EVE Getting Started

Prerequisites & Expectations

Prerequisites
Windows PC available including internet connection

Expectation
How to install and begin working with Code Composer Studio and EVE Simulator

Abstract

- EVE Simulator Installation process
  - This presentation shows the installation process of Code Composer Studio (CCS) and EVE simulator to begin working with EVE programming.
- Documentation and Links
  - It will reference to all documentation required in order to develop and test EVE software.
- Using EVE Simulator
  - The presentation will guide you through the use of EVE simulator under CCS and show how to import and debug first examples.

EVE Documentation

EVE Getting Started

EVE documentations

Latest documentation on CDDS (includes EVE Programmers Guide) :
https://cdds.ext.ti.com/ematrix/common/emxNavigator.jsp?objectId=28670.42872.18092.28443

Name	Content
VCOP CPU and Instruction set reference guide	Functional specification of the vector core: architecture, instruction pipeline, instruction set.
Embedded Vector Engine Programmer's Guide	Teach EVE programming from a high-level VCOP-C language and system perspective. Has examples for basic functions (point-to-point operation, filters) as well as examples leveraging VCOP's vision specific capabilities (histogram, scatter-gather, repeat-loop).
VCOP Kernel C Reference Guide	Description of the VCOP kernel compiler (vcc), a tool used for programming the EVE Vector Core (VCOP) at a C-like high-level language called VCOP-C.
EVE Subsystem Reference Guide	Specification for modules that are not compute-related in the EVE system: MMU, interrupts, safety features, SCTM, SMSET.
ARP32 CPU and Instruction Set Reference Guide	Description of the EVE's scalar core architecture and instruction set. Since APR32 is programmed in C, reading this document is optional.
ARP32 Compiler Reference Guide	Description of the ARP32 C-compiler tool usage. The reading of this document can be skipped.
ARP32 Assembly Language Tools Reference Guide	Description of the ARP32 Assembly Language. The reading of this document can be skipped.

EVE Tools and Software package

EVE Getting Started

Obtaining tools for installation

- EVE simulator works with Code Composer Studio Version 5.2 (CCSv5.2)
  - Download CCSv5.2 at the following link: (http://processors.wiki.ti.com/index.php/Download_CCS)
  - For introductory information on CCSv5: (http://processors.wiki.ti.com/index.php/CCSv5_Getting_Started_Guide)
- EVE simulator tools download from TI
  - EVE Simulator Pack available for customers with NDA signed
  - Please contact your local TI representative for download location

EVE – Simulator Install

The following steps describe the installation process for EVE simpack.

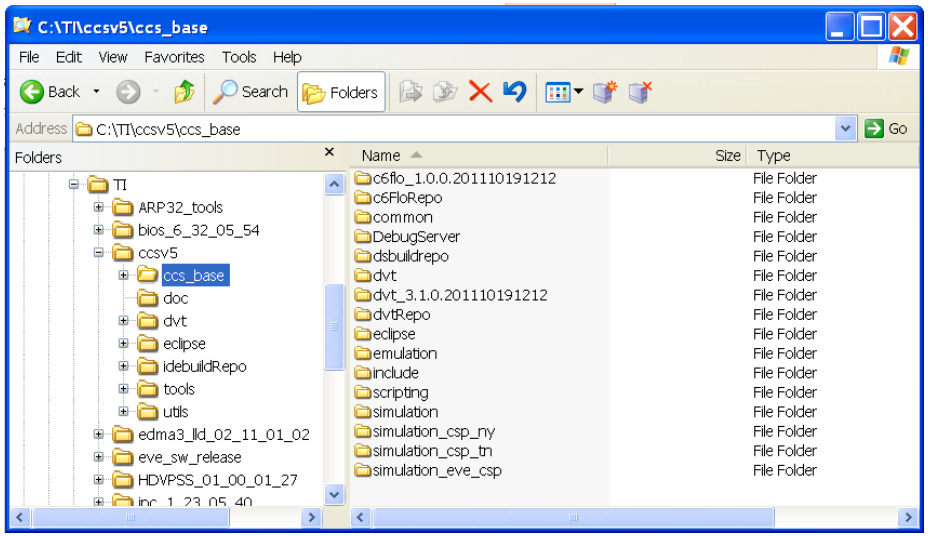
1. Install CodeComposer Studio (CCS5.1 or newer)
2. Install EVE Simulator Pack (ti_simpack_csp_eve_setup_x.x.x.x.exe)

For **CCS5.2** select destination location **...\ccsv5\ccs_base**

The installer creates one folder „*simulation_eve_csp*“ at ccs_base

Documentations are available

in the subfolder docs/releasenotes



ARP32 Code Generation Tools Install

1. In the CCSv5 main window, select Help->Install New Software...

This brings up the Available Software window.

2. In the Available Software window, type the EVE P2 installation server

URL into the "Work with:" text box, and press Enter.

Server URL:

Win32:http://software-dl.ti.com/dsps/dsps_public_sw/sdo_ccstudio/CCSv5/Updates/cgtools/ARP32/p2win32

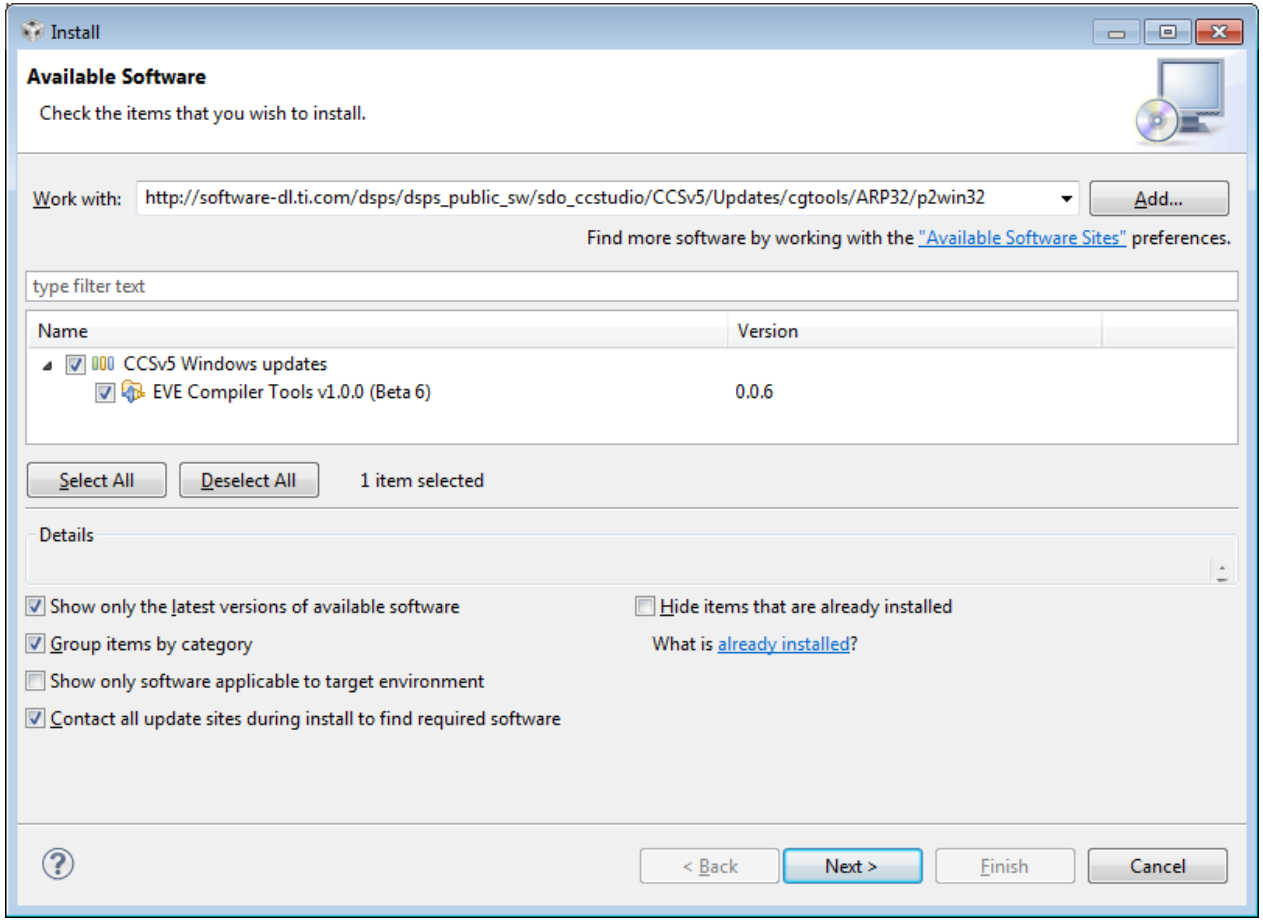
Linux:http://software-dl.ti.com/dsps/dsps_public_sw/sdo_ccstudio/CCSv5/Updates/cgtools/ARP32/p2linux

The contents of the site should appear in the selection area in the middle of the window. You should see "CCSv5 Windows updates".

3. Expand "CCSv5 Windows updates".

- You should see "EVE Compiler Tools v1.0.0 (Beta 6)".

4. Select the EVE Compiler Tools checkbox, then click "Next>".



EVE software package

- The EVE software package provides the following components:
  - evestarterware_xx_xx_xx_xx: library and source code of functions to control different modules of the EVE subsystem.
  - evelib_xx_xx_xx_xx: kernel-C code of vision, image and signal processing functions and example codes using them.
  - arp32_cgt_xxx in avdsk_xx_xx_xx_xx directory: arp32 codegen tools. May be absent if you downloaded a late version of the EVE software package. In this case use Code Composer update to obtain the tools.

  Since the Makefiles released with the evelib's examples use the harcoded path C:\Program Files\Texas Instruments\ARP32_tools, need to copy ARP32_tools installed in your CCS directory to the aforementioned path.

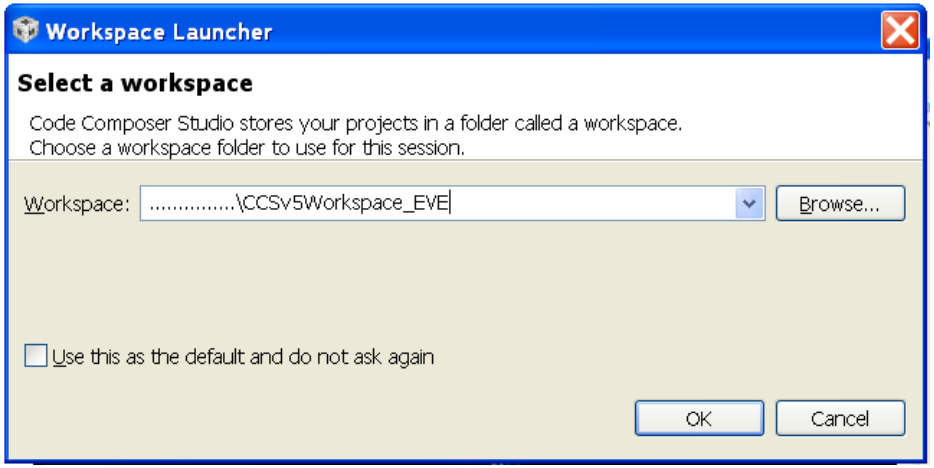
- EVE software package download from CDDS
  - EVE software package is available for customers with NDA signed on CDDS
  - As coordinated by your local TI contact you will be receiving a link to make general registration on the CDDS database system.

EVE and CCS v5.2

EVE Getting Started

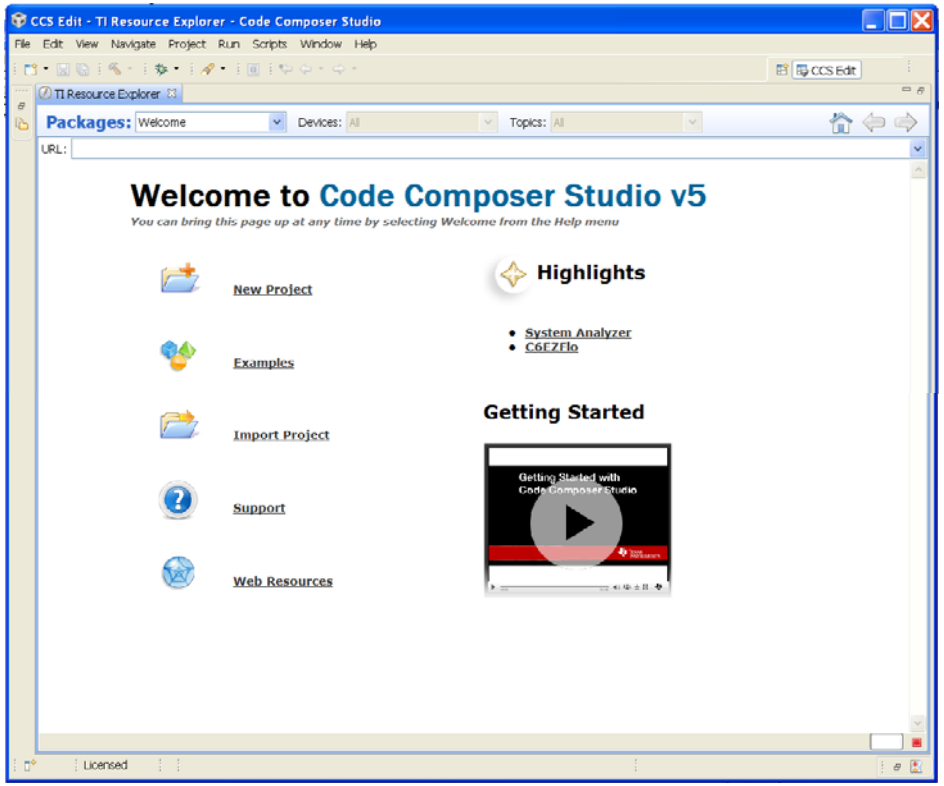
CCS View modes

- Open CCSv5 will prompt to select or create a workspace
- It is recommended to create one workspace for EVE development



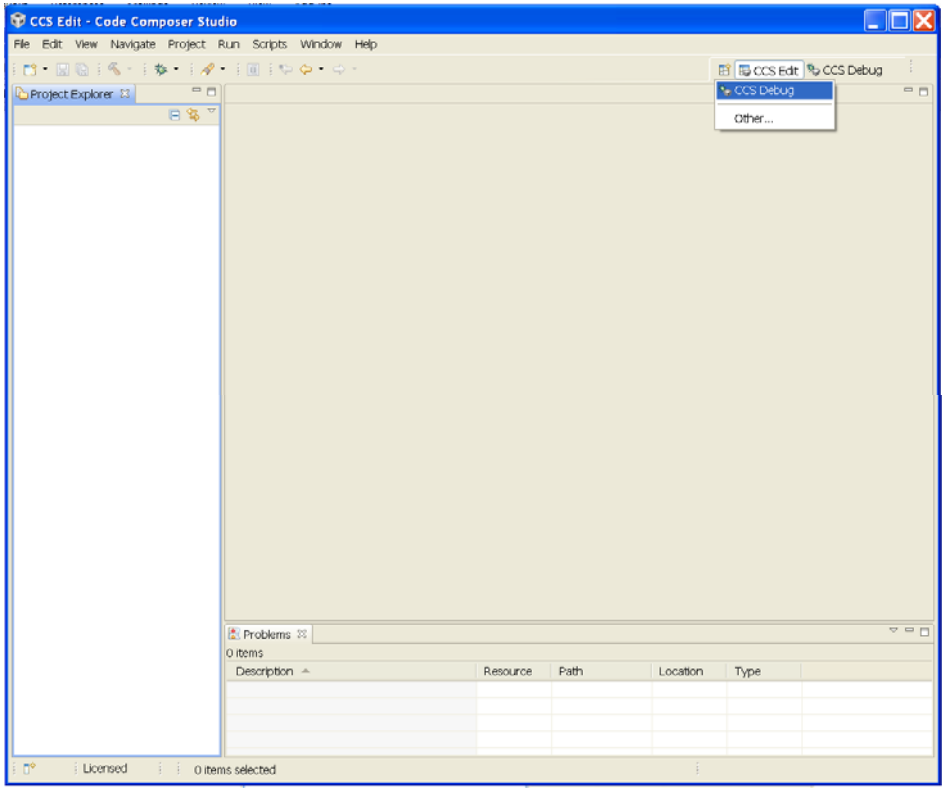
CCS View modes

- Creating a new workspace CCS comes up with the welcome page
- Close the page using the tap “TI Resource Explorer”



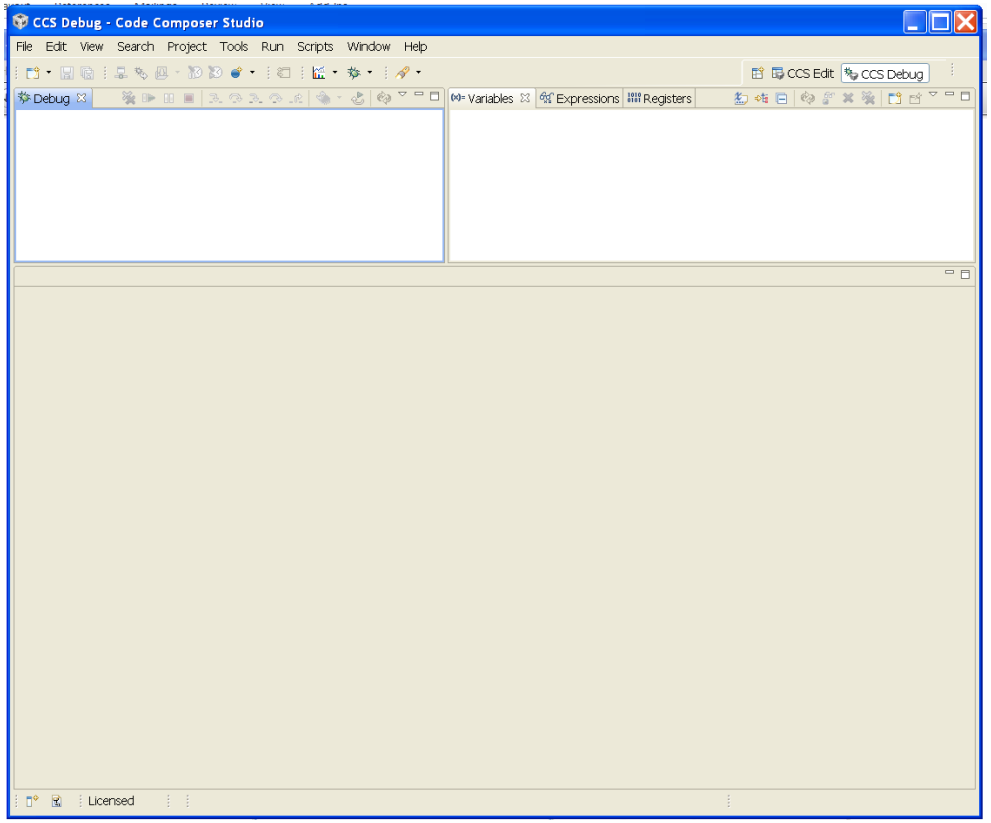
CCS View modes

- One click to the “Open Perspective” button will add the CCS Debug perspective (small button on the left of the CCS Edit button).



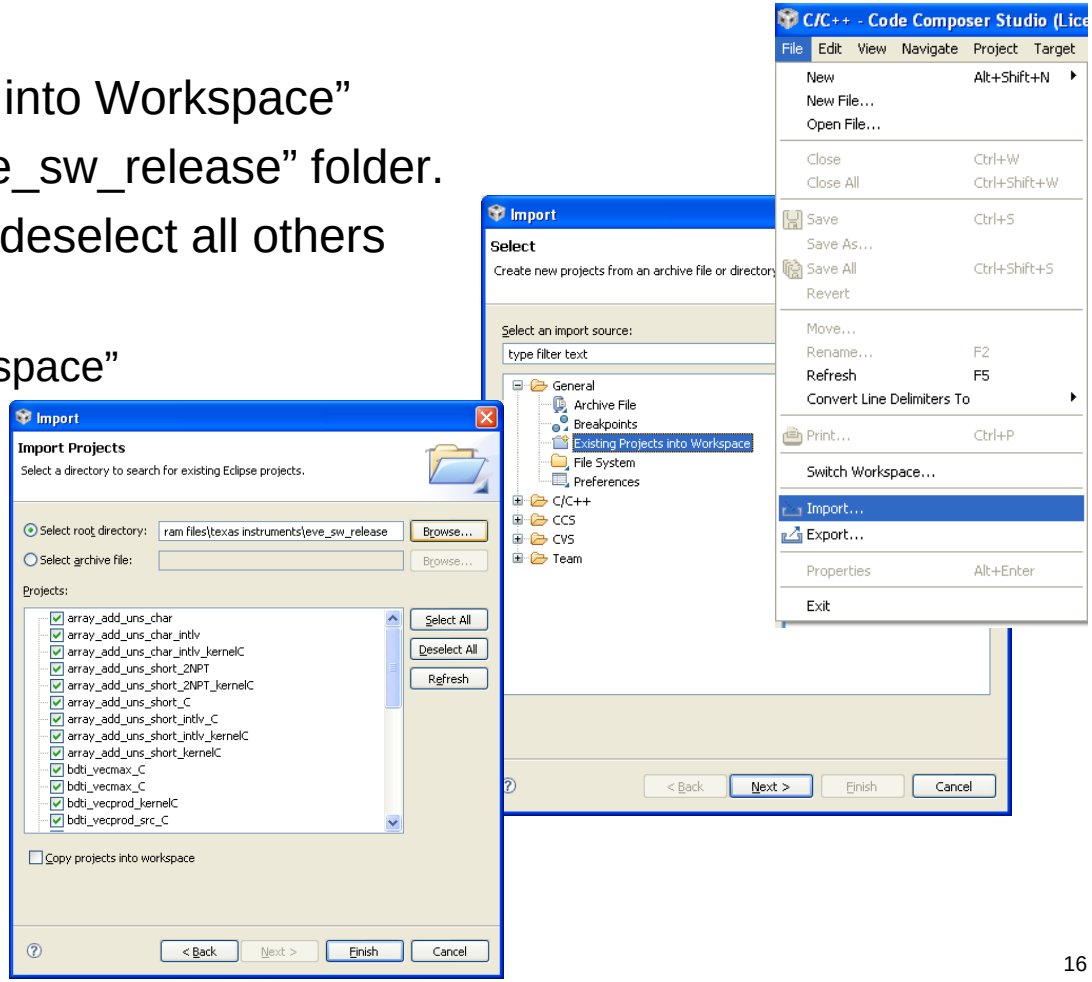
CCS View modes

- Two default perspectives are available
  - CCS Edit perspective to manage SW projects including coding
  - CCS Debug perspective to debug the system using simulator or emulator



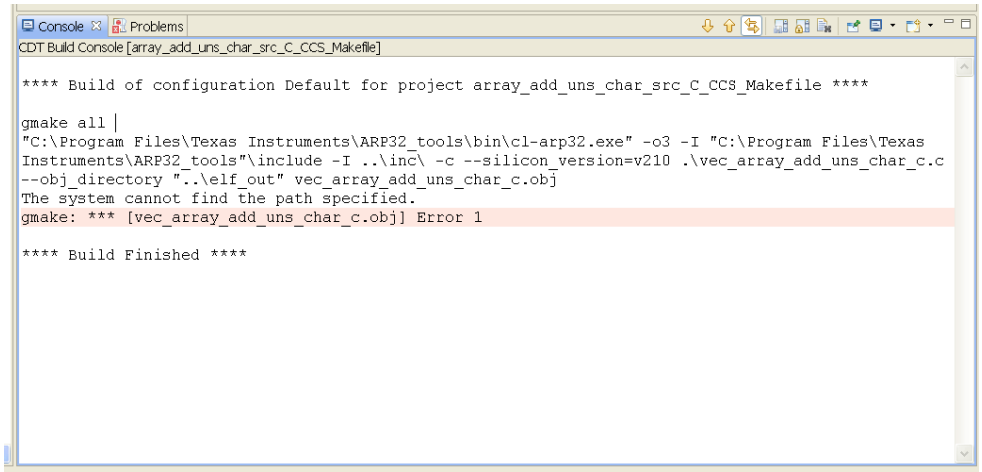
Load existing EVE SW projects

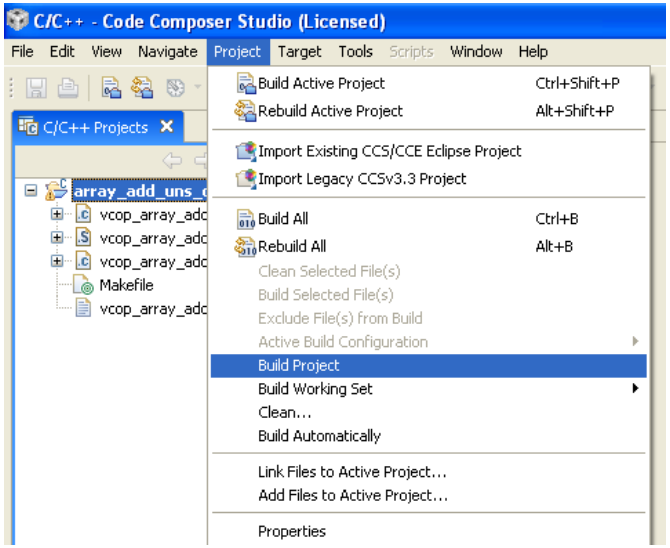
- Switch to CCS Edit perspective and import existing projects
  - Menu: File>Import
  - Select “Existing Projects into Workspace”
  - Browse and select “..\eve_sw_release” folder.
  - Select first example and deselect all others
  - Don’t tick the option:
    - “Copy projects into workspace”
  - Complete with “Finish”



Build existing EVE SW project

- Select the project to build “Project tab > Build project”
- If the system responses with the following error:
  - Copy the folder “ARP32_tools” into
  - *C:\Program Files\Texas Instruments*
  - Build again



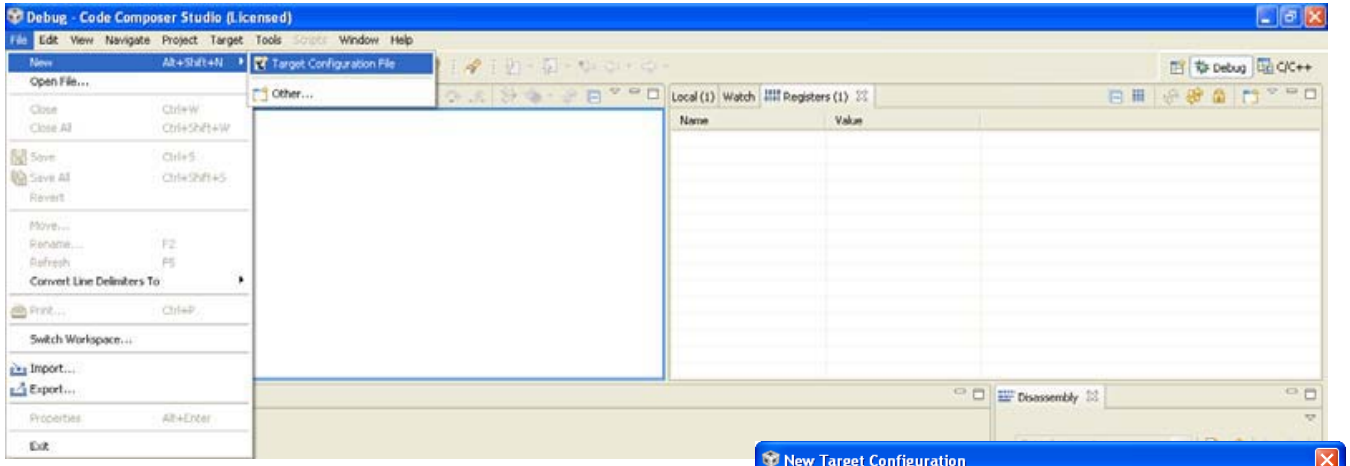


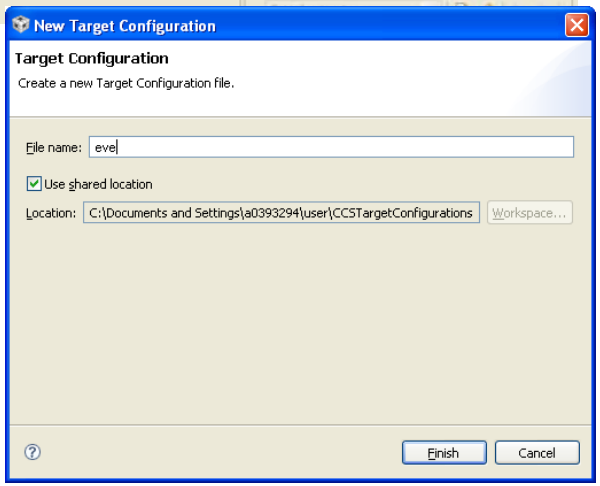
Build existing EVE SW project

- Once build was successful the executable file can be found
  - *Location: "<Project_folder>\elf_out" folder*
  - *Executable name: "<Project_name>.out"*

Target creation for EVE simulator

- Create a Target Configuration within the CCS Debug perspective
  - *Menu: "File>New>Target Configuration"*

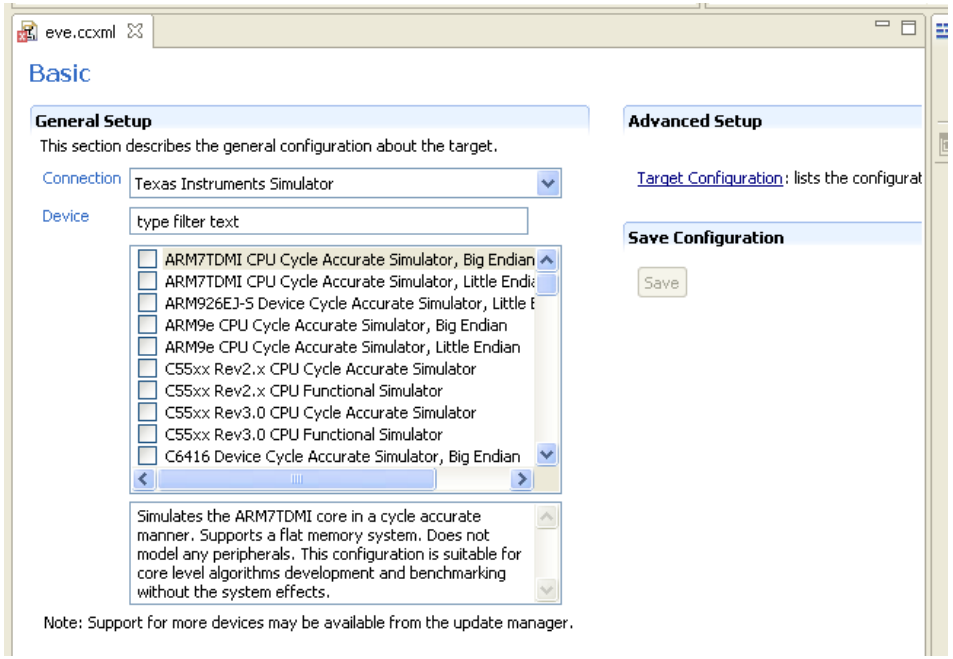




  - Give a file name to create a target

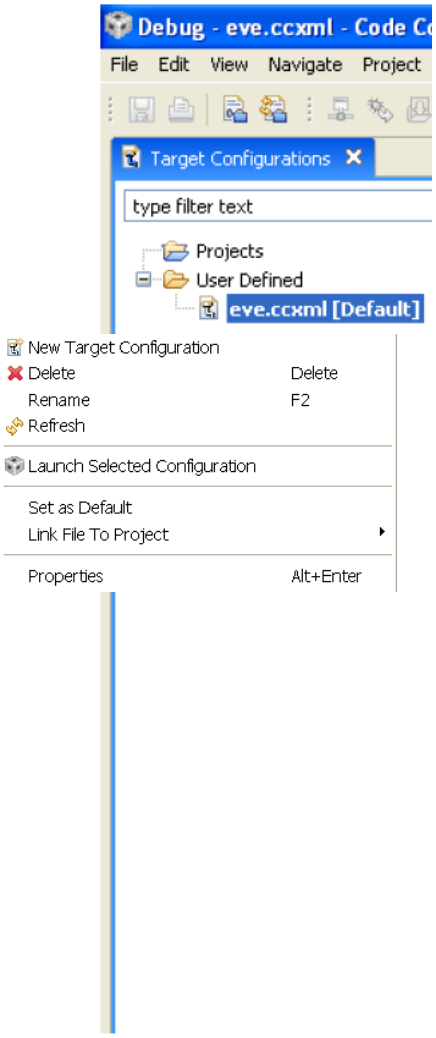
Target creation for EVE simulator

- Select “EVE(45nm)1.0 Simulator, Little Endian” as the device
  - Typing “eve” in the device filter will show eve simulators only
- Complete with “Save” button and close window



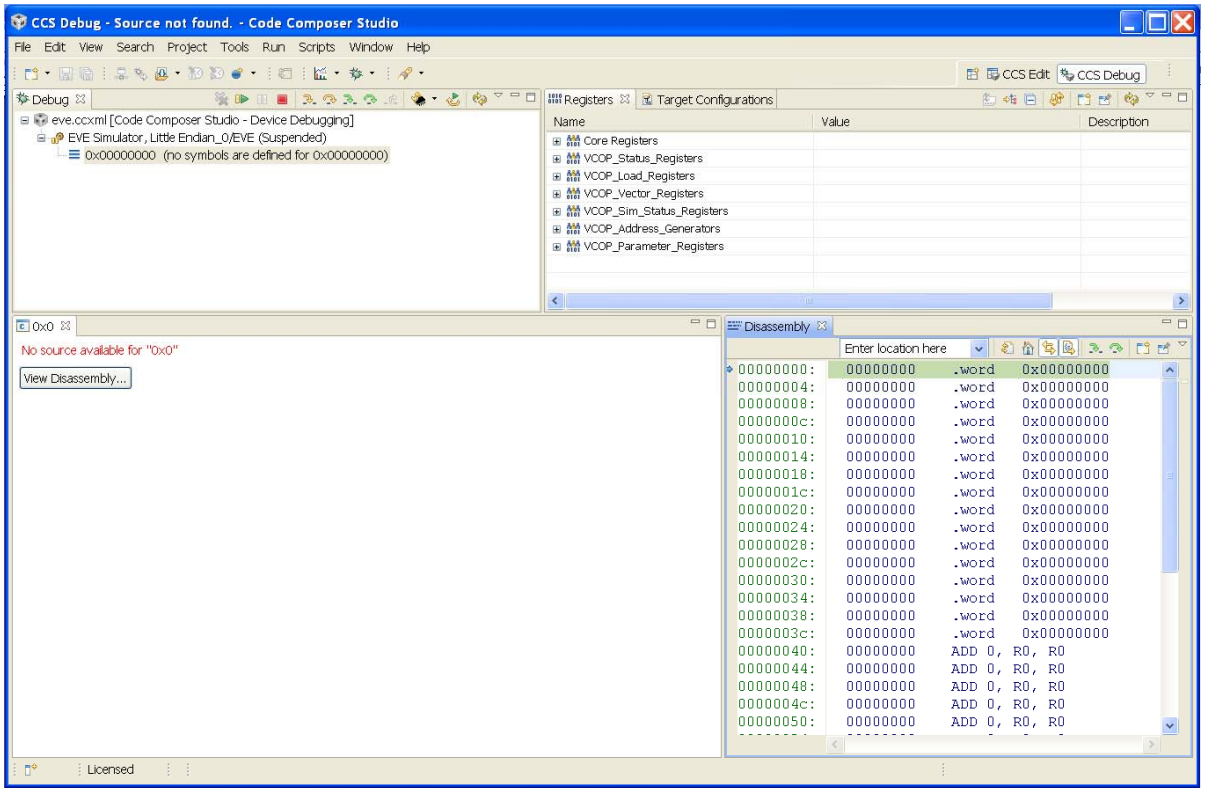
Launch EVE simulator

- Under menu “View” the target configurations window can be opened
- Right click on the eve target file and choose
  - “Set as Default”
- Right click again to choose
  - “Launch Selected Configuration”
- The simulator will be launched.



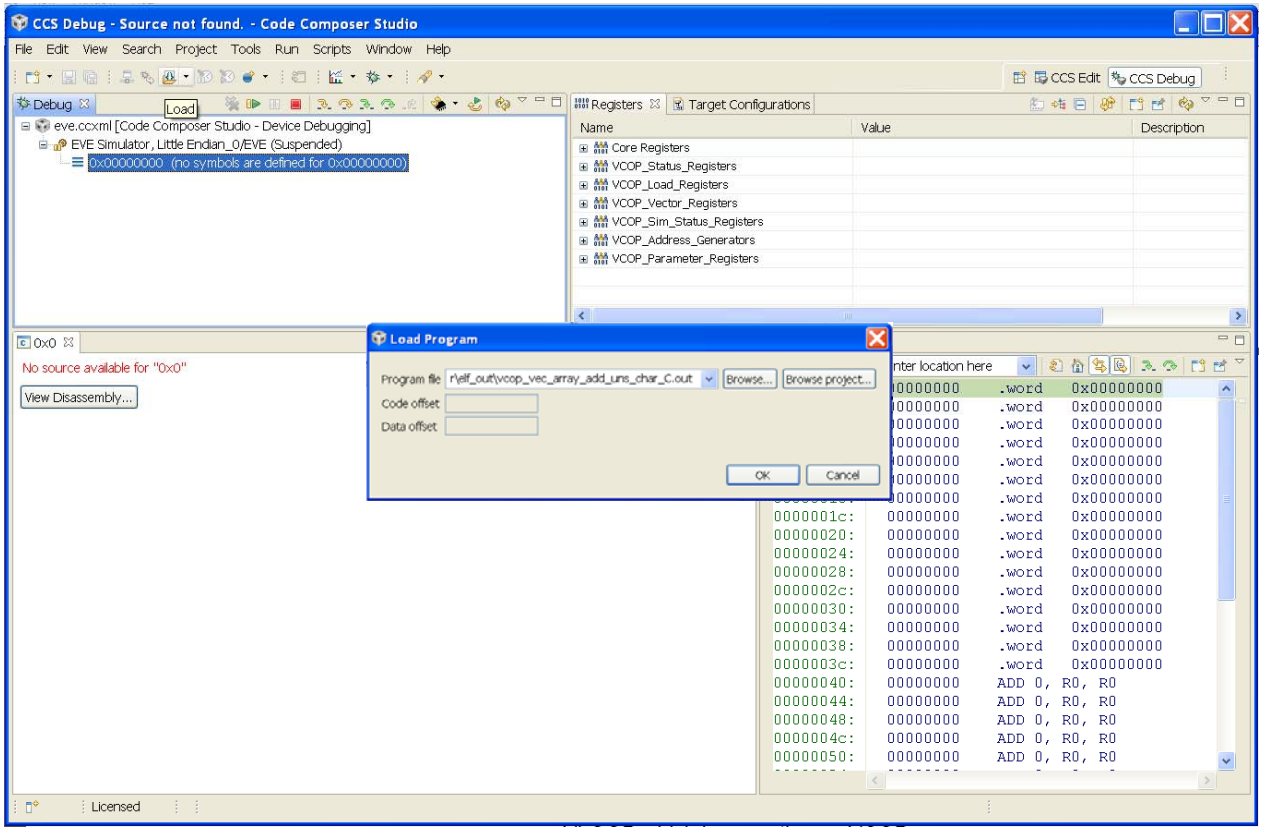
Launch EVE simulator

- EVE simulator provides all the basic debug features as such as
  - Step in, Step over, Assembly in, Assembly over, Breakpoint and Reset
  - The menu “View” allows to open further windows as such as “Disassembly”



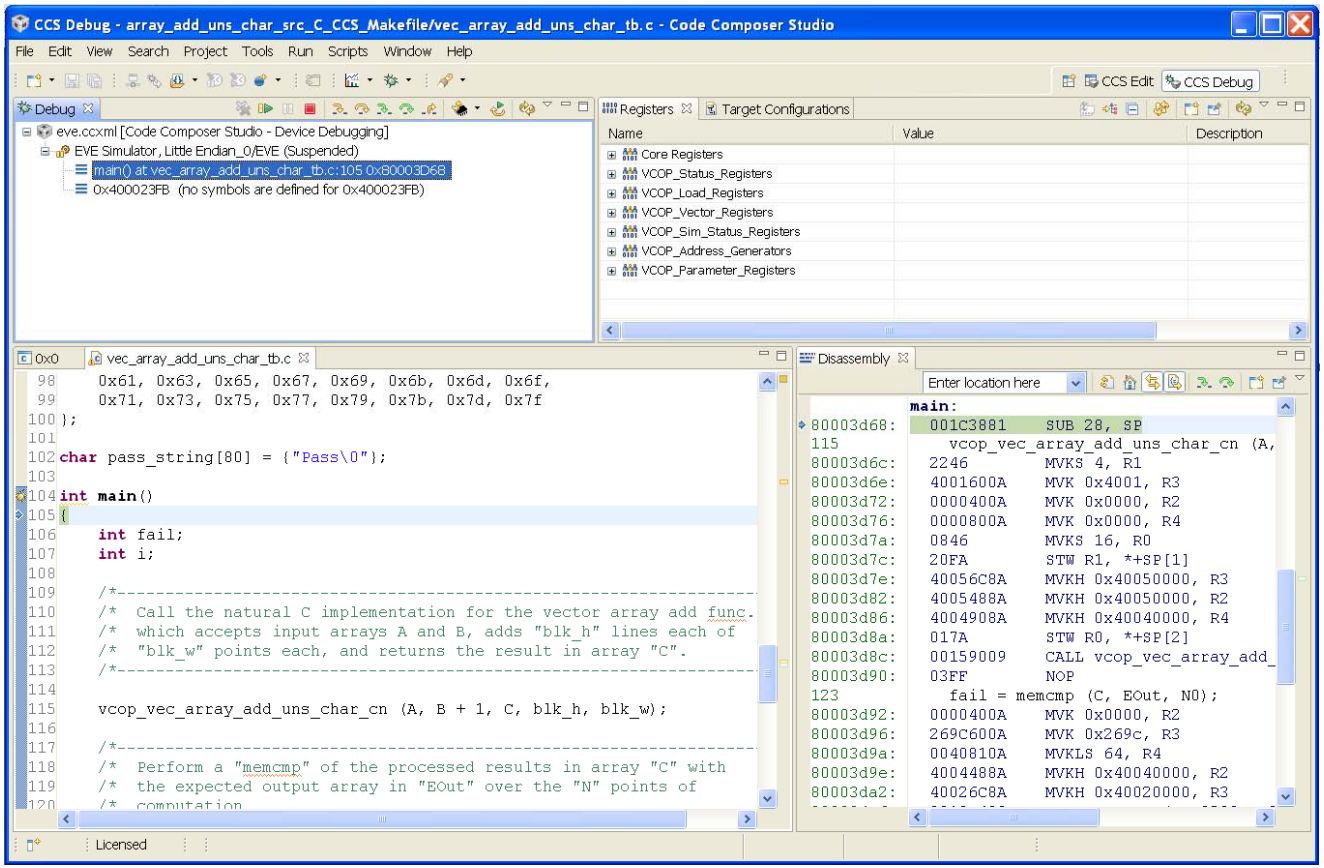
Load first executable into EVE simulator

- Hit the “Load” button to select the executable to be loaded
  - The executable file can be found within the *“<Project_folder>\elf_out” folder*



Load first executable into EVE simulator

- On load completed the PC will be holding on main() with source file opened.



Learning from EVE examples

- EVE simulator release contains an eve_sw_release directory
- It contains N example implementations of
  - ~100 algorithmic functions
  - ~40 application & DMA projects
- Great source to learn and apply knowledge to own functions (→ N+1)
- Can be used to copy/paste for building more complex algorithms