

Digital Media Solution	
Product Description	07/20/2010
Comm. Sys. Solutions	1 of 8



Digital Media Solution

Product Description

Note

Information in this document may be subject to change at any time.

CSS' Digital Media Solution is targeted for the Texas Instruments' DaVinci™ platform. A production ready media framework is available which delivers the modern kernels and peripheral support, modern debugging tools, and a software framework which encapsulates the popular VL42 drivers, ALSA drivers, and XDM codecs.

The CSS Advantage

CSS' Digital Media Solution offers several advantages.

1. Efficient, extensible, and flexible software framework.
 - a. A well designed, structured C++ framework with error handling allows the customer to easily develop applications, without chasing common A/V issues for months.
 - b. Modern multi-threaded programming allows for minimal CPU overhead while delivering A/V with minimal delay.
 - c. A/V time stamping and synchronization is not an afterthought. It is accounted for all the way from the drivers, through the framework, and to the customer's application.
 - d. The framework is easily extended by the customer or by CSS. Source code is provided so that it can be expanded for additional uses.
2. Rapid Development
 - a. The modular C++ approach allows for simple required interfaces while providing more advanced optional control.
 - b. Basic applications can be written in days, using the standard audio/video processing threads which are supplied.
 - c. Tricky A/V details are dealt with by the lower level classes and hidden from the customer. The customer's applications do not need to deal with things such as "physical addresses", "A/V synchronization", and "driver control".
 - d. A full development system consisting of a toolchain, debugger, and source code is delivered. This greatly aids in development and debugging of the customer's application. No need to guess as to what a function may be doing. Just examine the source code or step into it with the included debugger.
3. Well Tested Production Quality Releases
 - a. The framework has been designed as a product and not as an engineering prototype.

Digital Media Solution	
Product Description	07/20/2010
Comm. Sys. Solutions	2 of 8



- b. The software framework and hardware designs have been used in real products where issues that are found have been used to improve the robustness.

Product List

Item	Description
<i>BSP</i>	<i>Board Support Package – Source Code Release</i>
BSP-DM355	DM355 Platform
BSP-DM365	DM365 Platform
BSP-DM6446	DM6446 Platform
BSP-DM6467	DM6467 Platform
BSP-DM6467T	DM6467T Platform
BSP-DM8168	DM8168 Platform
BSP-DM814x	DM814x Platform
<i>DMFW</i>	<i>Digital Media Framework – Source Code Release</i>
DMFW-DM355	DM355 Platform - MPEG-4
DMFW-DM365	DM365 Platform – MPEG-2, MPEG-4, H.264
DMFW-DM6446	DM6446 Platform – MPEG-2, MPEG-4, H.264
DMFW-DM6467	DM6467 Platform – MPEG-2, MPEG-4, H.264
DMFW-DM6467T	DM6467T Platform – MPEG-2, MPEG-4, H.264
DMFW-DM8168	DM8168 Platform – MPEG-2, MPEG-4, H.264
DMFW-DM814x	DM814x Platform – MPEG-2, MPEG-4, H.264

Digital Media Solution	
Product Description	07/20/2010
Comm. Sys. Solutions	3 of 8



Item	Description
<i>DMALG</i>	<i>Digital Media Algorithms – Binary Release</i>
DMALG-AEC-ARM9	Acoustic Echo Canceller - ARM9
DMALG-AEC-C64	Acoustic Echo Canceller – C64 DSP
DMALG-XCOD	Transcoder – Supports CSS™ DM framework codecs.
<i>GUIFW</i>	<i>GUI Framework – Source Code Release</i>
GUIFW-DM355	DM355 Platform
GUIFW-DM365	DM365 Platform
GUIFW-DM6446	DM6446 Platform
GUIFW-DM6467	DM6467 Platform
GUIFW-DM6467T	DM6467T Platform
GUIFW-DM8168	DM8168 Platform
GUIFW-DM814x	DM814x Platform
<i>DEMO</i>	<i>Demo Applications – Source Code Release</i>
DEMO-PMP	Portable Media Player
DEMO-STREAM	Streaming Media
DEMO-VIDCONF	Video Conference

Item	Description
<i>HW</i>	<i>Hardware Support</i>
SCH	Schematic Design
PCB	PCB Layout Design
PROTO	Prototype Manufacturing
MFG	Manufacturing
<i>Support</i>	<i>Demo Applications – Source Code Release</i>
SUPT-ALG	Algorithm Design
SUPT-BSPFW	BSP/Framework Support
SUPT-APP	Application Support
SUPT-TECH	Technical Support
SUPT-SCHPCB	SCH/PCB Design Support

Digital Media Solution	
Product Description	07/20/2010
Comm. Sys. Solutions	4 of 8



Board Support Package

Board support packages (BSP) are available for several platforms in binary and source code form. The BSPs contain the following components.

1. Bootloader support
 - a. UBL/U-boot
 - b. Method to flash bootloader via USB or SD card.
2. Linux 2.6.28 kernel, optimized for the specific DaVinci™ platforms.
 - a. DM355, DM365, DM6446, DM6467, DM6467T, DM8168, DM814x
 - b. Optimized for fast bootloading
3. Kernel drivers supporting A/V input and output.
 - a. V4L2 / FB support for several hardware video codecs.
 - b. ALSA support for several AICxx audio codecs.
 - c. Userptr or MMAP support.
 - d. Fast DMA for moving audio/video, when required.
4. Peripheral drivers
 - a. UART, CF, SD, SDHC, USB gadget/host, Ethernet
 - b. Custom drivers.
5. Production support
 - a. Windows/Linux host drivers for flash loading.
 - b. Windows/Linux applications for flash loading.
 - c. Test system.
6. Filesystem
 - a. CRAMFS/YAFFS2/JFFS2/UBIFS support
 - b. Busybox system, less than 32 MB.
 - c. Read-only and Read-write support.
 - d. Optional UDEV support.
 - e. Support for external devices.
7. Development tools
 - a. Modern GCC/GLIBC/GDB development kit.
 - b. Eclipse development support, including remote debugging.
 - c. Eclipse plug-in for the CSS toolchain.

Digital Media Framework

The digital media framework from CSS is an application framework that handles the audio/video details which are often problematic in designing these systems. A/V synchronization as well as proper algorithm error handling is handled, eliminating these concerns from the application developer. The framework is built on top of XDM codecs and thus can easily handle the changing of codecs. Additionally, several custom algorithms are available for integration into the XDM Codec Server. A sample of the development platform as well as the MPEG Encode class is shown in Illustration 1. The framework is written in C++ but can be used in a C system. The C++ is efficiently written in such a way as to maximize performance.

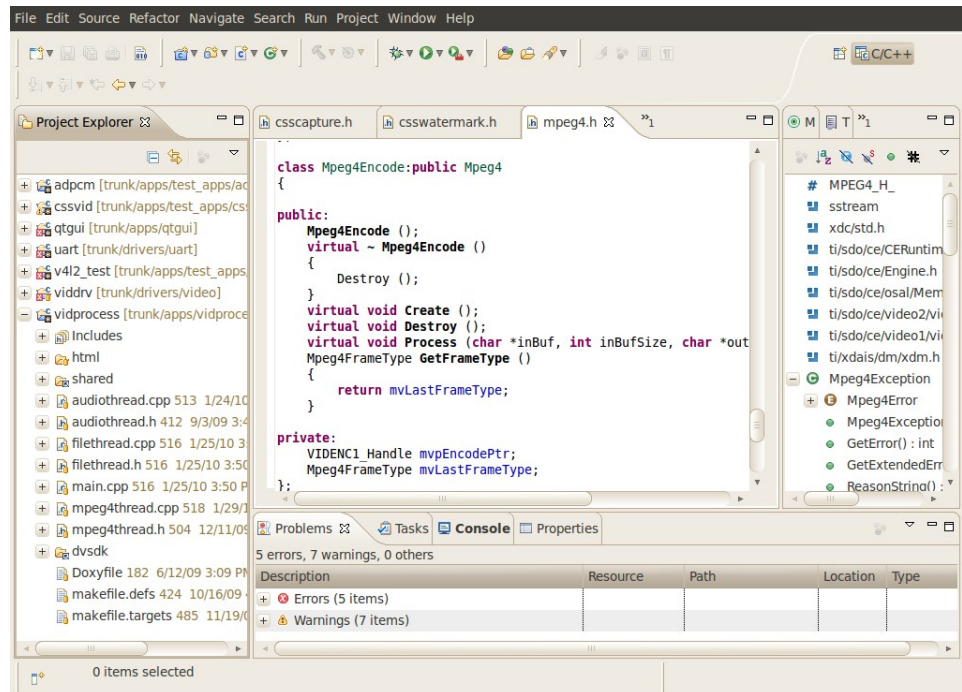


Illustration 1: Eclipse Development Platform

The digital media framework contains the following components.

1. Video capture/display support.
 - a. Interfaces to any V4L2 drivers.
 - b. MMAP/USERPTR support.
 - c. Simple interface:

```
void Start ();
void Stop ();
int GetFrameReady ();
int GetFrame ();
void PutFrame (int index);
int GetFrame (void *&buffer, int &length);
void PutFrame (int index, void *buffer, int length);
VideoBuffer & operator [] (int index)
```

- d. Buffer management
 - i. Encapsulated for ease of use.
 - ii. Physical address / virtual address properly maintained.
2. Video encode/decode support
 - a. Encapsulated XDM codecs.
 - b. Transcoding encapsulation.
 - c. Codec statistics accumulated.
 - d. Robust error handling using C++ exceptions.
 - e. Simple interface for all codecs:

```
virtual void Create () = 0;
```

Digital Media Solution	
Product Description	07/20/2010
Comm. Sys. Solutions	6 of 8



```
virtual void Destroy () = 0;
virtual void Process (char *inBuf, int inBufSize,
                    char *outBuf, int &outBufSize);
```

f. Advanced custom interface also available.

3. Thread support

a. Extensible thread class

```
CssThread ();
~CssThread ();
virtual void Stop (bool wait = false);
bool IsRunning ();
virtual void Start ();
void SetPriority (int priority);

virtual void Thread () = 0; // Inherit and override for user threads
```

b. Video capture/display threads

```
CssVideoThread ();
~CssVideoThread ();
bool HasBuffer ();
bool HasWatermarkBuffer ();
CssMemoryBlock GetBuffer ();
void PutBuffer (CssMemoryBlock & data);
CssMemoryBlock GetWatermarkBuffer ();
void PutWatermarkBuffer (CssMemoryBlock & data);
int GetFrameDrop ();
```

c. Video encode/decode threads

```
Mpeg4Thread ();
virtual ~ Mpeg4Thread ();
bool HasBuffer ();
bool HasWatermarkBuffer ();
CssMemoryBlock GetBuffer ();
void PutBuffer (CssMemoryBlock & data);
CssMemoryBlock GetWatermarkBuffer ();
void PutWatermarkBuffer (CssMemoryBlock & data);
void SetImageBuffer (CssMemoryBlock & image);
void Capture (std::string filename); // optional file support
void View (std::string filename); // optional file support
void SetMode (ProgramMode newMode);
ProgramMode GetMode ();
ProgramState GetState ();
bool IsStarved ();
virtual void Stop (bool wait = false);
void SetFifos (CssFifo * enc, CssFifo * dec);
```

d. Audio threads

e. User application selects the flow by linking the FIFOs

```
CssCaptureThread captureThread;
CssDisplayThread displayThread;
Mpeg4Thread mpeg4Thread;
```

Digital Media Solution	
Product Description	07/20/2010
Comm. Sys. Solutions	7 of 8



```

captureThread.SetPriority (-1);
displayThread.SetPriority (-1);
mpeg4Thread.SetPriority (-2);

try {
    std::cout << "Allocating capture frames." << std::endl;
    for (int i = 0; i < 2; i++) {
        int id = cssMemoryManager.Allocate (720 * 480 * 2);
        captureThread.PutBuffer (cssMemoryManager[id]);
        id = cssMemoryManager.Allocate (720 * 480 * 2);
        displayThread.PutBuffer (cssMemoryManager[id]);
    }
    std::cout << "Allocation done." << std::endl;
}
catch (CssMemException memException) {
    std::cout << memException.ReasonString () << std::endl;
    return -3;
}

CssFifo videoEncFifo, videoDecFifo;

captureThread.SetFifo (&videoEncFifo);
mpeg4Thread.SetFifos (&videoEncFifo, &videoDecFifo);
displayThread.SetFifo (&videoDecFifo);

captureThread.Start ();
displayThread.Start ();
mpeg4Thread.Start ();

while (Running) {
    // Application processing
    usleep (10000);
}

```

- f. Container support
 - i. MP4/AVI
 - ii.
- 4. Miscellenous support
 - a. Resizer support

```

class ResizeCopy {

    void ConfigSrc (int width, int height, int pitch);
    void ConfigDst (int width, int height, int pitch);
    void Execute (unsigned long dst, unsigned long src);

```

- b. Command execution

```

static bool DoCommand(std::string command, std::string &result);

```

- c. Watermarking

```

CssWatermark (int width = 720, int height = 480, int bt = 2);

```

- d. Advanced watchdog support

Digital Media Solution	
Product Description	07/20/2010
Comm. Sys. Solutions	8 of 8



Create a watchdog for each process you want to control:

```
CssWatchdog( const char * name );
```

Service the watchdogs in their own threads:

```
void Service( );
```

When all watchdogs have been serviced, the Linux kernel watchdog is then serviced to prevent a system restart. If any one of the watchdogs fail, the system will restart.

- e. Timestamp support.
- f. Pipe support.
- g. ALSA mixer support.

Digital Media Algorithms

Various algorithms are available under the XDM standard from CSS and other 3rd party vendors. These algorithms can be integrated into the release package and used with the standard API. Presently available for inclusion are: acoustic echo cancellation, line echo cancellation, DTMF decoding and generation, sample rate conversion, watermarking, cadenced tone generation/detection, FSK/caller ID, advanced 1-d/2-d analysis, and more.

GUI Framework

A GUI framework is available which provides OSD support, a menu system, and configuration storage. The GUI system is customizable in appearance and icons. It is easily modified and extended.

Demos

Various demos are included in the release package to illustrate the usage of the framework and to serve as starting points for the customer's design.