2000Ware Resources

- Download [Code Composer Studio](#)
- Download [C2000Ware](#)
- Test out CCS Cloud and Resource Explorer at [TI DevTools](#)
  
- Learning Material
  - [Code Composer Studio Documentation Overview](#)
  - [Code Composer Studio User's Guide](#)
  - [C2000Ware Quick Start Guide](#)
  - [C2000 Academy](#) with Hands-on Labs

Check Video Description for Additional Resources