TI Technology Days 2010

Code Composer Studio v4 and BIOS6

IDE and Real-Time OS for all TI Processors

Andreas Görgner

Texas Instruments
FAE Embedded Processors

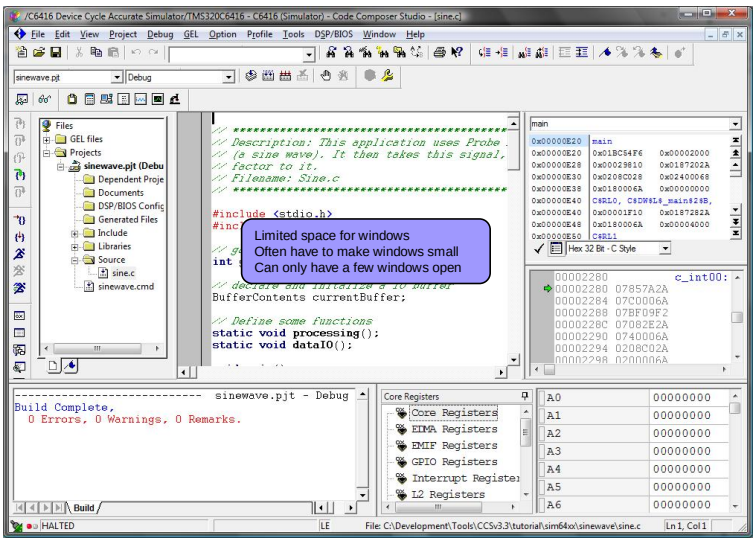
Agenda

- Code Composer Studio v4: IDE
  - What is CCS?
  - CCSv3.3 vs. CCSv4
  - Key features: project manager and editor
  - Code Generation Tools; Emulation
  - Available CCS Versions
  - Getting support
- BIOS6: Real-Time Operating System

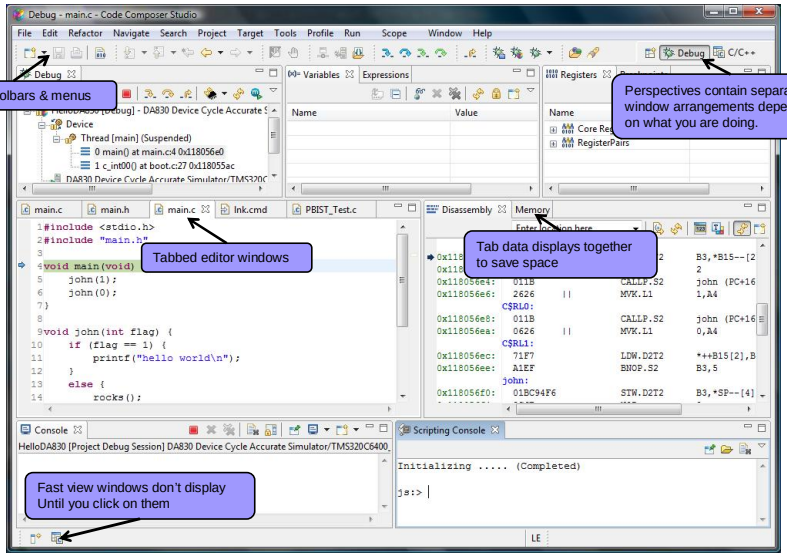
Code Composer Studio Summary

- What is CCS?
  - Integrated development environment for TI embedded processors
  - Based on Eclipse open source software framework
  - A suite of development tools for compiling, editing, debugging, profiling and analyzing embedded applications
- Why Eclipse?
  - Quickly becoming a standard for IDEs
  - Excellent software architecture
  - Ability to leverage the work of others
  - Wide selection of 3rd party plug-ins available
- What device families does CCS support?
  - MSP430, C28x, Stellaris, TMS470, TMS570, Sitara, OMAP, DaVinci, C5000, C6000…

CCSv3.3 Environment

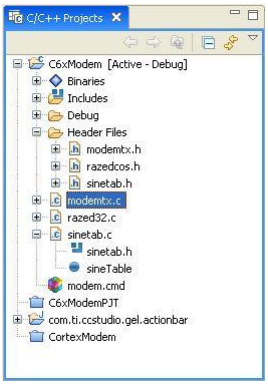


Now: CCSv4 Environment



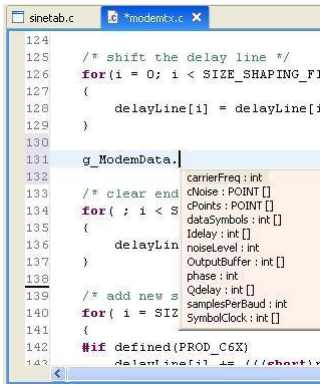
C/C++ Projects View

- All files in the project directory are in the project
- "Add" files copies them into the project directory
- "Link" files just references the files and leaves them in their original location
- Folder structure represents directory structure of project
  - Dragging files between folders will move them
- "Includes" displays header search paths
- Can use certain compiler and DSP/BIOS version for each project
  - Use "older" tools for projects in maintenance mode
  - Use latest tools for current projects



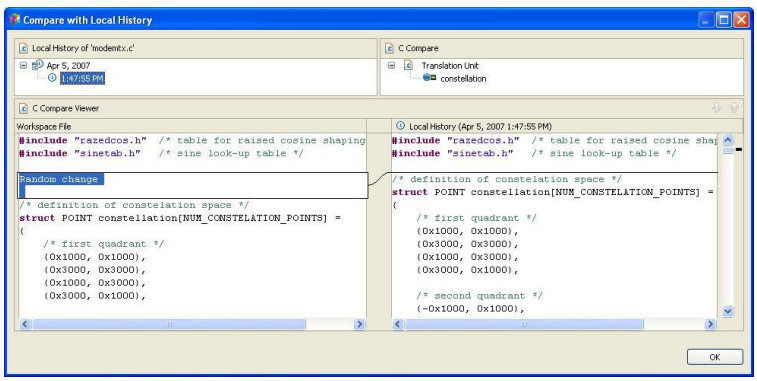
Advanced Editor Features

- Code Completion
  - Complete word
  - Auto-member information
  - Auto-parameter information
  - …
- Navigation
  - Back/Forward buttons
  - Back to last edit button
  - Go to definition
  - Go to declaration
- Show line numbers
- Code Folding
  - Collapse functions



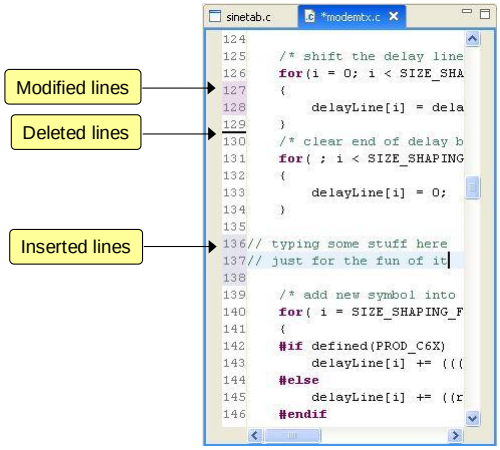
Local History

- Eclipse keeps a local history of source changes
- You can compare your current source file against any previous version or replace it with any previous version



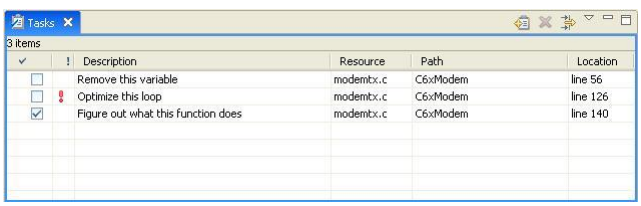
Edit Markers

- If you have the line number column on it also indicates changes in your source file since your last save



Tasks

- Your own TODO list
  - Allows you to keep track of things to do
- Associate tasks with source lines
  - Add a task by right clicking in the editor left selection margin
- Tasks view shows tasks that are contained in your workspace



Update Delivery

- Problems:
  - People are unsure of what updates are needed
  - Downloading updates is painful
- Solution:
  - CCS will automatically check for updates on startup and indicate if content is available
  - Spectrum Digital & Blackhawk drivers are included in the CCS install
  - Service releases will only install content relevant to your installation (i.e. C2000 users only see C2000 content)
  - Much faster file server!!!!!!!!

Compilers

- Goals
  - Make performance of TI silicon easily accessible via high-level language
  - DSP & MCU customers should be able to write all their code in C (or C++) and meet their performance and code size requirements
- Key Requirements
  - Expressiveness - within widely accepted standards
  - Optimization - targeted to DSP / SIMD code
  - Robustness - via validation and internal usage
  - Standards – strict ISO C/C++ compliance
- Availability
  - Available as a standalone package, or part of Code Composer Studio
  - TI DSP Compilers are a FREE download
    - https://www-a.ti.com/downloads/sds_support/TICodegenerationTools/index.htm
    - Download site includes binaries, READMEs, and Documentation

JTAG Emulation

- XDS100 class
  - Super low cost at $79, includes free CCS license limited to use with XDS100
  - Robust and efficient JTAG emulation controller
  - XDS100 v1: C28x, C54x, C55x, C674x processors
  - XDS100 v2 adds ARM7/9/R4/A8 and C64x+
  - http://tiexpressdsp.com/wiki/index.php?title=XDS100

- XDS510 class
  - Industry benchmark product designed for reliability and simplicity.
  - Balance of cost and performance
  - Available from many partners

- XDS560 class
  - Technologically most advanced product offering high speed data transfers through processor and system level tracing
  - High performance with faster TCLK rates
  - Available in a variety of interfaces: USB, Ethernet, PCI
  - XDS560T model supports DSP pin Trace

- MSP430 FET
  - MSP430 is supported via separate USB Flash Emulation Tool at ~$100

Available CCS Versions

- Microcontroller Edition, which supports
  - MSP430
  - TMS320C2800
  - Stellaris Cortex-M3, TMS570 (Cortex-R4)
- Platinum Edition, which supports
  - TMS320C5500
  - TMS320C6000
  - TMS320C2800
  - MSP430
  - Stellaris
  - ARM7, ARM 9, ARM 11
  - ARM Cortex M3, ARM Cortex R4, ARM Cortex A8
  - TMS470,
  - TMS570

Licensing Options

- Evaluation license
  - Full version for 120 days (30 days plus 90 days extension)
- Free version
  - Tied to XDS100 emulator or Starter Kits
- Node locked
  - Tied to one PC
- Floating
  - Install on multiple PCs
  - Use on a single PC

CCSv4 Pricing Summary

Item	Description	Price
Platinum Eval Tools	Full tools with time limit	FREE
Platinum Bundle	EVM/DSK, sim, XDS100 use	FREE
Platinum Node Locked	Full tools tied to a machine	$1995
Platinum Floating*	Full tools shared across machines	$2995
Microcontroller Core	MSP/C2000 code size limited	FREE
Microcontroller Node Locked	MSP/C2000/Stellaris	$495
Microcontroller Floating*	MSP/C2000/Stellaris	$795

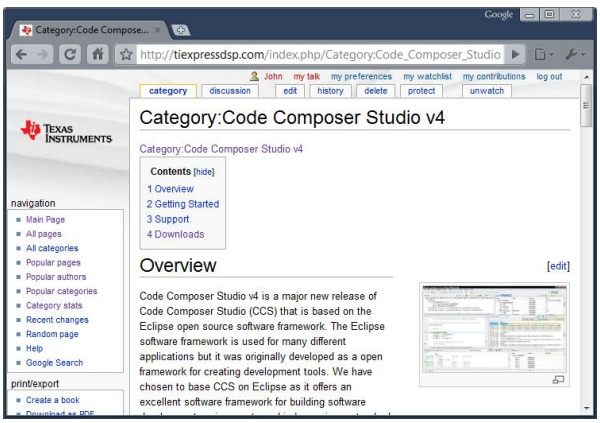
*3, 5, 10, 25 user bundles available as well
Part #s: http://tiexpressdsp.com/index.php/Licensing_-_CCSv4#Part_.23s

Licensing - FAQs

- Who gets v4 Upgrades?
  - Code Composer Studio users and under subscription.
  - Code Composer Essentials Pro users who purchased after June 1 2008
- What happens with 3.3 “site” licenses?
  - Every customer with a site license will be contacted by TI and provided with an equivalent v4 floating license package
- Can I upgrade free eval tools to full tools?
  - Yes
- Can TI extend free eval tools?
  - Yes
- What happens if my computer is off the network and I am using a floating license?
  - You can continue to use CCS, it will just let you know it could not get a license in the title bar
- How do I get the free XDS100 license?
  - Just download the CCSv4 image and click the button to generate a bundle license
- Full set of FAQs on the CCSv4 mediawiki
  - http://tiexpressdsp.com/index.php/FAQ_-_CCSv4#Licensing

#1 Resource for Information: TI Wiki

- CCSv4 Mediawiki
  - http://tiexpressdsp.com/wiki/index.php?title=Category:Code_Composer_Studio_v4
  - Documentation
  - FAQs
  - License info
  - Training
  - Downloads

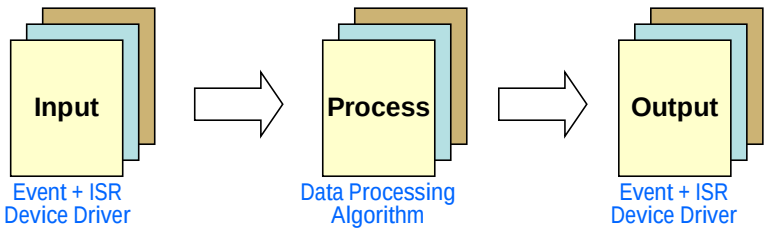


Getting Support

- See if the question is already answered
  - Check the FAQs and topics on the wiki
  - Search the e2e forums
- Post a question or issue to the Code Composer Studio or TI C/C++ Compiler forums on e2e community
  - www.ti.com/e2e
- Check the status of a bug
  - https://cqweb.ext.ti.com/pages/SDO-Web.html
  - Can create your own queries to track issues important to you

DSP/BIOS: A Free RTOS For TI Processors

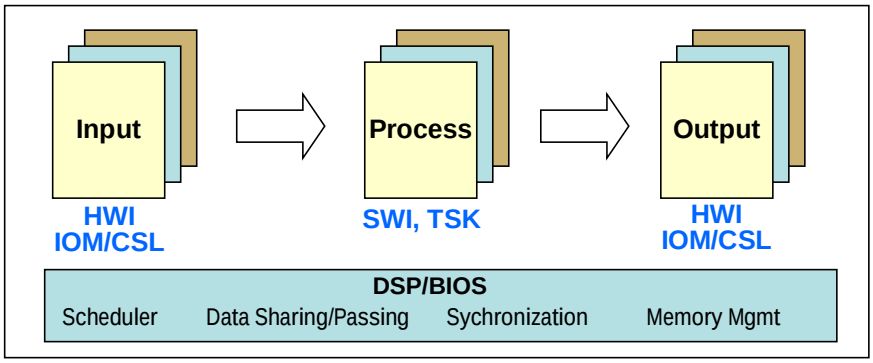
Need for an Operating System



- Simple system: single I-P-O is easy to manage
- As system complexity increases (multiple I-P-O):
  - Ø Can they all meet real time ?
  - Ø Synchronization of events?
  - Ø Priorities of threads/algos ?
  - Ø Data sharing/passing ?
- 2 options: "home-grown" or use existing (DSP/BIOS)
  (either option requires overhead)
- If you choose an existing O/S, what should you consider ?
  - Ø Is it modular ?
  - Ø Is it reliable?
  - Ø Is it easy to use ?
  - Ø Data sharing/passing ?
  - Ø How much does it cost ?
  - Ø What code overhead exists?

TTO Technical Training Organization

DSP/BIOS Overview



DSP/BIOS is a scalable, real-time kernel used in 1000s of systems today:

- Pre-emptive Scheduler to design system to meet real-time (including sync/priorities)
- Easy to use through a set of standard APIs
- Modular – pre-defined interface for interthread communications
- Reliable – 1000s of applications have used it for more than 10 years
- Footprint – deterministic, small code size, can choose which modules you desire
- Cost – free of charge

TTO Technical Training Organization

What does BIOS give to me?

- **Real-time event and statistics gathering**
- **Priority-based multi-threaded operations**
- **Basic inter-thread communication and synchronization**
- **Basic I/O for interacting with devices and other threads**
- **Basic interrupt handling capabilities**

The Scheduler

- **Started by BIOS_start()**
- **Manages all threads & synchronization**
- **Keeps track of (one or several) stacks**
- Runs highest priority thread that is ready
- **Scheduler code is run only**
  - **at scheduling related BIOS APIs to decide if the currently running thread blocked, or another thread with higher priority becomes ready**
    - **e.g. SEM_pend, SWI_post, …**
  - **at each CLK tick (system heart beat)**

BIOS Modules Summary

Instrumentation / Real-Time Analysis

LOG	Message log manager
STS	Statistics accumulator manager
TRC	Trace manager
RTDX	Real-Time Data eXchange manager

Thread Types

HWI	Hardware interrupt manager
SWI	Software interrupt manager
TSK	Multi-tasking manager
IDL	Idle function & process loop manager

Clock and Periodic Functions

CLK	System clock manager
PRD	Periodic function manager

Comm / Synch Between Threads

SEM	Semaphores manager
MBX	Mailboxes manager
LCK	Resource lock manager

Input / Output

PIP	Data pipe manager
HST	Host input / output manager
SIO	Stream I / O manager
DEV	Device driver interface

Memory and Low-Level Primitives

MEM	Memory manager
SYS	System services manager
QUE	Queue manager
ATM	Atomic functions
GBL	Global setting manager

Thread Classification

Priority

HWI — Hardware Interrupts

- Used to implement 'urgent' part of real-time event
- Triggered by hardware interrupt
- HWI priorities set by hardware

SWI — Software Interrupts

- Use SWI to perform HWI '*follow-up*' activity
- SWI's are '*posted*' by HWI's or SWI's
- Multiple SWIs at each of 15 priority levels

TSK — Tasks

- Use TSK to run different programs concurrently under separate contexts
- TSK's are usually enabled to run by setting a flag, called a '*semaphore*'

IDL — Background

- Multiple IDL functions
- Run round-robin
- Includes exchange of BIOS Real Time Analysis (RTA) information with host

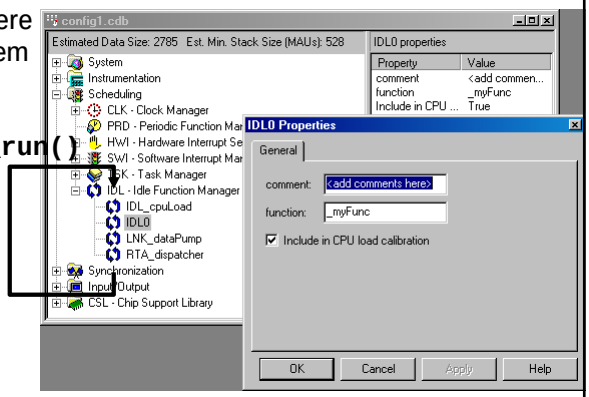
IDL

Priority

Background loop when there is nothing else in the system to do

```
void IDL_loop()
{
   while( 1 )
        IDL_run();
}
```

IDL functions **must not** block!



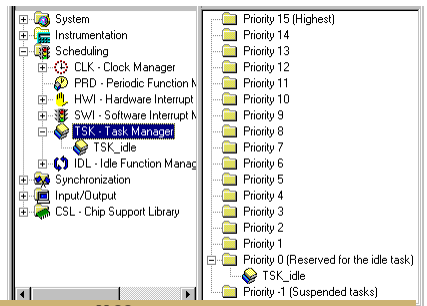
IDL
Background

- **Multiple IDL functions**
- **Run round-robin**
- **Includes exchange of BIOS Real Time Analysis (RTA) information with host**

TSK

Priority

```
void typicalTskFunc( params )
{
    init();
    while( !shutDown )
    {
        SEM_pend( … ); // block
        doStuff();
    }
}
```



TSK
Tasks

- **Use TSK to run different programs concurrently under separate contexts**
- **TSK's are usually enabled to run by setting a flag, called a '*semaphore*'**

- Each TSK has its own stack
- TSK function properties allow to specify arguments to function

SWI

Priority

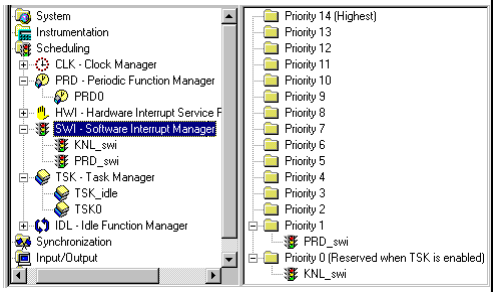
```
void typicalSwiFunc( params )
{
    doStuffQuickly();
    return;
}
```

```
void typicalIsr()
{
    lowLatencyStuff();
    SWI_post( &SWI0 );
}
```

SWI
Software Interrupts

- **Use SWI to perform HWI '*follow-up*' activity**
- **SWI's are '*posted*' by HWI's or SWI's**
- **Multiple SWIs at each of 15 priority levels**

- All SWIs run from the system stack
- SWIs have to be triggered in order to run (typically from ISR)
- Can not be terminated before end of function
- When the same SWI object is posted more than once before it has a chance to run, it only runs once



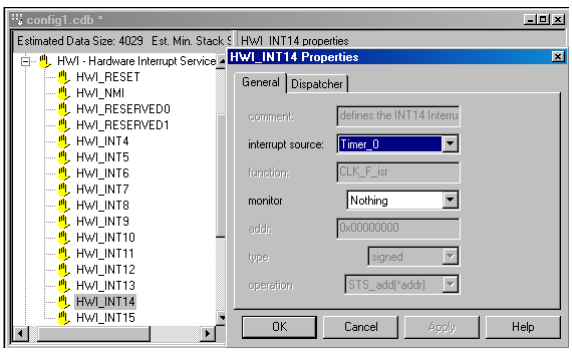
HWI

Priority

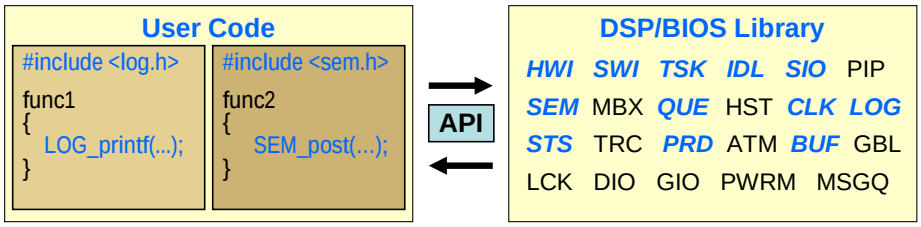
HWI
Hardware Interrupts

- Used to implement 'urgent' part of real-time event
- Triggered by hardware interrupt
- HWI priorities set by hardware

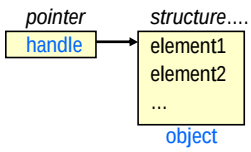
- Interrupt Service Table generated automatically
- HWIs run on system stack
- Dispatcher allows to write ISRs in C
- BIOS uses INT14 (Timer_0) by default



DSP/BIOS Environment



- DSP/BIOS is a library that contains a collection of modules with a particular
  - Interface and calling conventions
  - Set of data structures defined in the module's header file
- Application Program Interfaces (API) define the interactions with a module
  - Relates a set of constants, types, variables and functions visible to user programs
  - Object based: global parameters that control operation of *each* instance
- Objects - are structures that define the state of a component
  - Pointers to objects are called handles
  - References to the object are via the handle
  - Object based programming offers:
    - *Better encapsulation and abstraction*
    - *Multiple instance ability*



TTO
Technical Training Organization

BIOS in CCSv4

- CCSv4 includes two versions of DSP/BIOS
  - 5.40
    - Binary compatible with BIOS5.3
    - Updated to support CCSv4 integration
    - Recommended if migrating existing projects to CCSv4
    - Does not support execution graph
  - 6.20
    - Supports complete real-time analysis tools including execution graph
    - BIOS now available for all TI processor families (including ARM9, Cortex M3, MSP430)
      - No BIOS6 support for C5000, new C55x devices supported in BIOS 5.x
    - Many new kernel and analysis features
    - New APIs with enhanced error handling
    - 100% compatible DSP/BIOS 5.3x API layer
    - Legacy and new APIs can be used in the same application

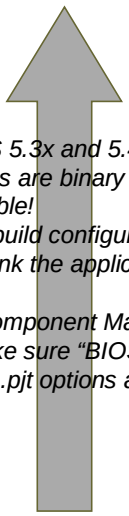
Texas Instruments

DSP/BIOS 5.x

- DSP/BIOS 5.x will continue to be supported in “maintenance mode”
  - Bug fixes. Minor device variants. No new features.
- We still support and maintain DSP/BIOS releases from 5-10 years ago
  - Source code is under version control and we can rebuild old versions with original compilers to be bit-exact with the original release.
- New devices and features will be supported in DSP/BIOS 6.x

Recent DSP/BIOS 5.x History

- DSP/BIOS 5.31 (Aug 2006)
  - Added TCI6488 and TCI6486 Support
  - 5.31.01 – 5.31.08 maintenance releases
- DSP/BIOS 5.32 (Oct 2007)
  - Added 28x Floating Point Device
  - 5.32.01 – 5.32.04 maintenance releases
- DSP/BIOS 5.33 (Sept 2008)
  - Added C674x (floating point GEM) Support
  - 5.33.01 – 5.33.06 maintenance releases
- DSP/BIOS 5.40 (April 2009)
  - CCSv4 Support (tooling and project integration)
  - Identical target code as 5.33.06
- DSP/BIOS 5.41 (Sept 2009)
  - CCSv3 and CCSv4 in single release
  - Different installer for CCSv3 tooling vs CCSv4 tooling
  - DSP/BIOS 5.41 target code is nearly identical to DSP/BIOS 5.33.06



All BIOS 5.3x and 5.4x Releases are binary and API compatible!
Just re-build configuration and re-link the application.

(use “Component Manager” and make sure “BIOS Builder” .pjt options are in synch)

DSP/BIOS 6.x Scope

- DSP/BIOS 6.x is a major ‘rewrite’ of the Kernel and Tools
  - Refactored the kernel to be more modular and portable
    - Added key features as requested by large BIOS 5 customer base
    - Mostly ‘C’ code instead of assembly code
  - Real-time Analysis (RTA) and Run-time Object View (ROV) based on Data Visualization Toolkit (DVT) and CCSv4/Eclipse IDE
    - Better user interface and expanded feature set
  - Design is based on Real-Time Software Components (RTSC) Eclipse Project
    - See rtsc.eclipse.org for more information
  - Open and Extensible Configuration tooling
    - Configuration tool now supports non-BIOS target content
- Existing test suites are used to ensure quality

Kernel Highlights

- Open Source
  - Eclipse Public License for DSP/BIOS 6.21 and beyond
- Support for ARM and DSP Cores
- Many new Kernel and Analysis Features
- New APIs with unified error handling
- Compatible DSP/BIOS 5.3x API layer
  - Legacy and new APIs can be used in the same application
  - A few (seldom or never used) modules are not supported
  - Zero source changes are required

BIOS Schedule

- Current Version: 6.30
  - Support for C2000, C6000, Stellaris M3, MSP430
  - Includes source with BSD open-sourse license
- With version 6.30: name change from DSP/BIOS to SYS/BIOS
- No support for C55x planned **L**

J Thank You! J