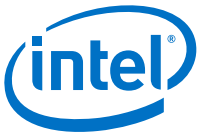
Intel® Embedded Media and Graphics Driver, EFI Video Driver, and Video BIOS v1.16

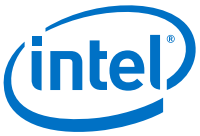
User Guide

October 2012

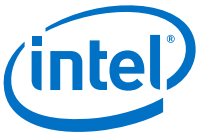
Document Number: 442076-026US



Contents



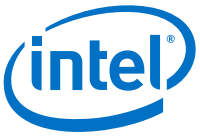
Figures



Tables

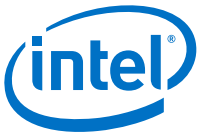
This document may have been updated since the release shown below. See http://edc.intel.com/Software/Downloads/ for the most recent version.

Date	Revision	Description
October 2012	026	Intel® EMGD 1.16 for Windows XP/Linux/Windows Embedded Compact 7 Release
September 2012	025	Intel® EMGD 1.16 for Windows XP/Linux/Windows Embedded Compact 7 Preliminary Release
August 2012	024	Intel® EMGD 1.16 for Windows XP/Linux/Windows Embedded Compact 7 Preliminary Release
April 2012	023	Intel® EMGD 1.14 for Windows XP/Linux/Windows Embedded Compact 7 Release
March 2012	022	Intel® EMGD 1.14 for Windows XP/Linux/Windows Embedded Compact 7 Preliminary Release
January 2012	021	Intel® EMGD 1.10.1 Hotfix for Windows XP/Linux/Windows Embedded Compact 7 Release
November 2011	020	Intel® EMGD v1.10 for Windows XP/Linux Release
October 2011	019	Intel® EMGD v1.10 for Windows XP/Linux Preliminary Release
October 2011	018	Intel® EMGD v1.10 for Windows Embedded Compact 7 Release
September 2011	017	Intel® EMGD v1.8.1 Hotfix Release; updated references to “Fedora Timesys 14” to “'Timesys Fedora Remix 14”
July 2011	016	Intel® EMGD v1.10 for Windows Embedded Compact 7 Preliminary Release
June 2011	015	Intel® EMGD v1.8 for Windows XP/Linux Release
May 2011	014	Intel® EMGD v1.8 for Windows Embedded Compact 7 Preliminary Release
April 2011	013	Intel® EMGD v1.6 for Windows XP and Linux Release
January 2011	012	Intel® EMGD v1.5.3 for Windows CE Release
January 2011	011	Intel® EMGD v1.5.2 for Windows XP and Linux Release



Date	Revision	Description
December 2010	010	Intel® EMGD v1.5 for Windows CE Release
October 2010	009	Intel® EMGD v1.5 for Windows CE Preliminary Release
October 2010	008	Intel® EMGD v1.5 for Windows XP and Linux Release
July 2010	007	Intel® EMGD v1.5 for Windows CE Preliminary Release
July 2010	006	Intel® EMGD v1.0 for Linux Release
June 2010	005	Intel® EMGD v1.0/v1.5 for Linux Beta Release
June 2010	004	Intel® EMGD v1.0 Mid-year Release for Windows XP
June 2010	003	Intel® EMGD EC Release for Linux
June 2010	002	Intel® EMGD Mid-year Release for Windows XP
April 2010	001	Intel® EMGD Beta Release
February 2010	001	Intel® EMGD Alpha Release
November 2009	001	Intel® EMGD Pre-alpha Release

§ §



1.0 Introduction

The Intel® Embedded Media and Graphics Driver (Intel® EMGD) comprises a suite of multi-platform graphics drivers designed to meet the requirements of embedded applications. Featuring Intel® Dynamic Display Configuration Technology (DDCT), the drivers run on the following Embedded Intel® Architecture (eIA) chipsets:

- Intel® Atom™ Processor E6xx (Linux/Windows XP/Windows Embedded Compact 7)
- Intel® System Controller Hub US15W/US15WP/WPT chipset (Linux/Windows XP)

Note: If you need support for a chipset that is not listed above but is in the same family as those listed, please contact your Intel representative.

The Intel® Embedded Media and Graphics Driver supports the following types of display devices:

- Analog CRT (through sDVO)
- LVDS flat panels
- TMDS DVI displays (through sDVO)
- HDMI (through sDVO)
- TV Output (through sDVO)

Intel® EMGD is designed to work with fixed-function systems, such as Point-of-Sale (POS) devices, ATMs, gaming devices, In-vehicle Information/Entertainment systems, etc. It can be configured to work with various hardware and software systems and supports both Microsoft Windows* and Linux* operating systems, including embedded versions of these operating systems.

Intel® EMGD contains a Video BIOS (VBIOS) component. Both Intel® EMGD and the VBIOS component are configurable and work together to provide a wide range of features. This document provides information on configuring and using both the Intel® EMGD and the VBIOS.

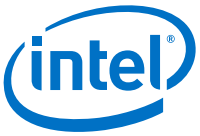
For a list of features supported in this Intel® Embedded Media and Graphics Driver release, please refer to the EMGD Feature Matrix (available on the EDC website) and release notes for details.

1.1 Purpose

This manual provides information on both firmware and software, providing hardware design considerations, installation requirements, and static configuration options.

1.2 Intended Audience

This document is targeted at all platform and system developers who need to interface with the graphics subsystem. This includes, but is not limited to: platform designers, system BIOS developers, system integrators, original equipment manufacturers, system control application developers, as well as end users.



1.3 Related Documents

The following documents provide additional information that may be useful when using the Intel® Embedded Media and Graphics Driver. Additional resources are available at http://edc.intel.com/Software/Downloads/EMGD/.

- *Early Direct Camera Presentation on Intel® Atom™ Processor E6xx Series Application Note* (Document Number: 511280)
- *Intel® Atom™ Processor E6xx B0 Silicon Erratum #9: Clipped SDVO Display on Dual Displays or Sprite Plane-Enabled SDVO Display Frequently Asked Questions* (Document Number: 455133)
- *Display Flickering Sightings and Characterization on Intel® Atom™ Processor E6xx Series (B0-Stepping) White Paper* (Document Number: 324737)
- *Intel® Embedded Graphics Drivers for Embedded Intel® Architecture-based Chipsets Product Brief* (Document Number: 315587)
- *Intel® Atom™ Processor Z5xx Series Datasheet* (Document Number: 319535)
- *Intel® System Controller Hub (Intel® SCH) Datasheet* (Document Number: 319537)
- *Intel® I/O Controller Hub 9 (ICH9) Family Datasheet* (Document Number: 316972)
- *Integrated Dual Independent Display on Intel® Digital Security Surveillance Multifunction Platforms Application Brief*
- *Intel® Embedded Media and Graphics Driver Direct Camera Presentation Interface White Paper* (Document Number: 498844)
- *Display Panel Debugging with the Intel Graphics Memory Controller Hub* (Document Number: 305964)
- *Hardware Accelerated Adobe* Flash* Linux on Embedded Devices* (Document Number: 325389)
- *Intel® Embedded Graphics Drivers, Intel® Embedded Media and Graphics Driver, EFI Video Driver, and Video BIOS API Reference Manual* (Document Number: 498519)
- *VESA BIOS Extensions/Display Data Channel Standard*
  This document provides information on the 4F VBE functions, which are supported by the Intel embedded Video BIOS.
- *VESA BIOS Extension (VBE) Core Functions Standard Version 3.0*
  Contains information on the VESA BIOS Extension (VBE) specification for standard software access to graphics display controllers that support resolutions, color depths, and framebuffer organizations beyond the VGA hardware standard.

Note: The above two documents are available from http://www.vesa.org. Membership may be required to access these documents. Reproductions may also be available from elsewhere on the Internet.

1.4 Conventions

The following conventions are used throughout this document.

Boldface	Represents text that you type and text that appears on a screen.
Italics	Introduces new terms and titles of documents.
`Courier New`	Identifies the names of files, executable program names, and text that appears in a file.
Angle Brackets (< >)	Encloses variable values in syntax or value ranges that you must replace with actual values.
Vertical Bar (|)	Used to separate choices (for example, TRUE | FALSE)

1.5 Acronyms and Terminology

The table below lists the acronyms and terminology used throughout this document.

Table 1. Acronyms and Terminology (Sheet 1 of 4)

Term	Description
ADD Card	AGP Digital Display. An adapter card that can be inserted into the PCIe x16 port of Intel chipset family-based systems. ADD cards allow configurations for TV-out, LVDS, and TMDS output (i.e., televisions, digital displays, and flat panel displays).
AIM	Add In Module.
API	Application Programming Interface.
BDA	BIOS Data Area. A storage area that contains information about the current state of a display, including mode number, number of columns, cursor position, etc.
BIOS	Basic Input/Output System. The Intel® Embedded Media and Graphics Driver interacts with two BIOS systems: system BIOS and Video BIOS (VBIOS). VBIOS is a component of the system BIOS.
BLDK	Boot Loader Development Kit.
CED	Configuration EDitor. Graphical pre-installation utility allows easy creation of consolidated driver installation packages for Windows* and Linux *operating systems, and VBIOS across numerous platforms and display combinations.
Clone Display Configuration	A type of display configuration that drives two display devices, each displaying the same content, but can have different resolutions and (independent) timings. Compare DIH Display Configuration.
Contrast Ratio	Contrast ratio is the measure of the difference between light and dark on a display. If the contrast is increased, the difference between light and dark is increased. So something white will be very bright and something black will be very dark. Brightness and Contrast Controls differ in function between CRTs and LCDs.
COPP	Certified Output Protection Protocol* is a Microsoft-defined API to provide application with information about what output protection options are available on a system.
D3D	Microsoft Direct3D*. A3D graphics API as a component of DirectX* technology.
DC	Display Configuration.
DDCT	Intel® Dynamic Display Configuration Technology.
DirectDraw*	A component of the DirectX* Graphics API in Microsoft Windows OS.

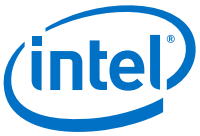


Table 1. Acronyms and Terminology (Sheet 2 of 4)

Term	Description
DIH Display Configuration	Dual Independent Head. A type of display configuration that supports two displays with different content on each display device. The Intel® Embedded Media and Graphics Driver supports Extended mode for Microsoft Windows systems and Xinerama for Linux systems.
DTD	Detailed Timing Descriptor. A set of timing values used for EDID-less devices.
DVI	Digital Video Interface.
DVO	Digital Video Output.
EBDA	Extended BIOS Data Area. An interface that allows the system BIOS and Option ROMs to request access to additional memory.
EDID	Extended Display Identification Data. A VESA standard that allows the display device to send identification and capabilities information to the Intel® Embedded Media and Graphics Driver. Intel® EMGD reads all EDID data, including resolution and timing data, from the display, thus negating the need for configuring DTD data for the device.
EDID-less	A display that does not have the capability to send identification and timing information to the driver and requires DTD information to be defined in the driver.
EFI	Extensible Firmware Interface.
eIA	Embedded Intel® Architecture.
EMI	Electromagnetic Interference.
EPOG	Embedded Pre-OS Graphics feature.
Extended Clone Mode	A feature that allows you to have different sized displays in Clone mode.
Framebuffer	A region of physical memory used to store and render graphics to a display.
GDI	Graphics Device Interface. A low-level API used with Microsoft Windows operating systems.
GMA	Intel Graphics Media Accelerator. Refers to both the graphic hardware in Intel chipsets as well as the desktop/mobile driver. The GMA driver is not intended for use in embedded applications.
GMS	Graphics Mode Select (stolen memory).
HAL	Hardware Abstraction Layer. An API that allows access to the Intel® chipsets.
HDCP	High-bandwidth Digital-Content Protection. A specification that uses the DVI interface. HDCP encrypts the transmission of digital content between the video source (transmitter) and the digital display (receiver).
HDMI	High-Definition Multimedia Interface, an uncompressed, all-digital, audio/video interface.
IAL	Interface Abstraction Layer. An API that allows access to graphics interfaces including the GDI, and DirectDraw*.
iDCT	Inverse Discrete Cosine Transformation (hardware feature).
INF file	A standard Microsoft Windows text file, referred to as an information file, used by Microsoft Windows OS to provide information to the driver. The default .inf file for the Intel® Embedded Media and Graphics Driver is `emgd.inf`. You can create customized parameters using the CED utility.
LPCM	Linear Pulse Code Modulation (LPCM). A method of encoding audio information digitally. The term also refers collectively to formats using this method of encoding.
LVDS	Low Voltage Differential Signaling. Used with flat panel displays, such as a laptop computer display.

Table 1. Acronyms and Terminology (Sheet 3 of 4)

Term	Description
NTSC	National Television Standards Committee. An analog TV standard used primarily in North and Central America, Japan, the Philippines, South Korea, and Taiwan. Its resolutions are based on 525-line systems. Compare PAL.
OAL	Operating system Abstraction Layer. An API that provides access to operating systems, including Microsoft Windows and Linux.
Option ROM (OROM)	Code that is integrated with the system BIOS and resides on a flash chip on the motherboard. The Intel Embedded Video BIOS is an example of an option ROM.
OS	Operating System.
PAL	Phase Alternating Lines. An analog TV standard used in Europe, South America, Africa, and Australia. Its resolutions are based on 625-line systems. Compare NTSC.
PCF	Parameters Configuration File.
PCI	Peripheral Component Interface.
Port Driver	A driver used with the sDVO interfaces of the System Controller Hub (SCH).
POST	Power On Self Test.
PWM	Pulse Width Modulation.
Reserved Memory	A region of physical memory in a Windows Embedded Compact 7 system set aside for BIOS, VBIOS, and graphics driver operations. Reserved memory can be configured for use by the operating system and other applications when not in use by the BIOS.
Saturation	Monitors and scanners are based on the “additive” color system using RGB, starting with black and then adding Red, Green, and Blue to achieve color. Saturation is the colorfulness of an area judged in proportion to its brightness. Full saturation of RGB gives the perception of white, and images are created that radiate varying amounts of RGB, or varying saturation of RGB.
SCART	French Acronym - Syndicat des Constructeurs d'Appareils Radiorecepterus et Televiseurs. A video interface possessing up to four analog signals (Red/Green/Blue/Composite PAL). S-Video (Luma/ Chroma) is possible over the SCART interface as well.
SCH	System Controller Hub.
SCS	Software Compliance Statement.
sDVO	Serial Digital Video Output.
Single Display Configuration	A type of display configuration that supports one and only one display device.
SSC	Spread Spectrum Clock.
Stolen Memory	A region of physical memory (RAM) set aside by the system BIOS for input and output operations. The amount of stolen memory is configurable. Stolen memory is not accessible to the operating system or applications.
System BIOS	The standard BIOS used for basic input and output operations on PCs.
TMDS	Transitioned Minimized Differential Signaling. Used with DVI displays, such as plasma TVs.
TOM	Top Of Memory.
TSR	Terminate and Stay Resident. A program that is loaded and executes in RAM, but when it terminates, the program stays resident in memory and can be executed again immediately without being reloaded into memory.
VBIOS	Video Basic Input Output System. A component of system BIOS that drives graphics input and output.

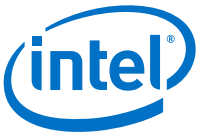


Table 1. Acronyms and Terminology (Sheet 4 of 4)

Term	Description
VESA	Video Electronics Standards Organization.
VEXT Display Configuration	Vertical Extended. A type of display configuration that enables both Primary and secondary displays. Primary and secondary displays can be configured with separate timings. The resolution for the secondary display must be the same as the primary. Content comes from a single framebuffer that spans both displays oriented vertically.
VGA	Video Graphics Array. A graphics display standard developed by IBM* that uses analog signals rather than digital signals.
VLD	Variable Length Decoding.
VMR	Video Mixing Render.
WHQL	Windows* Hardware Quality Labs. WHQL is a testing organization responsible for certifying the quality of Windows drivers and hardware that runs on Windows operating systems.
YUV	Informal, but imprecise reference to the video image format, Y'CbCr. The Y' component is luma, a nonlinear video quality derived from RGB data denoted without color. The chroma components, Cb and Cr, correspond nonlinearly with U and V as differences between the blue and luma, and between the red and luma, respectively.

1.6 Downloading Intel® EMGD and Video BIOS

Download Intel® EMGD and the Video BIOS (VBIOS) from one of the following locations:

- From the Intel Embedded Design Center (http://www.intel.com/p/en_US/embedded/hwsw/software/emgd#download) only, where the following is available:

  Intel Embedded Media and Graphics Driver Configuration Editor (CED) release

  — includes the Intel® EMGD drivers for VBIOS, Linux* OS, and all Windows* operating systems, plus an embedded help system

  — currently runs only on Windows operating systems. If you only require to configure the xorg.conf file, there is a CED Lite tool available that runs on Linux OS. See Section 7.4.3, “Linux* OS Configuration Using CED Lite” on page 138 for more information.

- From the QuAD system: Intel Premier Support (QuAD) (https://premier.intel.com)
- From the new IPS system: https://businessportal.intel.com. You will be redirected to: https://welcome.intel.com/login.aspx where you will need to log in.

  Click on the **Product Support** tab.

  Note: DO NOT use the **Design & Technology** tab, which takes you to the old IPS system.

Note: The Embedded Video BIOS version 1.16 is recommended for use with each of the graphics drivers in most cases. Click the following link to see the FAQ page for details on the differences of these versions.

http://www.intel.com/p/en_US/embedded/hwsw/software/emgd?iid=3747#faqs

After you have downloaded, installed, and run CED, you can configure and customize the drivers and VBIOS following the procedures in this document. After they have been configured, you can integrate the VBIOS with the system BIOS ROM and install Intel® EMGD on your operating system.

§ §

2.0 Architectural Overview

2.1 Introduction

The Intel® Embedded Media and Graphics Driver is composed of a runtime graphics driver and a Video BIOS (VBIOS) firmware component. (See the illustrations below.) Both the driver and VBIOS control the SCH to perform display and render operations. The VBIOS is predominantly leveraged by System BIOS during system boot but is also used at runtime by the driver to handle full-screen text mode on Microsoft Windows* operating systems.

Figure 1. Intel® Embedded Media and Graphics Driver

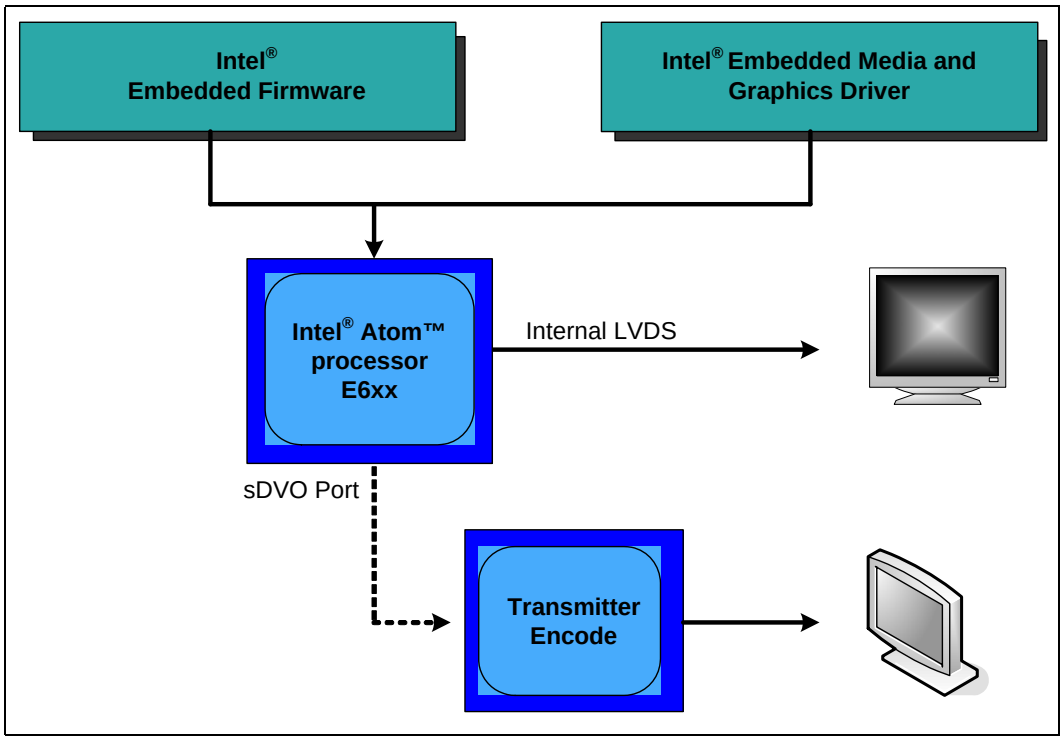


Figure 2. Graphics Driver Architecture

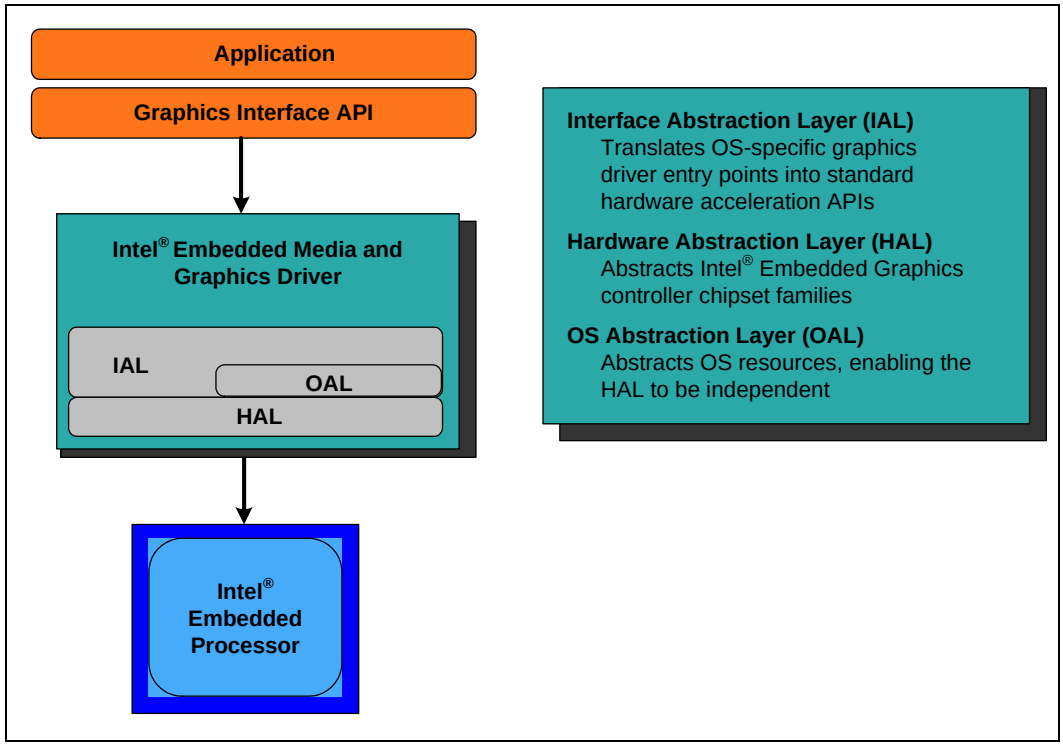
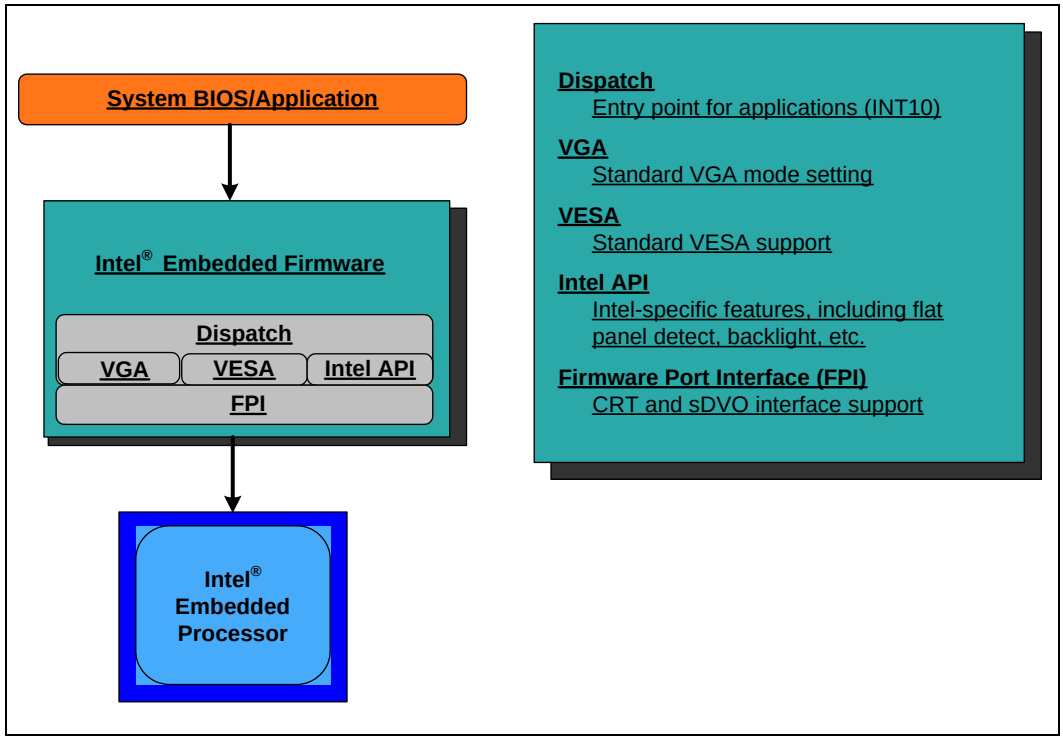


Figure 3. Firmware Architecture



2.1.1 Display Options

The following section describes the types of displays and configurations supported by the Intel® Embedded Media and Graphics Driver.

2.1.1.1 Types of Displays

The table below lists the types of displays supported by the Intel® Embedded Media and Graphics Driver.

Table 2. Types of Displays Supported

Display	Description
CRT	Analog CRT, supported with an external transmitter via an sDVO port.
Flat Panel	TMDS and LVDS compliant flat panels are supported with the use of an external transmitter via an sDVO port. Integrated LVDS flat panels are also natively supported on the Intel® System Controller Hub US15W/US15WP/WPT chipset and Intel® Atom™ Processor E6xx.
TV	TV-out is supported via an external encoder sDVO port. ***Note:*** TV-Out is enabled via supported internal capability, or external transmitters sDVO (where available).

2.1.1.2 Display Configuration

Intel® EMGD supports driving two displays simultaneously. Several configurations are supported, dependent on operating system and chipset. The various display configurations are described in the table below.

Table 3. Display Configuration Definitions

Display Configuration Mode	Description
Single	Normal desktop configuration, single monitor
Clone*	Two displays, same content, different resolutions, independent timings
Extended*	Two displays, continuous content (available in Windows only)
DIH*	Dual Independent Head. Two displays, different content, independent resolutions
* Supported display depends on driver and hardware availability. See the RelNotes.txt for more information.	

The table below summarizes which display configurations are supported by Intel chipsets.

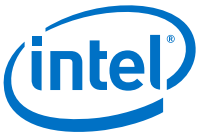


Table 4. Supported Display Configurations

Chipset	Operating System		
	Windows* XP	Windows* Embedded Compact 7 (E6xx only)	Linux*
Intel® Atom™ Processor E6xx	Single, Clone, Extended	Single, Clone, Extended	Single, Clone, Xinerama, DIH ***Note:*** Xinerama is not available for MeeGo.
Intel® US15W/US15WP/ WPT	Single, Clone, Extended	N/A	Single, Clone, Xinerama, DIH ***Note:*** Xinerama is not available for MeeGo.

Intel® EMGD supports Clone mode through custom APIs. In contrast, Microsoft Windows and Linux operating systems (X.Org*) both natively support Extended and DIH.

2.2 Features

The following sections describe major features Intel® EMGD supports.

2.2.1 Chipsets Supported

The table below lists Intel® EMGD-supported chipsets.

Table 5. Chipsets Supported by the Intel® EMGD

Chipset	Intel® EMGD VBIOS Support	Intel® EMGD Support
Intel® Atom™ Processor E6xx	Yes	Yes
Intel® System Controller Hub US15W/ US15WP/WPT chipset	Yes	Yes

All supported chipsets provide for SINGLE LVDS output. In addition, digital monitors, CRTs and TVs are supported through the US15W/US15WP/WPT and MCH/Intel® Atom™ Processor E6xx sDVO interfaces, depending on hardware availability.

2.2.2 OS and API Support

The Intel® Embedded Media and Graphics Driver and Video BIOS support the following operating systems and APIs. Intel® EMGD does not support updating your software past the versions specified here.

- MeeGo *IVI 1.2 (kernel 2.6.37.X, X Server 1.9.3)
- Timesys* Fedora Remix 14 (kernel 2.6.35.6, X Server 1.9.3)
- Microsoft Windows* XP with Service Pack 3, Windows* XP Embedded with Embedded Standard 2009
  - — DirectX* 9.0C (DirectDraw* and Direct3D*)
- API for hardware video decode (DirectShow*)
- OpenGL 2.0, OpenGL ES 1.1, OpenGL ES 2.0, OpenGL 2.1 (Linux), and OpenVG 1.1 (Windows Embedded Compact 7 and Linux)
- Microsoft Windows* Embedded Compact 7 (E6xx only)

Note: The following features are NOT supported in Intel® Embedded Media and Graphics Driver v1.16:

- D3D in Microsoft Windows* Embedded Compact 7

2.2.3 DisplayID Support

The Intel® Embedded Media and Graphics Driver supports the DisplayID specification. DisplayID is a VESA specification (www.vesa.org) that describes the data format for the display configuration parameters and provides the capability to unify the display data structure thereby decreasing the need to rely on proprietary extensions. For more information on DisplayID, its uses and parameters please reference the VESA specification (www.vesa.org).

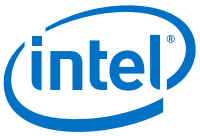
2.2.4 EDID-Less Configuration

EDID-less support is the ability to run a display panel that does not have display timing information within the panel. Therefore, the user has to provide the display timing information to the graphics drivers during configuration using CED. See “Creating a New Customized DTD” on page 23.

This document describes only the necessary edits to the configuration files that are required to implement the graphics driver and VBIOS, and not specific settings for EDID-less panel configuration. Please refer to the manufacturer’s specifications for the DTD settings to use for your EDID-less panels.

2.2.4.1 EDID-Less Panel Type Detection

The Intel® Embedded Media and Graphics Driver supports EDID-less displays that do not export timing modes. This is accomplished by allowing configuration of a Detailed Timing Descriptor (DTD), and associating that DTD with a specific display port.



2.2.5 Rotation

Rotation is the ability to rotate the display for the Intel® Embedded Media and Graphics Driver. Rotation support includes 0°, 90°, 180°, 270°. Rotation is supported only on the following chipsets using Windows* XP, and Linux* operating systems:

- Intel® Atom™ Processor E6xx
- Intel® System Controller Hub US15W/US15WP/WPT chipset

Note: Rotation is not supported with the VBIOS. Rotation is supported with Windows* Embedded Compact 7 but only in static mode.

§ §

3.0 Platform Configuration Using CED

The Intel® EMGD Configuration Editor (CED) is a Windows-based Graphical User Interface (GUI) that allows you to create configurations, package the configurations, and create installations that can be loaded directly on a specific OS or Video BIOS platform. Configurations are associated with a specific chipset and can be created for any one of the following supported chipsets:

- Intel® Atom™ Processor E6xx
- Intel® System Controller Hub US15W/US15WP/WPT chipset

Refer to Section 2.2.2, "OS and API Support" on page 19 for a list of supported operating systems and APIs.

The CED GUI is designed for ease of use and configuration of the Intel® EMGD. Each configuration page has online help available and each data field is validated. If you enter an incorrect value, CED displays an error message at the top of the page and displays the valid range of values for the field. You cannot finish a configuration until all fields contain valid values.

The following sections show how to create a configuration for any of the supported chipsets, operating systems, and the Intel® EMGD Video BIOS.

- "Creating a New Customized DTD" on page 23
- "Creating a New Configuration" on page 27
- "Creating a New Package" on page 44
- "Generating an Installation" on page 53

Note: There are two versions of CED, one for Windows XP/Linux and another for Windows Embedded Compact 7. Not all options covered here may be available, depending on the version of CED you are using.

3.1 Before You Begin

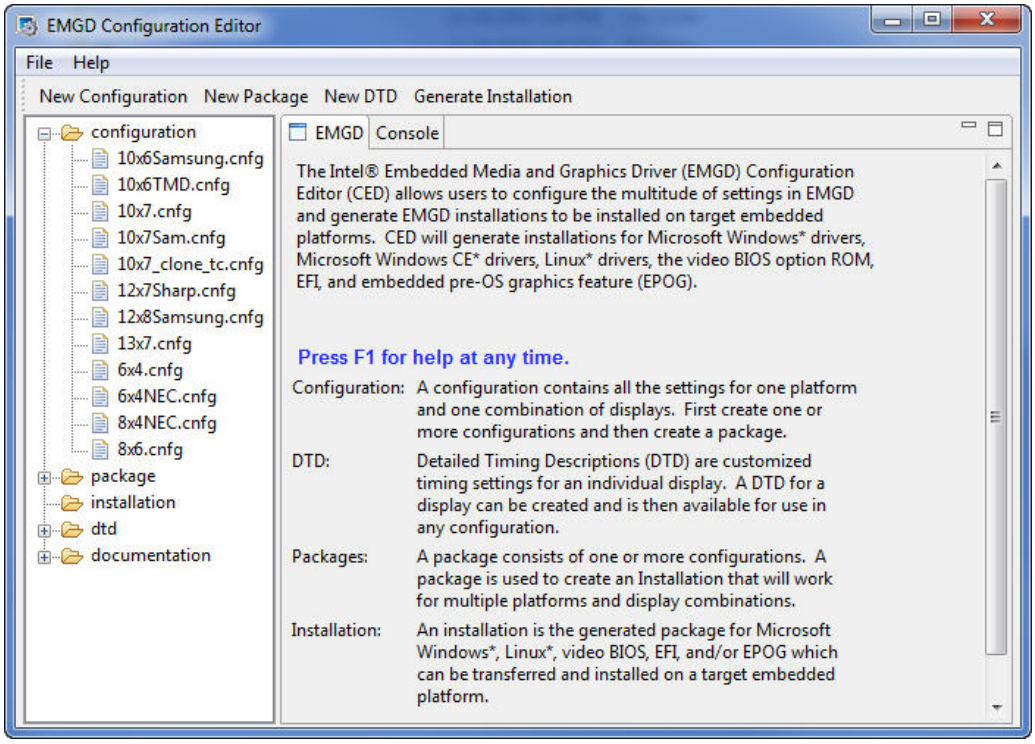
To configure the Intel® EMGD software using CED, you will need some information on the panel you are using. This information is usually found in the product specifications. In some cases the terminology used in CED may not match the labels used in your panel's product specification. Refer to Table 7, "Timing Specification Example Values" on page 26 for hints on which specs correspond to CED Detailed Timings Descriptor (DTD) fields. After you obtain the correct specification values, you may need to derive other values for the DTD fields.

3.2 Creating a Configuration in CED – Summary Steps

The following steps present a sample CED configuration.

1. (Optional) If you have custom panels and timings you may want to create your own DTD; otherwise you can use the standard DTDs provided by CED. If needed, select **New DTD**.
   - Choose the DTD Type that most closely aligns with your display parameters, enter parameters, and then click **Finish**. Or, to create a DTD, see "Creating a New Customized DTD" on page 23.
2. Select **New Configuration**.
   - Enter a name for the configuration, select the mode, chipset, ports, port drivers, DTDs, etc., for the configuration and then click **Finish**. For details, see "Creating a New Configuration" on page 27.
3. Select **New Package**.
   - Enter a name for the package, select the configurations for your package, the platforms for the installation, and then click **Finish**. For details, see "Creating a New Package" on page 44.
4. Select the created package and then select **Generate Installation**.
   The generated files are placed in the installation folder. The zip files (for Linux, Windows, and Windows Embedded Compact 7 operating systems) contain the generated configuration files. For details, see "Generating an Installation" on page 53.

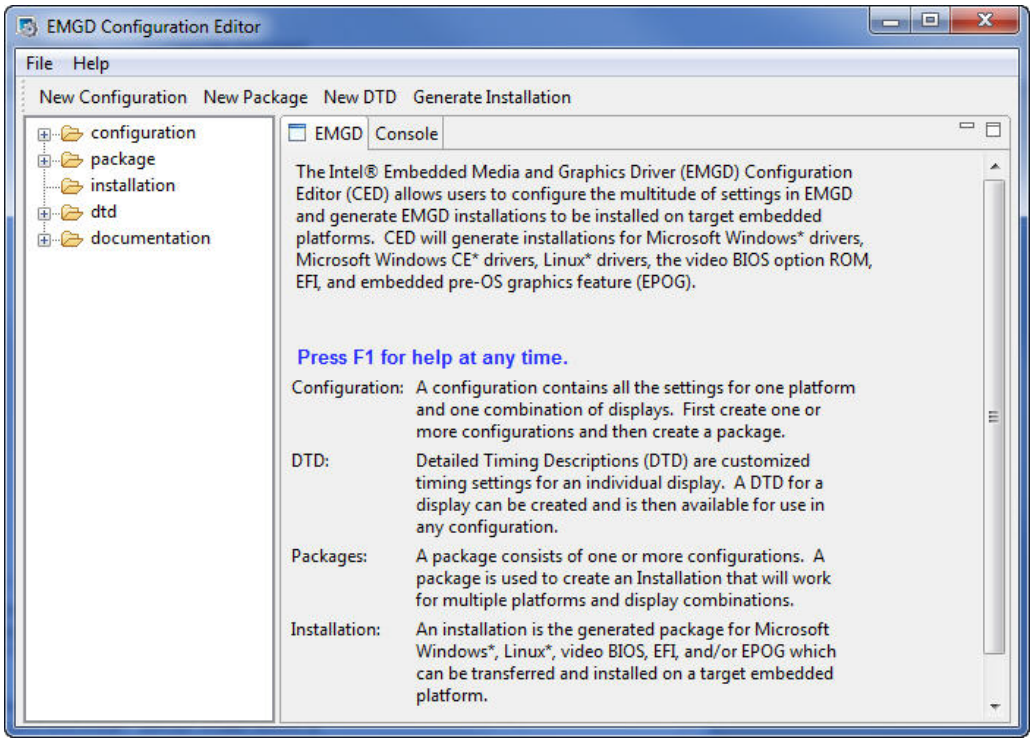
Figure 4. Sample CED Configuration Start Page



3.3 Starting CED

To start the Intel® EMGD CED, open the folder where you installed CED and click the `emgd-ced.exe` icon. The Intel® EMGD CED splash window appears for a few moments followed by the Intel® EMGD Configuration Editor main window.

Figure 5. Intel® EMGD Configuration Editor Main Window



From this window, you can create configurations, package the configurations, and create installations from the packages that can be installed directly on a platform. The main window also provides a Console tab that displays information when you build a package or an installation.

The following sections show how to create a configuration for any of the supported chipsets, operating systems, and the Intel® EMGD Video BIOS.

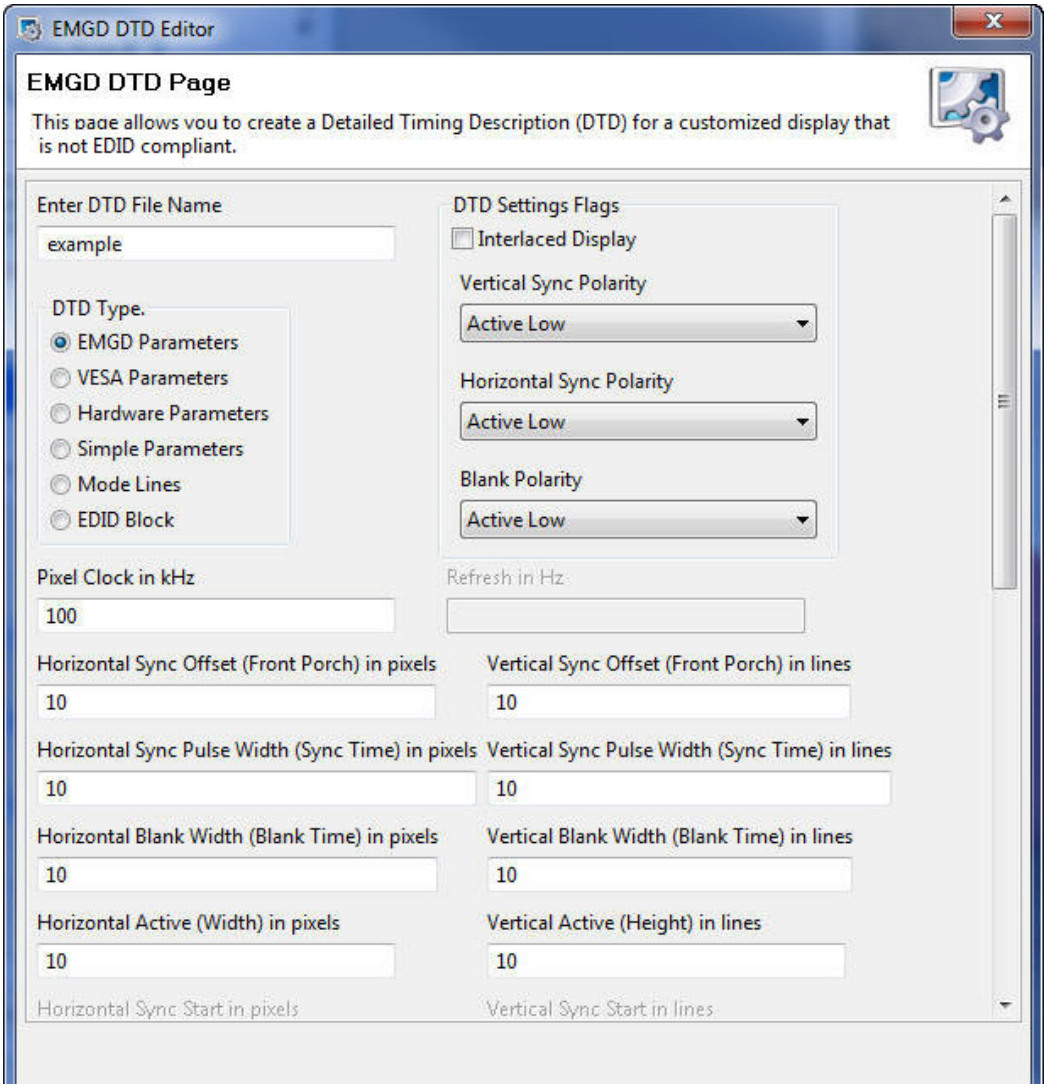
3.4 Creating a New Customized DTD

CED allows you to create Dynamic Timings Definitions (DTD) for EDID-less displays or displays for which you do not want to use the display's EDID settings. In either of those cases, you can create your own DTD using the steps below. Otherwise you can use one of the standard DTDs included in CED.

You can create a new DTD by clicking the **New DTD** link at the top of the main CED window, or you can create DTDs for each configured port when you create a new configuration. Any DTDs you create will be available for all configurations.

When you select **New DTD** from the main CED window, the following Intel® EMGD DTD Page appears.

Figure 6. EMGD DTD Page



To create a custom DTD setting:

1. From the CED main screen, select **New DTD**.
2. Enter a name for the DTD in the text box provided, for example, *test_LVDS*.
3. Using the data sheet from the panel being used, enter the DTD timings in the appropriate fields. Refer to Table 6, "Intel® EMGD DTD Setting Options" for field descriptions.
   The screen will be similar to the example shown in Figure 6.
4. Click **Finish**.
   The custom DTD is complete.

Table 6. Intel® EMGD DTD Setting Options (Sheet 1 of 2)

DTD Parameter	Description
Enter DTD File Name	Enter a name for this customized DTD. This is a required field and the name must be between 1 and 50 characters and may contain spaces and underscores.
DTD Type	Select the DTD Type that most closely aligns with your display parameters. Options are: • Intel® EMGD **Parameters**: The Intel® EMGD Parameters are the same as the current PCF/CED DTD parameters. • **VESA Parameters**: The VESA Parameters allow the user to create a DTD from a VESA monitor timing standard. • **Hardware Parameters**: The Hardware Parameters are the parameters that are used by Intel® EMGD. • **Simple Parameters**: The Simple Parameters (CVT Standard) is a process for computing standard timing specifications. The method for developing Reduced Blanking timings is not included. • **Mode Lines**: The Mode Lines are a video timing spec used by X.Org. The X.Org timing setting for Mode Lines is “name” I A B C D E F G H. For example: “640x480@8bpp” 25.175 640 672 728 816 480 489 501 526. • **EDID Block**: The EDID Block is the detailed timing section (18 bytes) of the basic 128-byte EDID data structure. The detailed timing section starts at 36h of the 128-byte EDID data structure. Enter the EDID block 1 byte at a time. Example: a0 0f 20 00 31 58 1c 20 d2 1a 14 00 f6 b8 00 00 00 18
Pixel Clock	Pixel clock value in KHz. Range 0-0x7fffffff.
DTD Settings Flags	This section allows you to set flags for Interlace, Vertical Sync Polarity, Horizontal Sync Polarity, and Blank Sync Polarity. Each field in this section is described below. **Interlaced Display**: • Check for Interlaced • Cleared for Non-interlaced **Vertical Sync Polarity**: • Active Low (Default) • Active High **Horizontal Sync Polarity**: • Active Low (Default) • Active High **Blank Sync Polarity**: • Active Low (Default) • Active High ***Note:*** These flags are Intel® EMGD-specific and do not correspond to VESA 3.0 flags.
Horizontal Sync Offset (Front Porch) in pixels	Specifies the amount of time after a line of the active video ends and the horizontal sync pulse starts (Horizontal Front Porch). Range 0-1023 [10 bits].
Horizontal Sync Pulse Width (Sync Time) in pixels	Width of the Horizontal Sync Pulse (Sync Time) which synchronizes the display and returns the beam to the left side of the display. Range 0-1023 [10 bits].
Horizontal Blank Width (Blank Time) in pixels	This parameter indicates the amount of time it takes to move the beam from the right side of the display to the left side of the display (Blank Time). During this time, the beam is shut off, or blanked. Range 0-4095 [12 bits].
Horizontal Active (Width) in pixels	Number of pixels displayed on a horizontal line (Width). Range 1-32767 [15 bits].

Table 6. Intel® EMGD DTD Setting Options (Sheet 2 of 2)

DTD Parameter	Description
Horizontal Sync Start in pixels	This parameter specifies the start of the horizontal active time. Range 0-40957.
Horizontal Sync End in pixels	This parameter specifies the end of the horizontal active time. Range 0-49148.
Horizontal Blank Start in pixels	This parameter specifies the start of one line of the video and margin period. Range 0-32766.
Horizontal Blank End in pixels	This parameter specifies the end of one line of the video and margin period. Range 0-65533.
Refresh in Hz	Also known as the Vertical Refresh, the rate the full display updates. Standard refresh rates are 50Hz, 60Hz, 75Hz, and 85Hz.
Vertical Sync Offset (Front Porch) in lines	Specifies the amount of time after last active line of video ends and vertical sync pulse starts (Vertical Front Porch). Range 0-4095 [12 bits].
Vertical Sync Pulse Width (Sync Time) in lines	Specifies the Width of the Vertical Sync Pulse which synchronizes the display on the vertical axis and returns the beam to the top, left side of the display. Range 0-63 [6 bits].
Vertical Blank Width (Blank Time) in lines	The amount of time for the complete vertical blanking operation to complete. It indicates the time it takes to move the beam from the bottom right to the top, left side of the display (Blank Time). During this time, the beam is shut off, or blanked. Range 0-4095 [12 bits].
Vertical Active (Height) in lines	The number of active lines displayed (Height). Range 1-4095 [12 bits].
Vertical Sync Start in lines	This parameter specifies the start of the vertical sync. Range 0-4157.
Vertical Sync End in lines	This parameter specifies the end of the vertical sync. Range 0-4220.
Vertical Blank Start in lines	This parameter specifies the start of display vertical blanking including margin period. Range 0-4094.
Vertical Blank End in lines	This parameter specifies the end of vertical blanking. Range 0-8189.

3.4.1 DTD Example Specifications

The following table shows example product specifications that can be used in the timing fields.

Table 7. Timing Specification Example Values (Sheet 1 of 2)

Item		Symbol	Standard value			Unit
			Min.	Typ.	Max.	
Clock	Frequency	1/ts	29.91	33.231	36.55	MHz
	Period	ts	27.36	30.06	33.43	ns
	Hi-time	tsh	7	–	–	ns
	Low-time	tsl	7	–	–	ns
	DUTY ratio	th/tl	35	50	65	ns
Data	Setup time	tds	7	–	–	ns
	Hold time	tdh	4	–	–	ns
H sync.	Period	tlpl, tlpd	24.51	31.75	32.05	us
			880	1056	1088	clk
	Pulse width	tlw	3	128	200	clk
H display	Term	thd	800	800	800	clk
Enable	Setup time	tdrs	7	–	–	ns
	Hold time	tdrh	4	–	–	ns

Table 7. Timing Specification Example Values (Sheet 2 of 2)

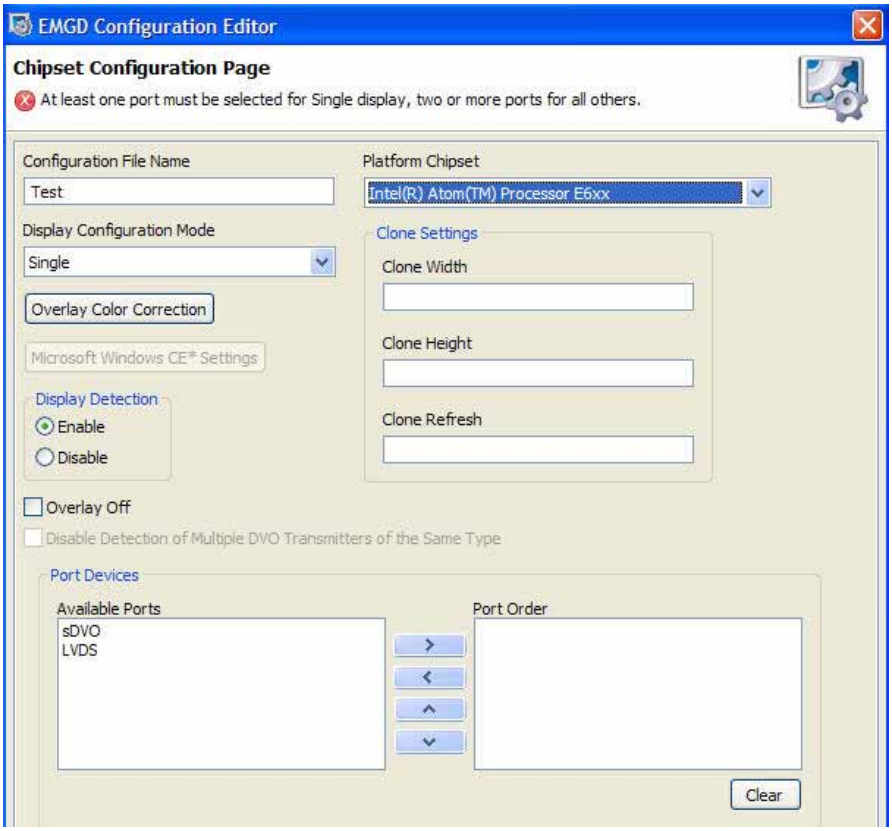
Item		Symbol	Standard value			Unit
			Min.	Typ.	Max.	
V sync.	Period	tfpf, tfpd	520	525	680	Line
	Pulse width	tfw	1	2	3	Line
V display	Term	tvd	480	480	480	Line
	Start	tfd	10	33	40	Line
Phase difference	H sync. ~ enable	tdrds	50	216	260	clk
	H sync. ~ clock	tls	7	–	–	ns
	H sync. ~V sync.	tn	7	–	–	ns

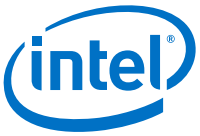
For information about creating DTDs for Windows Embedded Compact 7, see Chapter 6.0, "Configuring and Building Intel® EMGD for Microsoft Windows* Embedded Compact 7."

3.5 Creating a New Configuration

To create a new configuration, click the **New Configuration** selection located on the top of the Intel® EMGD CED main window. The Chipset Configuration Page appears, as shown in the next figure.

Figure 7. Chipset Configuration Page





The Chipset Configuration Page allows you to specify settings that apply to all OS, VBIOS, EFI, and EPOG platforms (Note: The EPOG feature is available only in single display mode.)

The table below describes each setting on the Chipset Configuration page.

Table 8. Chipset Configuration Page Settings (Sheet 1 of 2)

Setting	Description
Configuration File Name	Provide a name for the configuration you are creating. This name is required and is used when you create packages. The name can consist of any alphanumeric characters and any special characters and must be between 1 and 50 characters. You must enter a configuration before you can enter any other information on this page.
Platform Chipset	Select the target chipset for this configuration from the drop-down list.
Display Configuration Mode	Select the type of display configuration from the drop-down list. You can select any one of the following display configurations: • Single — Single display configuration. • Clone — Two displays where both displays have the same content but can have different resolutions and timings. • Vertical Extended — Two displays where primary and secondary displays can be configured with separate timings. Note the resolution for the secondary must be the same as the primary. Content comes from a single framebuffer that spans both displays oriented vertically. This option is available only for MeeGo OS. • DIH — Dual Independent Head. This is a configuration where both displays can have different resolutions, different refresh rates, and different content. ***Note:*** On Microsoft Windows* DIH configurations, the display DOES NOT automatically come up in extended display mode. You must go into the Display properties on the Control Panel and manually set the display to DIH mode.
Overlay Color Correction	Overlay Color Correction allows the Overlay plane to have color-correction settings that are different from the main frame buffer color-correction settings. See "Overlay Color Correction" on page 29.
Display Detection	Display Detection allows you to specify if the driver should detect displays on the system. The default is Disabled. For more information on Display Detection, refer to "Display Detection and Initialization" on page 65.
Port Devices (Available Ports, Port Order)	The Port Devices section lists the ports available based on the chipset selected. The Available Ports box lists the ports available to the chipset. You can move these port devices to the Port Order box to determine the search order for detecting attached displays. To move a port device to the Port Order box, either double-click the port device or click the port device to highlight it, and then click the right arrow button to move it from the Available Ports to the Port Order box. The Port Order section allows you to determine the search order for detecting attached displays for the Display Detection feature. When Display Detection is enabled, the Port Order determines which display is primary and which display is secondary. You can choose default ordering by not moving any of the Available Ports to the Port Order box and leaving the Port Order box empty. Default ordering is chipset-specific. See Table 41, "Default Search Order" on page 188 for more information on default port ordering based on chipset. When you move one or more ports to the Port Order box, you can configure each port by clicking **Next**. For each port listed in the Port Order box, you can click **Next** to configure each port. See "Configuring Ports" on page 31 for information on configuring ports.

Table 8. Chipset Configuration Page Settings (Sheet 2 of 2)

Setting	Description
Clone Settings Clone Width Clone Height Clone Refresh	If you are creating a clone display configuration, you can specify the width, height, and refresh rate for the clone display in this section. For more information about clone display configurations, refer to "Enhanced Clone Mode Support" on page 71.
Overlay Off	This field allows you disable Overlay support, which is enabled by default. ***Note:*** This field is only for Microsoft Windows* operating systems. The Linux* OS configuration for the xorg.conf provides a standard option that performs the same function.

3.5.1 Setting Color Correction

Color Correction is available for both overlays and framebuffers, and is accessed under the **New Configuration** link at the top of the main CED window. For both overlay and framebuffer color correction, user-assigned values must be between 0.6 to 6. By default, gamma is 1.0 (no correction).

3.5.1.1 Overlay Color Correction

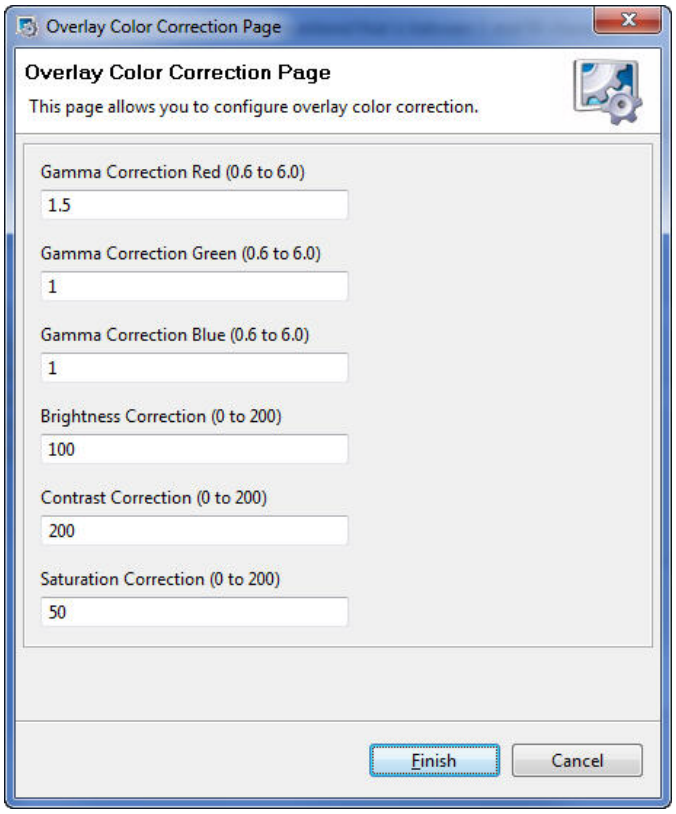
Overlay Color Correction allows the Overlay plane to have color-correction settings that are different from the main framebuffer color-correction settings. This feature lets you color-correct for red, green, and blue, plus it enables you to adjust brightness, contrast, and saturation.

Table 9. Overlay Color Correction Values (applies to ALL color)

Gamma:	0.6 to 6.0 (default value is 1)
Brightness:	0 to 200 (default value is 100)
Contrast:	0 to 200 (default value is 100)
Saturation:	0 to 200 (default value is 100)

To assign overlay color correction, click the **Overlay Color Correction** button on the Chipset Configuration Page. The Overlay Color Correction Page appears, as shown in the next figure.

Figure 8. Overlay Color Correction Page



Add your desired values to the correction fields and then click **Finish**.

3.5.1.2 Framebuffer Color Correction Attributes

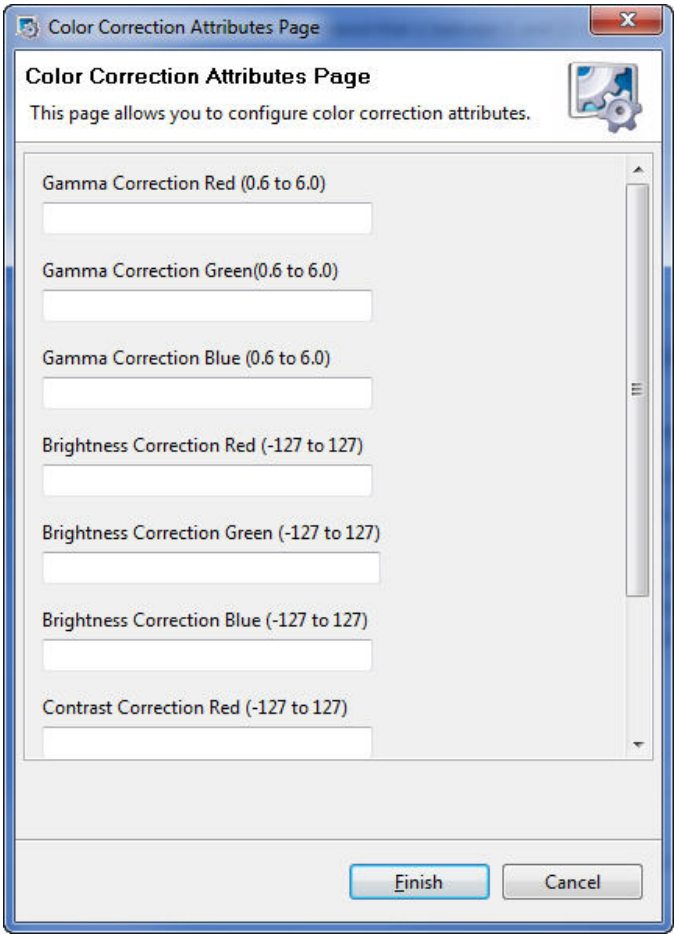
Framebuffer Color Correction Attributes lets you adjust the main color attributes. This feature allows you to color-correct for red, green, and blue, and enables you to adjust brightness and contrast.

Table 10. Framebuffer Color Correction Values (applies to R, G, B color)

Gamma:	0.6 to 6.0 (default value is 1)
Brightness:	-127 to 127 (default value is 0)
Contrast:	-127 to 127 (default value is 0)

To assign framebuffer color correction, click the **Framebuffer Color Correction Attributes** button on the port configuration page (LVDS or sDVO). The Framebuffer Color Correction Page appears, as shown in Figure 9.

Figure 9. Framebuffer Color Correction Page

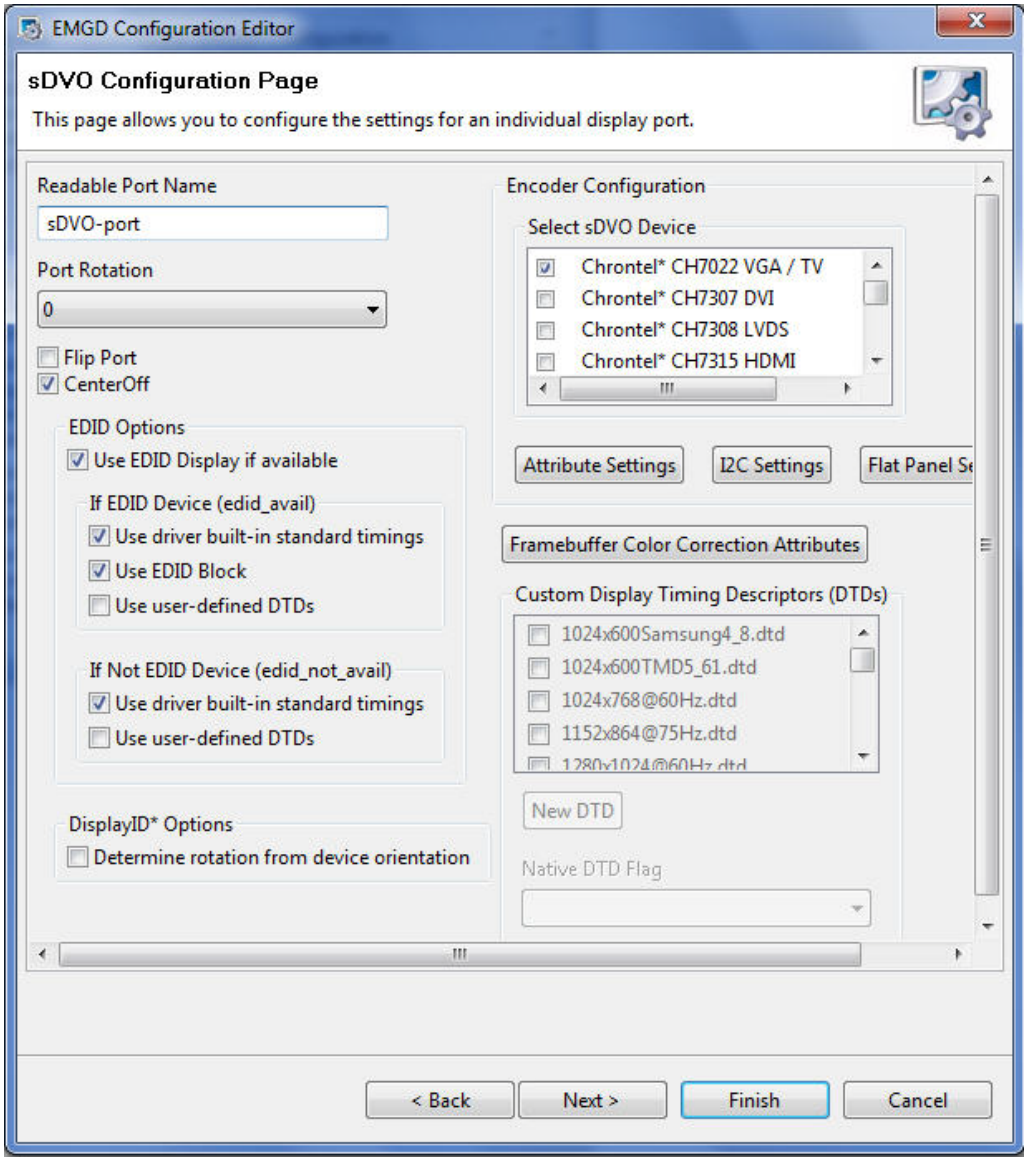


Add your desired values to the correction fields and then click **Finish**.

3.5.2 Configuring Ports

You can configure each port listed in the Port Order box of the Chipset Configuration Page by clicking **Next**. When you do, a port Configuration Page appears similar to the one shown following.

Figure 10. Port Configuration Page



The Port Configuration Page allows you to specify whether to use EDID timings or customized DTD timings for the display connected to this specific port. From this page, you can also specify Attribute Settings, I2C Settings, and Flat Panel Settings and create a new DTD that can be used with any configuration.

Table 11 describes each field on this page.

Table 11. Port Configuration Settings (Sheet 1 of 2)

Port Configuration Field	Description
Readable Port Name	Enter a name for the port. This is a required field and the name must be between 1 and 50 characters and may contain spaces.
Port Rotation	This list allows you select a rotation for the display connected to this port. You can choose between 0, 90, 180, and 270 degrees. The default is 0.
Flip Port	Check this box if you want the display connected to this port to be inverted horizontally. The default is not to invert horizontally.
CenterOff	When this option is enabled it DISABLES centering. Also, depending on the combination of “edid” + “user-dtd” + connected hardware, Intel® EMGD will add missing compatibility modes (6x4, 8x6, 10x7& 12x10) via centering. Use this option to disable this feature.
EDID Options	This section allows you to set EDID options for the display. The Intel® EMGD supports three different types of EDID display modes: • Built-in display modes: These modes are hard-coded in the Intel® EMGD. These modes can be filtered based on the EDID block. • EDID Block: These are Detailed Timing Descriptors read from an EDID display. An EDID display can contain DTD as well as other information about the display. • User-specified DTDs. If you want to use the display's EDID information if it is available, click the **Use EDID Display if Available** check box. If the display attached to this port contains EDID information, you can choose one or more of the following options from the If EDID Device section to determine which set of timings to use for the display connected to the port: • Use driver built in standard timings — If this box is checked, the standard timings built into the Intel® EMGD are used. • Use EDID block — If this box is checked, the EDID block is used. • Use user-defined DTDs — If this box is checked, a user-defined DTD is used. You can select which DTD to use by checking the appropriate box in the Custom Display Timings Descriptors (DTDs) section. If no DTDs are defined, you can click **New DTD** and create a custom DTD. For information on creating custom DTD, refer to Table 18, “Windows OS Setting Options” on page 49. If you select both **Use driver built-in standard timings** and **Use EDID block**, the Intel® EMGD uses its built-in display timings and the timings provided by the display. If the display attached to this port does not contain EDID information, you can choose one or both of the following options from the If Not EDID Device section: • Use driver built-in standard timings — If this box is checked, the standard timings are used. • Use user-defined DTDs — If this box is checked, a user defined DTD is used. You can select which DTD to use by checking the appropriate box in the Custom Display Timings Descriptors (DTDs) section. If no DTDs are defined, you can click **New DTD** and create a custom DTD. For information on creating custom DTD, refer to Table 18, “Windows OS Setting Options” on page 49. See “Sample Advanced EDID Configurations” on page 68 for example configurations.
DisplayID Options	This option provides flexibility to enable/disable display rotation configuration based on DisplayID file. To enable display rotation configuration based on DisplayID file, please ensure you do following: • Check the **Determine rotation from device orientation** option box • Leave the Port Rotation and Flip Port options under CED LVDS or sDVO configuration page unchecked or set to 0. • Check the **Use EDID display if available** option box and under “If EDID Device (edid_avail)” check the **EDID block** option for the graphics driver to get the rotation information from DisplayID file. ***Note:*** If you set the Port Rotation and Flip Port option in CED to a non-zero value, then the graphics will use your setting regardless of the DisplayID option enabled.

Table 11. Port Configuration Settings (Sheet 2 of 2)

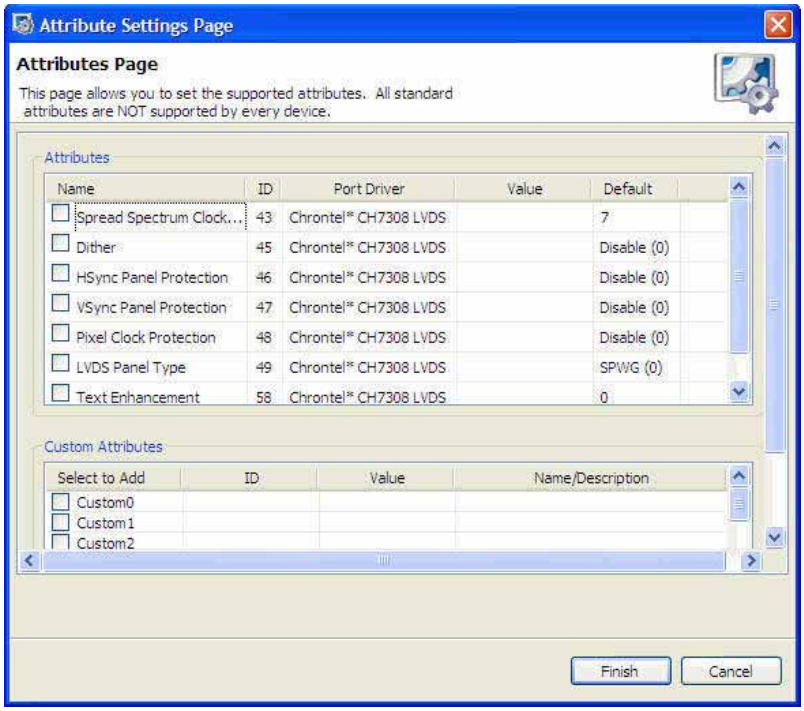
Port Configuration Field	Description
Encoder Configuration	This section lets you to specify the type of encoder connected to an sDVO port and encoder Attributes, I2C settings, and Flat Panel settings for the port. The **Select sDVO Device** drop-down list contains the list of all supported sDVO devices. Select the device that will be connected to this port. To change the device's attributes, click the **Attribute Settings** button. Refer to "Changing Port Attribute Settings" for information on device attributes. To change the device's I2C settings, click the **I2C Settings** button. See "Changing I2C Settings" on page 35 for information on I2C settings. To change the device's flat panel settings, click the **Flat Panel Settings** button. See "Changing Flat Panel Settings" on page 36 for information for changing flat panel settings.
Framebuffer Color Correction Attributes	Framebuffer Color Correction Attributes allow you to adjust the main Frame Buffer color attributes. See "Framebuffer Color Correction Attributes" on page 30.
Native DTD Flag	The Native DTD list lets you choose whether to use a display's built-in timings.

3.5.2.1 Changing Port Attribute Settings

When you click the **Attributes Settings** button from the Encoder Configuration section of the Port Configuration Page, CED displays a page of attributes for the selected encoder device. The actual page that appears depends upon the encoder device selected and only the attributes that apply to the selected encoder appear. For a full description of all attributes for all supported encoders, refer to Appendix B, "Port Driver Attributes".

Figure 11 shows a sample Attributes Settings Page for the Chrontel CH7022, CH7307, and CH7308 encoders.

Figure 11. Attribute Settings Page for the Chrontel CH7022/CH7307/CH7308 Encoders



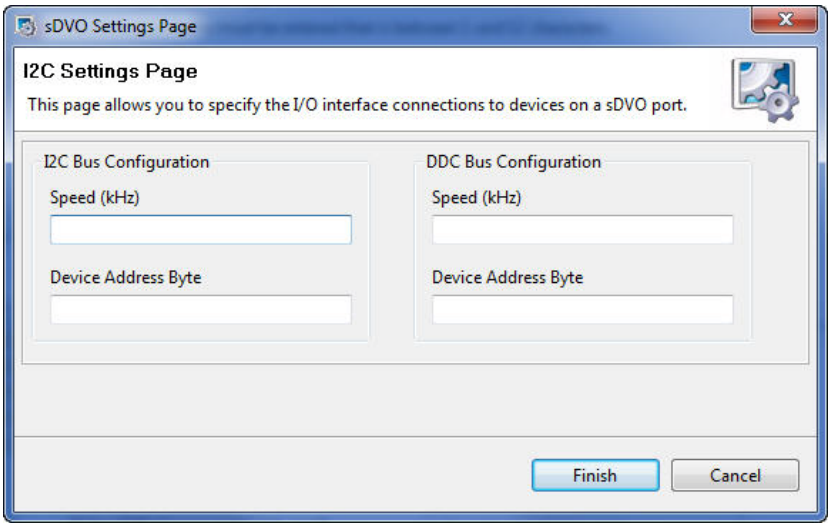
When the Attributes Settings Page first appears, it shows the **Use Default** box checked for all attributes.

To change a default value, clear the **Use Default** check box and enter a new value. For a description of all attributes for all supported encoders, see Appendix B, "Port Driver Attributes".

3.5.2.2 Changing I2C Settings

The I2C Settings Page allows you to specify the I/O interface connections to devices on an sDVO port. When you click **I2C Settings** from the Port Configuration Page, the following screen appears.

Figure 12. sDVO Settings Page



The following table describes each field on this page.

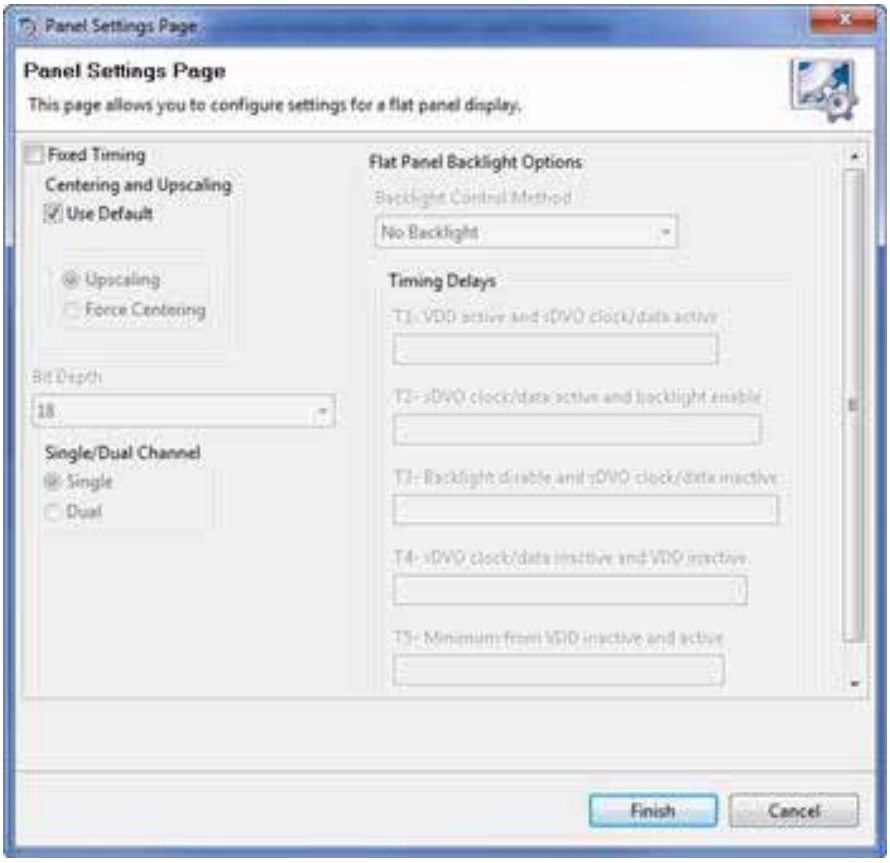
Table 12. I2C Settings

I2C / DDC Bus Configuration	Description
Speed (KHz)	Speed of I2C bus for the device and for the EDID device. The range for these two fields is 10-400 KHz.
Device Address Byte	Enter a device address byte for the device that this port is connected to in these boxes: • The I2C device address is for reading and writing device registers. The device address byte must be in 8-bit format with the 7-bit slave address assigned to its bits 7:1 and bit 0 set to 0. • The DDC Device Address Byte is the I2C device address for reading EDID data from the display through the DDC bus.

3.5.2.3 Changing Flat Panel Settings

The Panel Settings Page allows you to specify settings for a flat panel display connected to this sDVO port. When you click **Flat Panel Settings** from the Port Configuration Page, the following screen appears.

Figure 13. Panel Settings Page



The table below describes each section of this page.

Table 13. Panel Settings Options (Sheet 1 of 2)

Flat Panel Settings	Description
Fixed Timing	To use fixed timing for the attached display, select this option.
Centering and Upscaling	The **Use Default** check box lets you choose the default setting or either Upscaling or Force Centering.
Bit Depth	This list lets you select a color depth for the panel. You can choose either 18 or 24 bit color depth. The default is 18.
Flat Panel Backlight Options	This section provides options for controlling the backlight of the flat panel display and specifying timing delays. • The Backlight Control Methods list lets you choose either **No Backlight** or **Port Driver** to control the backlight. If you choose Port Driver, GMCH, or ICH, you can specify the timing delays in the Timing Delays section and the GPIO pin connections in the GPIO Pin Connections section. The default is No Backlight.

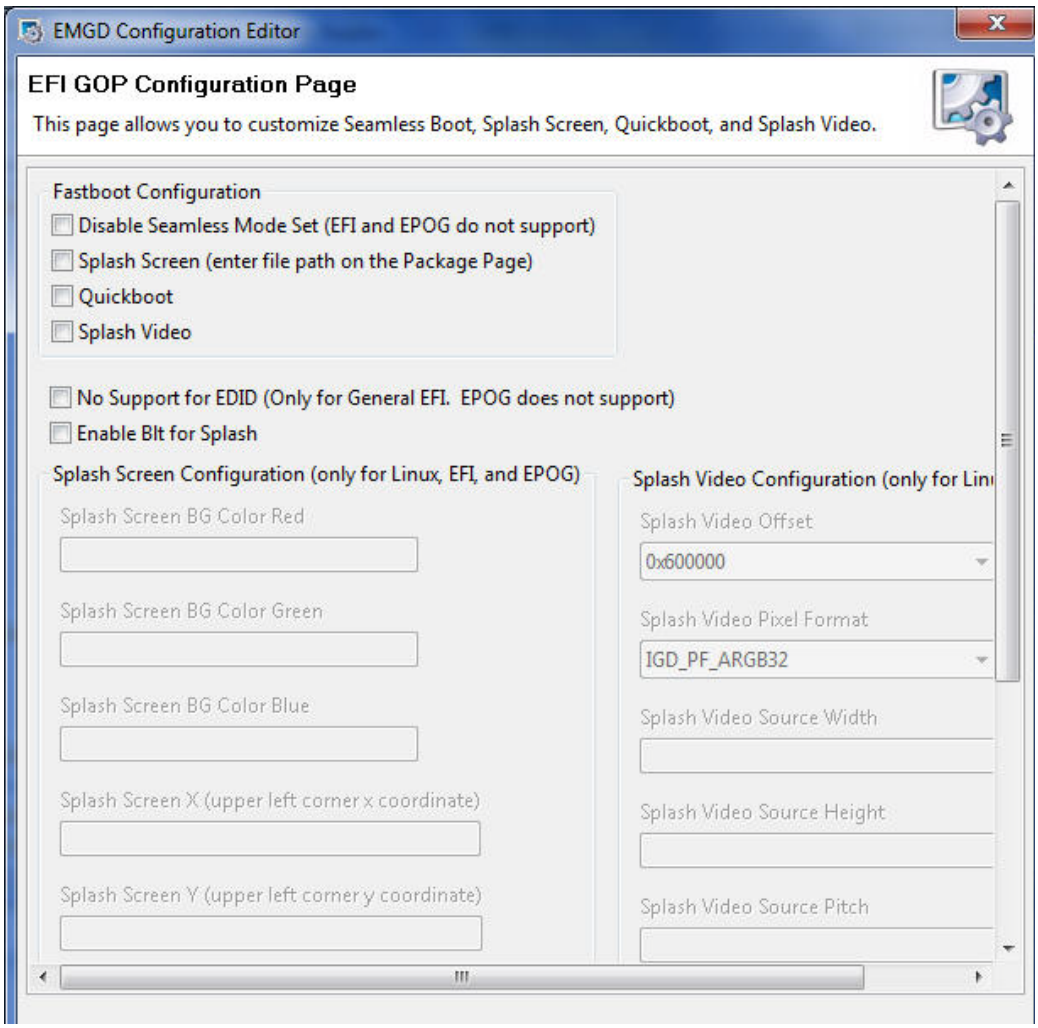
Table 13. Panel Settings Options (Sheet 2 of 2)

Flat Panel Settings	Description
Timing Delays	This section lets you specify timing delays for the backlight signals as follows: • T1-VDD active and sDVO clock/data active: 1-512, increment by 1. • T2-DVO active and backlight enable: 2-256, increment by 2. • T3-Backlight disable and DVO clock/data inactive: 2-256, increment by 2. • T4-DVO clock/data active and inactive: 1-512, increment by 1. • T5-Minimum from VDD inactive and active: 1-1600, increment by 50. ***Note:*** Timers are very specific to the panel you are using. If they are set incorrectly the display can be damaged or ruined. Please refer to the datasheet for your display to determine the correct settings.

3.5.3 Configuring Fastboot

Note: Intel® EMGD supports EDK and EDK II.

Figure 14. EFI GOP Configuration Page



The table below describes each section of this page. Note that fastboot is applicable only for Linux.

Note: Enter the file path for the splash video on the Package Page. See Figure 17, "Intel® EMGD Package Editor Page" on page 45.

Table 14. Fastboot Options (Sheet 1 of 2)

Fastboot Settings	Description
Enable Seamless Mode	The Seamless Mode setting ensures that on a properly configured embedded device there is only one mode set during EFI to DRM transition to avoid screen flicker and undesirable latency. This setting is applicable only to seamless switches from Clone mode to VEXT mode; refer to Section 7.4.18, "Seamless Switches from Clone Mode to VEXT Mode" on page 161 for details.
Splash Screen	The Splash screen feature provides a user-configurable splash screen image that is loaded to the framebuffer at the earliest possible time by the EPOG feature and EFI graphics driver and remains in place until overwritten by the OS or driver. Additionally the Intel® EMGD can be configured to suppress OS drawing to the on-screen framebuffer until notified by an application. Instead, drawing is redirected to an off-screen framebuffer. When notified by the application, the Intel® EMGD will flip the already prepared off-screen framebuffer to be on-screen and cease redirection of drawing. In this manner the configured splash screen will be displayed early during boot and remain in place until a time when the OS is fully loaded and the application interface has been prepared. The splash screen is limited to 500 KB in size and JPG and BMP formats. For Quickboot, only BMP format is allowed.
Quickboot	The quickboot feature optimizes the speed that Intel® EMGD loads at the expense of compatibility and ease of use. Quickboot disables non-critical features that affect the initialization time of the driver that are not needed for targeted embedded applications. For example, there is no port detection; it supports only an LVDS interface.
Splash Video	The Splash Video feature provides a mechanism to use a portion of the off-screen pre-allocated video memory ("Stolen Memory") as a video image that is displayed on an overlay to the framebuffer. The intention is that a video capture device external to Intel® EMGD will be configured to transfer a video stream to the configured location in video memory using DMA. The splash video remains in place until the Intel® EMGD is notified by an external application to disable the overlay.
No Support for EDID (Only for General EFI. EPOG does not support)	This feature provides an option to skip checking the EDID for optimizing the boot time. ***Note:*** It is not applicable to EPOG.
Enable BLT for Splash	This option enables the BLT function when a splash screen is enabled. If you are experiencing problems where the splash screen does not disappear after boot, make sure this option is enabled.
Splash Screen BG Color Red (EFI only)	Splash Screen BG Color Red must be between 0x0 and 0xFF.
Splash Screen BG Color Green (EFI only)	Splash Screen BG Color Green must be between 0x0 and 0xFF.
Splash Screen BG Color Blue (EFI only)	Splash Screen BG Color Blue must be between 0x0 and 0xFF.
Splash Screen X (upper left corner x coordinate) (EFI and EPOG feature only)	The X location, in pixels, where the Firmware Splash Screen will be placed. This number is a signed number in 2's complement. Positive numbers are offset from the left of the screen. Negative numbers are offset from the right of the screen.
Splash Screen Y (upper left corner y coordinate) (EFI and EPOG feature only)	The Y location, in pixels, where the Firmware Splash Screen will be placed. This number is a signed number in 2's complement. Positive numbers are offset from the top of the screen. Negative numbers are offset from the bottom of the screen.

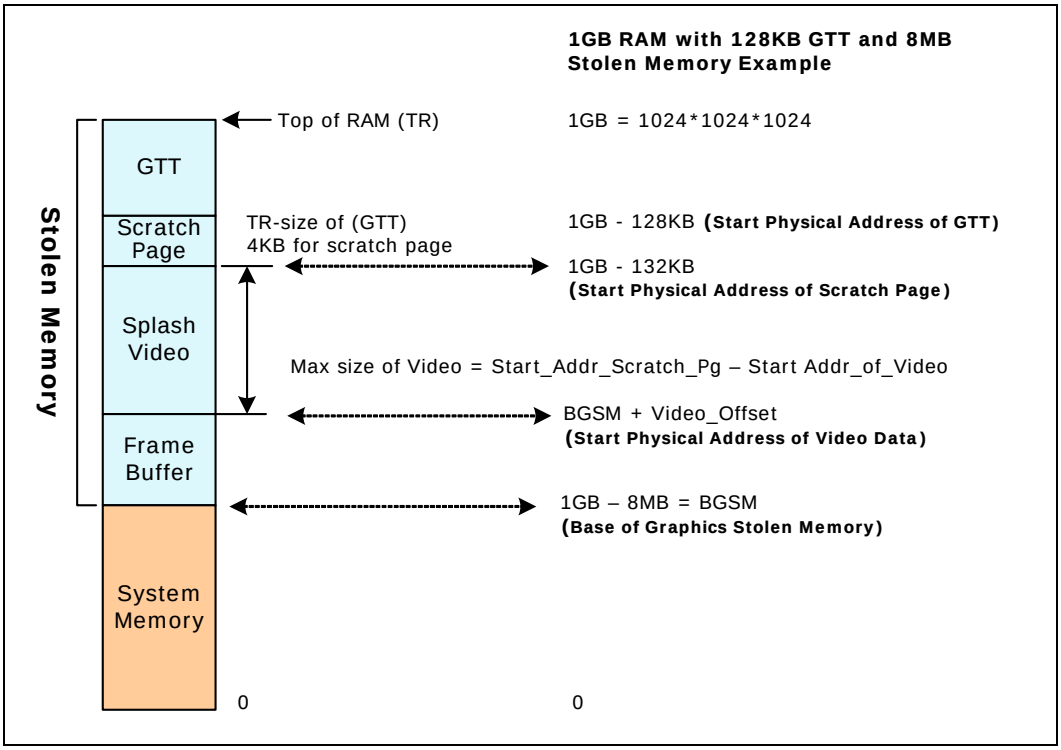
Table 14. Fastboot Options (Sheet 2 of 2)

Fastboot Settings	Description
Splash Video Offset (EFI and EPOG feature only)	The offset, in bytes, from the base of video memory where the Splash Video will be placed. Care must be taken to ensure that this location is past the end of the on-screen framebuffer and that the full Splash Video image fits within the pre-allocated video memory.
Splash Video Pixel Format (EFI and EPOG feature only)	The pixel format of the Splash Video image in memory. The available pixel formats are encoded values used within the Intel® EMGD.
Splash Video Source Width (EFI and EPOG feature only)	The width, in pixels, of the Splash Video image in memory.
Splash Video Source Height (EFI and EPOG feature only)	The height, in pixels, of the Splash Video image in memory.
Splash Video Source Pitch (EFI and EPOG feature only)	The pitch, in bytes, of the Splash Video image in memory. Pitch must be >= bytes per pixel * source width.
Splash Video Destination X (EFI only)	The X location, in pixels, where the Splash Video will be placed. This number is a signed number in 2's complement. Positive numbers are offset from the left of the screen. Negative numbers are offset from the right of the screen.
Splash Video Destination Y (EFI only)	The Y location, in pixels, where the Splash Video will be placed. This number is a signed number in 2's complement. Positive numbers are offset from the top of the screen. Negative numbers are offset from the bottom of the screen.
Splash Video Destination Height (EFI only)	The height, in pixels, of the Splash Video window on the screen. This number must currently be the same as SrcHeight.
Splash Video Destination Width (EFI only)	The width of the screen. This number must currently be the same as SrcWidth.

3.5.3.1 Configuring Splash Video

The splash video feature can be used to display a video while the system is booting to the operating system. This section describes how to configure the options needed.

Figure 15. Splash Video with 8 MB of Stolen Memory Example



The Video DMA area is where the video will be streamed. It is part of the stolen memory of our graphics device.

The external PCIe device that is connected to the camera needs to know the exact DDR RAM physical address to stream, or dump the video data at that memory location.

To calculate the Start DDR RAM physical address:

Start_Phy_Ram_Addr = BGSM + Video_Offset

where **BGSM** = Base of Graphics Stolen Memory

and **Video_Offset** = Offset where the video data is present. This is what you enter into the CED tool.

There are two ways to calculate BGSM:

- The recommended method is to use the `setpci` command in Linux to find the BGSM from the PCIe Config space.

  At the Linux command prompt, type the following:

  ```
  $ setpci -s 0:2.0 0x5C.L
  ```

OR

- Find the amount of physical RAM populated in the system, for example, 1 GB, and the stolen memory selected by the user in the system BIOS, for example, 8 MB.

  BGSM = 1 GB - 8 MB = 0x4000 0000 - 0x80 0000 = 0x3F80 0000

3.5.3.2 How to Select the Video_Offset

Determine the size of the maximum resolution of the framebuffer.

Size = framebuffer_height * framebuffer_pitch

where **framebuffer_pitch** = framebuffer_width * Bytes_per_Pixel (page aligned)

For example, 1024x768 at 32-bit BPP:

Size = 768 * (1024 * 4) = 3145728 = 0x30 0000

For some usage models, the framebuffer pitch is set to 8192 bytes. In that case:

Size = 768 * (8192) = 6291456 = 0x60 0000

The Video_Offset can start from **0x30 0000** or **0x60 0000** (if the pitch is 8192). See the notes below on the recommended values for the Video Offset.

Max Size of Splash Video = Size of Stolen Memory - Max Frame buffer size – Size of GTT – Size of Scratch Page (4 KB)

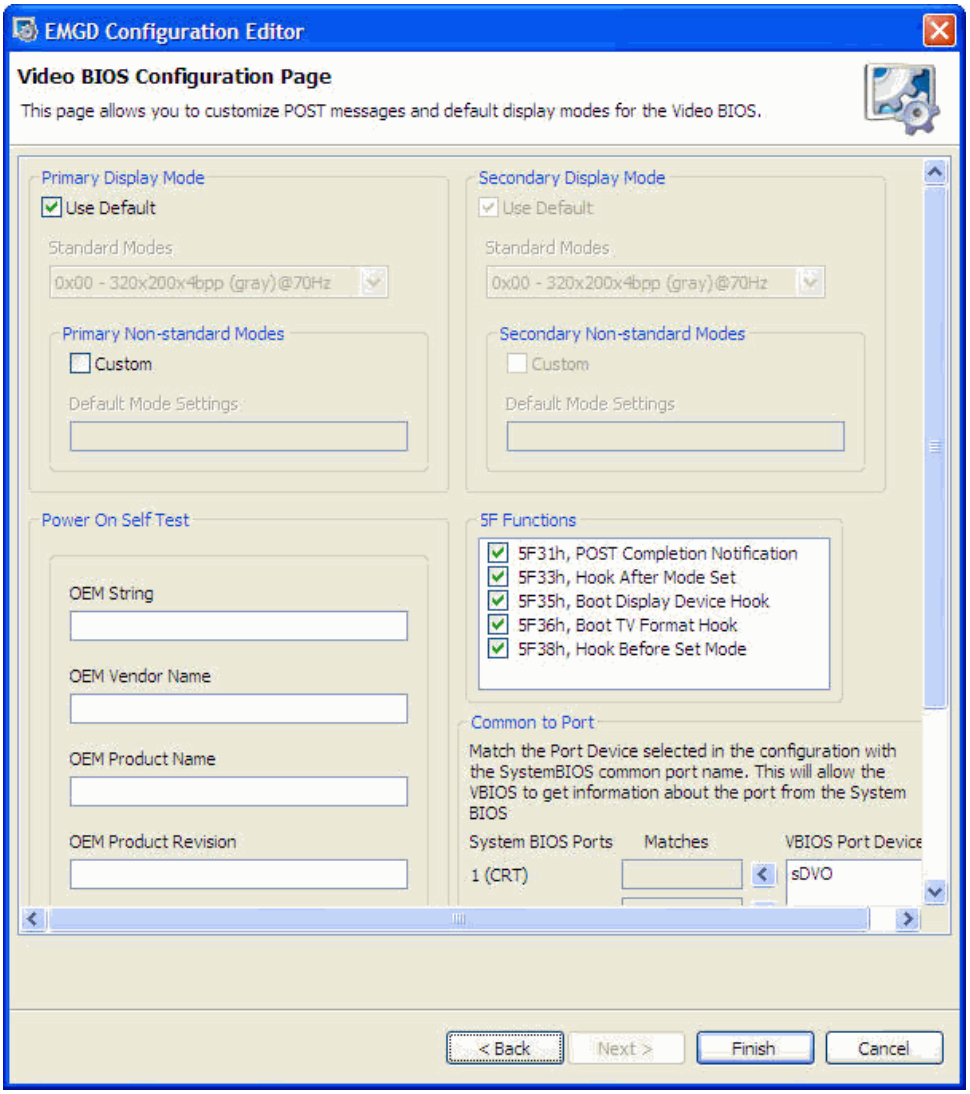
Notes:

1. For the Splash Video option the stolen memory MUST be a minimum of **8 MB**. This is selected in the BIOS menu.
2. The recommended Video Offsets for the splash video are **0x600000** and **0x700000**.
3. If the Size of the Video frame is more than **1 MB**, please choose **0x600000**.

3.5.4 Configuring the Video BIOS and EFI

The final page of the Intel® EMGD Configuration allows you to configure your video BIOS (if you are creating a configuration that includes the Video BIOS) and EFI. You can configure the Video BIOS by clicking **Next** after you configure each port. When you do, the following Video BIOS and EFI Configuration Page appears.

Figure 16. Video BIOS Configuration Page



From this page, you can customize POST (Power On Self Test) messages and default display modes as well as matching port devices to System BIOS ports.

The table below describes each field on this page.

Table 15. Video BIOS Settings Options (Sheet 1 of 2)

Video BIOS Settings	Description
Primary Display Mode	This section allows you to specify a standard or a customized display mode for the primary display. You can select a standard mode from any of the standard modes listed in the drop-down list. If you want to use a customized mode for the primary display, check the **Custom** check box and enter the mode number in the box. For a complete list of customized VGA and VESA modes, refer to Table 24, "Supported VGA Video Display Modes" on page 83 and Table 25, "VESA Modes Supported by Video BIOS" on page 85.
Secondary Display Mode	This section allows you to specify a standard or a customized display mode for the secondary display. You can select a standard mode from any of the standard modes listed in the drop-down list. If you want to use a customized mode for the secondary display, check the **Custom** check box and enter the mode number in the box. For a complete list of customized VGA and VESA modes, refer to Table 24, "Supported VGA Video Display Modes" on page 83 and Table 25, "VESA Modes Supported by Video BIOS" on page 85.
5F Functions	These settings allow you to enable or disable the five System BIOS 15h interrupt hooks. (Please see Appendix C, "Intel® 5F Extended Interface Functions" for more information on 5F functions.) All five functions are enabled by default.
Common to Port	The Common to Port section lets you match port devices with common System BIOS ports. This allows the Video BIOS to retrieve information about the port from the System BIOS. It allows you to associate standard display names used in most system BIOSs to specific ports that are recognized by Intel® EMGD (for example, LVDS, sDVO). The VBIOS makes this association when the VBIOS calls the System BIOS Intel® 5F interrupt functions. This setting consists of six numbers, where each number is associated with one of the System BIOS displays: 1 : CRT - Standard analog CRT 2 : TV1 - TV Output 1 3 : EFP1 - DVI Flat Panel 1 4 : LFP - Local Flat Panel (Internal LVDS display) 5 : TV2 - TV Output 2 6 : EFP2 - DVI Flat Panel 2 The values above are an example of the typical displays and corresponding order used by a system BIOS. However, this may vary depending on how your system BIOS has implemented the displays and the Intel 5F interrupt functions. The value in each position in the setting should be the associated port device. Using the typical settings above, if you want to associate CRT in the system BIOS with the internal CRT (port 1) and LFP in the system BIOS with internal LVDS (port 4) in the VBIOS, select **CRT** from the VBIOS Port Devices list and click the left arrow button next to the CRT row in the Matches column, and then select LFP from the vBIOS Port Devices list and click the left arrow button next to the LFP row in the Matches column. ***Notes:*** This feature must be compatible with the System BIOS. If the System BIOS does not properly implement the Intel 5F functions, then using the Common to Port feature could cause unpredictable results with the displays. If you are unsure, leave the Matches column blank for all ports to disable this feature. The Display Detect field on the Chipset Configuration page must be set to **Enable** in order for the Common to Port values to be used.
Enable POST messages to display	To enable Power On Self Test (POST) messages to display during the power on sequence, check this box. If left unchecked (i.e., cleared), the POST messages do not display.
OEM String	Enter a string of up to 100 characters. This string appears on the display when the Video BIOS starts up. The default is a blank string.

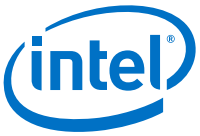


Table 15. Video BIOS Settings Options (Sheet 2 of 2)

Video BIOS Settings	Description
OEM Vendor Name	Enter a string of up to 80 characters that identifies the OEM Vendor. This string appears on the display when the Video BIOS starts up. The default is a blank string.
OEM Product Name	Enter a string of up to 80 characters that identifies the OEM Product Revision. This string appears on the display when the Video BIOS starts up. The default is a blank string.
OEM Product Revision	Enter a string of up to 80 characters that identifies the OEM Product Revision. This string appears on the display when the Video BIOS starts up. The default is a blank string.
Number of Seconds to Display	Enter the number of seconds to display the above information. The default is 1.

3.5.5 Creating an EPOG Configuration

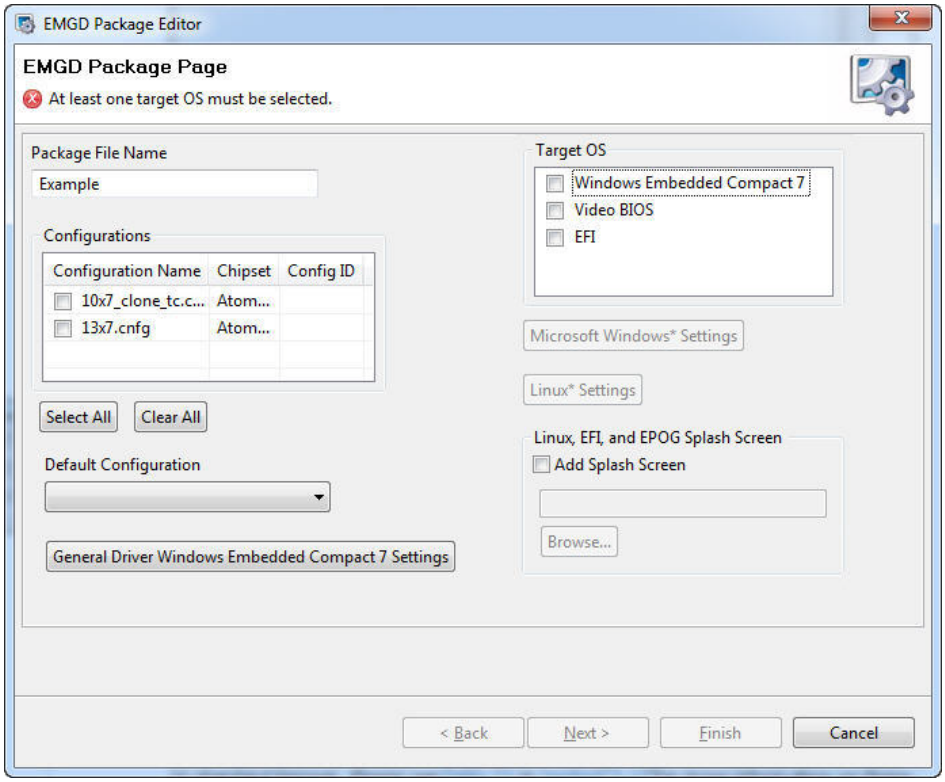
1. Create a new configuration, and select ONE display port only. Select the chipset and name the configuration.
2. At the port device configuration page, name the port and select the devices to be supported.
3. For LVDS panels, in the encoder configuration section, open the Attribute Settings page and set the values of Intensity and Inverter Frequency to **100** and **20300** respectively. These values may differ according to your hardware.
4. In the EDID Options section, clear all checked selections and then check the option **Use user-defined DTDs**.
5. Select ONE preconfigured DTD in the Custom Display Timing Descriptors section, or create a new DTD.
6. At the EFI GOP Configuration page, select any desired options. Only the Splash screen and Quickboot options are supported by EPOG.

3.6 Creating a New Package

A package consists of one or more configurations and is used to create an installation that works for multiple operating systems and chipset platforms and displays.

To create a new package, click the **New Package** link at the top of the main CED window. The Intel® EMGD Package Page appears.

Figure 17. Intel® EMGD Package Editor Page



The table below describes each field on this page.

Table 16. Intel® EMGD Package Editor Setting Options (Sheet 1 of 2)

Package Option	Description
Package File Name	Enter a name for the package. This is a required field and the name must be between 1 and 50 characters and may contain spaces.
Configurations	This block shows the configurations that are available to be packaged. Each package consists of one or more configurations, each of which is associated with a specific chipset. To select a configuration, click the check box next to the configuration name. You can select all available configurations by clicking **Select All** located below the Configurations block and clear all configurations by clicking **Clear All**. The Configuration Name column shows the name of each configuration and the Chipset column shows the chipset associated with each configuration. In the Config ID column, you must enter a configuration ID for each configuration. The configuration ID must be a number between 1 and 15. By default, the Package Editor automatically assigns the next available configuration ID when you select a configuration. You can change the default configuration ID by clicking in the edit box and entering a different value.

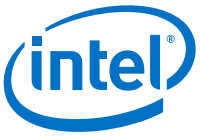


Table 16. Intel® EMGD Package Editor Setting Options (Sheet 2 of 2)

Package Option	Description
Default Configuration	The Default Configuration list box allows you to select a default configuration from the configurations you selected in the Configurations block. For single configurations the default is the one selected in the previous option. For multiple configurations, the default is the first one selected in the Configurations list. To have no default configuration, select **None**. See also Section 5.2.1, "Universal INF Configuration" on page 87.
Target OS	This block allows you to select one or more operating systems and Video BIOS for the package. For each target you select, CED produces a configuration file for the selected OS or Video BIOS platform. Please see the following section for settings on the Target OS: • "Entering Linux OS Options" on page 47 • "Entering Windows OS Options" on page 48 • "Generating a VBIOS Package" on page 50 • "Entering EFI Options" on page 50 • "Entering EPOG Options" on page 52
Microsoft Windows Settings	If you are creating a package for a Microsoft Windows* platform, click the **Microsoft Windows Settings** button for additional settings that may be required for your configuration. Please see "Entering Windows OS Options" on page 48 for descriptions of these settings.
Linux Settings	If you are creating a package for a Linux OS platform, click the **Linux Settings** button for additional settings that may be required for your configuration. Please see "Entering Linux OS Options" on page 47 for descriptions of these settings.
EFI and EPOG Splash Screen	The **Add Splash Screen** check box enables the use of a splash screen, which you define using the **Browse...** button to locate the file. The splash screen is limited to 500 KB in size and JPG and BMP formats. For Quickboot, only BMP format is allowed.

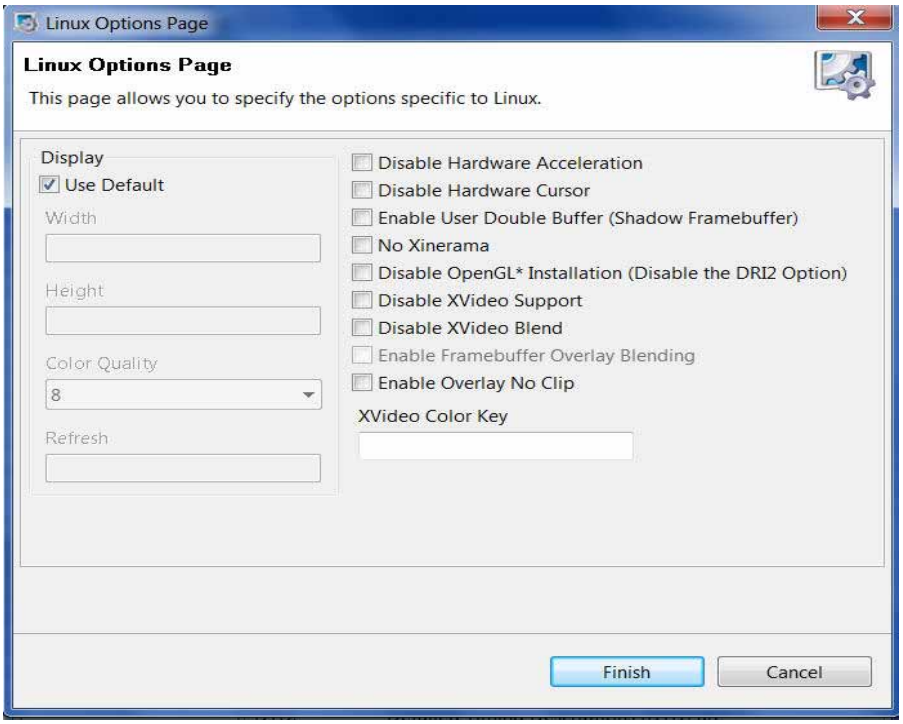
If you are not creating a VBIOS package, click **Finish**. When you click **Finish**, CED creates a package that can be used for generating an installation.

Note: CED supports generating a 127K vBIOS ROM image for the E6xx platform and a 64K vBIOS ROM image for the US15W platform.

3.6.1 Entering Linux OS Options

The Linux Options Page allows you to enter Linux OS-specific options into the configuration. When you click **Linux Settings** from the Intel® EMGD Package Page, the following page appears.

Figure 18. Linux Options Page



The table below describes each of these settings.

Table 17. Linux OS Settings Options (Sheet 1 of 2)

Linux OS Option	Description
Default Display Modes	The Default Display Modes section allows you select the default resolution, color depth, and refresh rate for the configuration. If you do not select a default display mode, the package uses the default display mode for the operating system it is installed on.
Disable Hardware Acceleration	Disable or enable hardware 2D acceleration. The default is to enable hardware acceleration, so to disable acceleration, click the check box.
Enable Hardware Cursor	Enable the use of the hardware cursor. By default, the hardware cursor is disabled.
Enable Use Double Buffer (Shadow Framebuffer)	Enable double buffering on the framebuffer. By default, double buffering is disabled. To enable it, click the check box.
No Xinerama	Xinerama support. Xinerama is an extension to the X Window System which allows applications and window managers to use the two (or more) physical displays as one large virtual display. By default, Xinerama is enabled. To disable it, click the check box.

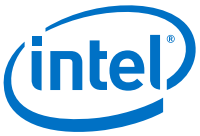


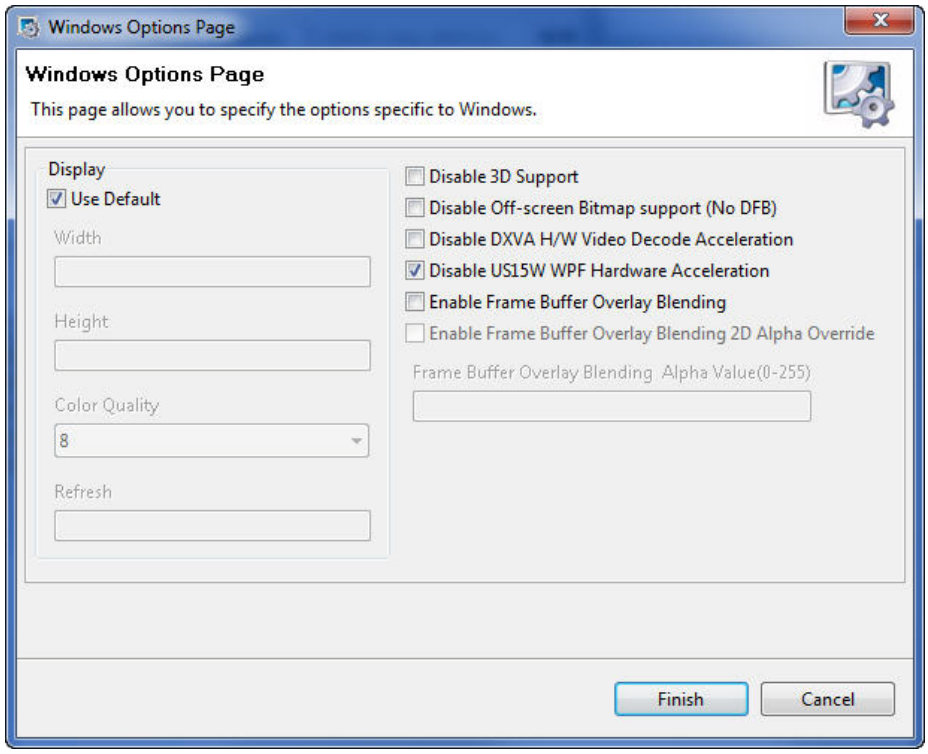
Table 17. Linux OS Settings Options (Sheet 2 of 2)

Linux OS Option	Description
Disable OpenGL* Installation (Disable the DRI2 Option)	OpenGL* (Disable the Direct Rendering Infrastructure (DRI) Option). DRI2 allows the client to directly write to DMA buffers that are used by the graphics hardware. To **disable** OpenGL, check the box. The option “DRI2” “0” will be set for every available display. This will turn off direct rendering and disable hardware accelerated OpenGL. By default, OpenGL is enabled. No “DRI2” line(s) are placed in the configuration file. The driver will intelligently determine if DRI2 can be supported and will enable it if possible. If you manually edit the configuration file and set option “DRI2” “1” on more than one display, deadlock will occur and OpenGL will fail. If you are unsure of which setting to use, just leave the box unchecked (i.e., cleared) and do not edit the DRI2 option in the configuration file and the driver will handle it automatically. This feature can be used if you want to test your applications with and without hardware accelerated OpenGL.
Disable XVideo Support	Disable XVideo support. In a dual independent head configuration, either the first display or the second display supports XVideo. Both displays can not support XVideo simultaneously. The default is XVideo support is enabled.
Disable XVideo Blend	Disable XVideo support using the 3D blend manager. This provides XVideo support in configurations that cannot be supported with overlay. For example, this is supported on both displays in a dual independent head setup. It is also supported when the display is rotated or flipped. Color key is only supported if ShadowFB is enabled and the VideoKey is defined. The default is XVideoBlend support is enabled.
Enable Frame Buffer Overlay Blending	When checked, this enables overlay blending with the framebuffer on both display outputs on US15W and when display mode resolution is 32-bit XRGB.
XVideo Color Key	This sets the color key for XVideo and XVideoBlend. This value is either a 24-bit value or a 16-bit value, depending on the pixel depth of the screen. The color key is always enabled for XVideo, even when it is not defined. The color key is always disabled for XVideoBlend unless both this option is defined and the ShadowFB option is enabled. The default color key for XVideo is 0x0000ff00. For XVideo Blend, the color key is disabled by default.

3.6.2 Entering Windows OS Options

The Windows Options Page allows you to enter Windows OS-specific options into the configuration. When you click **Microsoft Windows Settings** from the Intel® EMGD Package Page, the following page appears.

Figure 19. Windows Options Page



The table below describes each field on this page.

Table 18. Windows OS Setting Options (Sheet 1 of 2)

Windows OS Option	Description
Display	The Display section allows you to use the default settings by checking the **Use Default** check box or to select the default width, height, color quality, and refresh rate for the configuration.
Disable 3D Support	Specifies whether to enable D3D. The default is to enable 3D support (not checked).
Disable Off-screen Bitmap support (No DFB)	This option turns OFF the driver capabilities to create and use offscreen bitmaps that are used to improve GDI and DirectDraw* performance in the driver. When this option is ON, you may see some GDI and DirectDraw performance degradation. The drv functions below will be affected when this option is turned on. • DrvCreateDeviceBitmap • DrvDeleteDeviceBitmap • DrvDeriveSurface
Disable DXVA H/W Video Decode Acceleration	This option is enabled by default in Intel® EMGD, however, by selecting this option, you can disable DXVA hardware video decode acceleration.

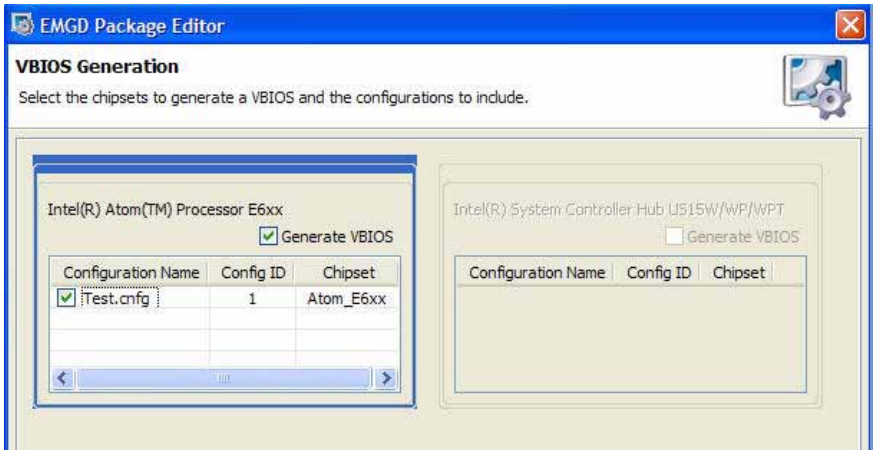
Table 18. Windows OS Setting Options (Sheet 2 of 2)

Windows OS Option	Description
Enable Frame Buffer Overlay Blending	When checked, this option enables overlay blending with the framebuffer on both display outputs (if in VEXT mode) on US15W and when display mode resolution is 32-bit XRGB. This option is currently not applicable for Atom E6xx.
Enable Frame Buffer Overlay Blending 2D Alpha Override	This option applies only to Windows XP and US15W. When checked, it enables an override to the frame buffer overlay blending 2D alpha. ***Note:*** Checking the Frame Buffer Overlay Blending option and running a 3D alpha blending application on overlay [non full screen mode] causes the black icons on the desktop to appear. This is expected behavior as the operating system sets the 2D alpha values. To overcome this behavior, choose **Enable Frame Buffer Overlay Blending 2D Alpha Override** option and then enter the alpha value. This alpha override will cause performance impact when a lot of 2D blitting operations take place.
Frame Buffer Overlay Blending Alpha Value	The valid range is from 0x00 to 0xFF.

3.6.3 Generating a VBIOS Package

If you are creating a package for a VBIOS installation, click **Next**. CED displays the VBIOS Generation page.

Figure 20. VBIOS Generation Page

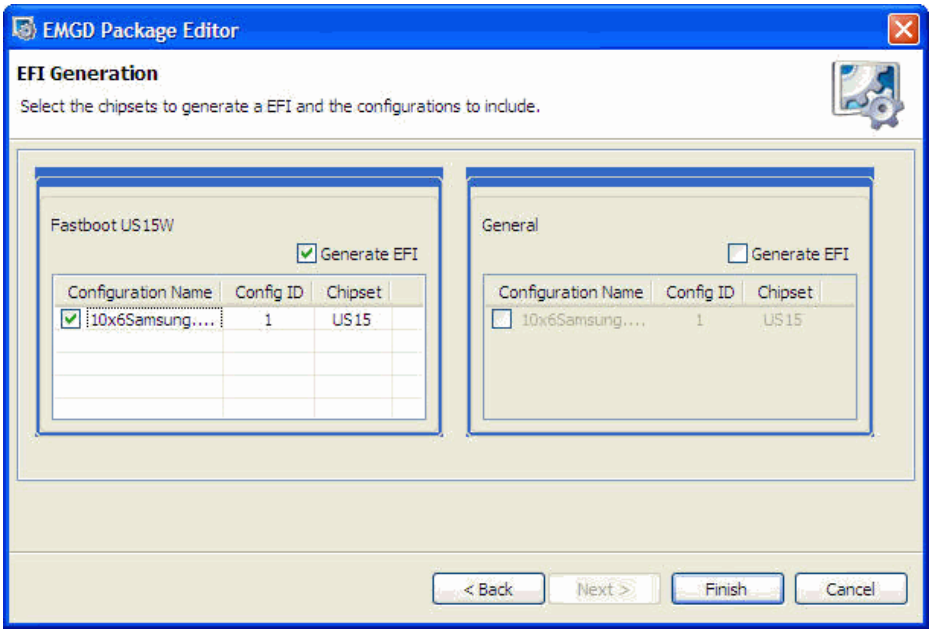


To generate a VBIOS, click the **Generate VBIOS** check box and select the configurations to include. After selecting the chipset and the configurations, click **Finish**. CED generates a package that includes both the OROM and the TSR for the chipsets and the configurations you selected.

3.6.4 Entering EFI Options

If you are creating a package for a EFI installation, click **Next**. CED displays the EFI Generation page.

Figure 21. EFI Generation Page



To generate an EFI configuration:

1. In the Fastboot and/or General modes sections, click the **Generate EFI** checkbox.
2. Select the chipset and configuration(s) to include.
3. Click **Finish**.
   CED generates a package that includes the EFI driver for the modes, chipsets and the configurations you selected.

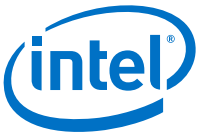
3.6.4.1 Using the Generated EFI Configuration

Use Intel® EMGD CED to configure and build an EFI video driver for your platform, as described in Section 3.6.4 and then follow the instructions below to install the driver.

1. After building the EFI driver, copy the appropriate module to your working directory where you keep your Aptio MMTOOL and EFI BIOS that needs to be updated.
   The file is typically called `IEGD.DXE` and is found in the IEGD ZIP file in the installations folder under EFI.
2. Make a working copy of your EFI BIOS image.
   For example, copy `CBCHAxxx.ROM` to `CBCHAxxx_IEGD_EFI.ROM`
   where *xxx* = the release version of Standard BIOS

   OR

   Copy `CBFBAxxx.ROM` to `CBFBAxxx_IEGD_EFI.ROM`
   where *xxx* = the release version of Fast Boot BIOS)
3. Start the MMTOOL in GUI mode.
4. Load the EFI BIOS image using the **Load Image** button.
   After it loads you will be presented with a list of existing modules.
5. Select `CBCHAxxx_IEGD_EFI.ROM` or `CBFBAxxx_IEGD_EFI.ROM` (from step 2.)



6. If it exists, delete any legacy VBIOS by highlighting the old video solution, select the DELETE tab at the top, and then press the **DELETE** button.

Note: The EFI Fast Boot images typically do NOT contain a video module.

For example, for `CBCHAxxx.ROM` you will see a CSMVIDEO module. This is the Compatibility Software Module for a legacy VBIOS.

7. If it exists, delete any old versions of the Intel® EMGD EFI Fast Boot Video Driver. Look for an unnamed module with a GUID that starts with "2B13E5F0-" or with a module name that includes "IEGD". If it exists, select the DELETE tab, highlight the module and then click the **DELETE** button.
8. Insert the new video module by clicking on the INSERT tab, specifying the module file name, and then clicking the **INSERT** button. You may browse to locate the file, for example, `iegd.dxe`.)
9. Save image by clicking the **Save Image** button and then close the dialog box.
10. Flash the image into your flash chip and install it on the board. You can either use the hardware flash programmer or the Aptio AFUDOS tool for this purpose.

3.6.5 Entering EPOG Options

If you are creating a package for an EPOG installation, follow the steps below.

1. From the Target OS section, select **EPOG**.
2. Select the configuration desired configuration. For help on creating an EPOG configuration, see "Creating an EPOG Configuration" on page 44.
3. If you want to use a splash screen and have set up your configuration with the correct options (see Section 3.5.5), select the Add Splash Screen check box and then browse to the `.bmp` file you want to use.
4. Click **Next**. Select the relevant checkboxes and then click **Finish**.

3.6.5.1 Using the Generated Embedded Pre-OS Graphics Feature Configuration

Use Intel® EMGD CED to configure and build a driver with the embedded pre-OS graphics feature, as described in "Entering EPOG Options" and then follow the instructions below to install the driver.

1. Generate an installation from your EPOG package.
2. Install the Boot Loader Development Kit (BLDK) on a Linux OS running on the target platform. It can be obtained from http://edc.intel.com/, under the 'Boot loader technology' option in the Software tab.
3. CED has generated the following files: `libepog.a` and `epog.h`. Copy these files to the `bldk` folders on the target platform.
   a. Copy `libepog.a` to `bldk/core/lib/elf`.
   b. Copy `epog.h` to `bldk/core/include`.
4. On the target platform, go to the folder `bldk/core/target/brd_crown_bay`.
   a. Edit the file `Makefile`.
   b. Change the value of the option `CFG_GFX` to `1`.
   c. Change the value of the option `CFG_VBIOS` to `0`.
5. Run the commands:
   make clean
   make

6. Copy the generated `rom.bin` file from the `bldk/core/target/brd_crown_bay/elf` directory.
7. Flash the rom.bin file to your BIOS chip. If the BIOS chip is 2 MB, use the start address of 0x100,000.

3.7 Generating an Installation

After you have created a package, you can generate an installation for the package by following this procedure.

1. Select a package from the Package folder located on the left pane of the CED main window.
2. Click **Generate Installation**. While the installation is building, CED displays a progress bar. When the installation is complete, CED places the output in the Installation folder on the left pane of the CED window.

For each OS and VBIOS platform specified in the package, CED generates a folder in the `...\workspace\installation` folder under the current folder. For example, if you select a package that contains configurations for all supported operating systems and the VBIOS, CED generates the following folders:

```
...\workspace\installation\<package name_installation>\IEMGD_HEAD_Linux
...\workspace\installation\<package name_installation>\IEMGD_HEAD_WINDOWS
...\workspace\installation\<package name_installation>\IEMGD_HEAD_WINCE70
...\workspace\installation\<package name_installation>\IEMGD_HEAD_VBIOS
...\workspace\installation\<package name_installation>\IEMGD_HEAD_EFI
```

These folders contain all the subfolders required for the installation onto the target systems. To complete the installations on the target systems, refer to the following sections:

- “Installing and Configuring Linux* OS Drivers” on page 133
- “Configuring and Installing Microsoft Windows Drivers” on page 87
- “Configuring and Building Intel® EMGD for Microsoft Windows* Embedded Compact 7” on page 101

3.8 Configuring the System BIOS for Use with the Intel® EMGD

Some aspects of configuring the Intel® Embedded Media and Graphics Driver are common across the Video BIOS (VBIOS), EFI, and the drivers for the supported operating systems. The following sections provide an overview for configuring both the VBIOS and Intel® EMGD and describe in detail the common components and tools. This section also describes how to configure the system BIOS for the supported systems.

3.9 System BIOS Settings

Before installing Intel® EMGD, you must first configure the system BIOS. The following sections describe the required settings. These descriptions are based on AMIBIOS8* from American Megatrends, Inc., which is the recommended system BIOS to use with Intel® EMGD. Settings may vary if a different system BIOS is used.

3.9.1 GMCH PCIe Device Enabling

The PCIe Device Enabling feature on the Graphics and Memory Controller Hub (GMCH) should be set as specified in the table below.

Table 19. GMCH Device 2, Function 1 BIOS Setting

OS	Chipset
	Intel® Atom™ Processor E6xx, Intel® US15W/US15WP/WPT
Microsoft Windows* XP and Microsoft Windows XPe*	Disabled
Linux*	Disabled

3.9.2 Graphics Mode Select (GMS)

The System BIOS typically allows a portion of physical memory to be dedicated to firmware and graphics driver use. This dedicated memory is known as stolen memory since it is not available to the operating system. The size of this memory is selectable and chipset-specific. Stolen memory is typically used by the firmware and graphics driver to locate the framebuffer, but can also be used as scratch and surface memory. Because it is programmatically set aside during boot by the System BIOS, access to it is direct and does not require OS memory allocation services. Firmware is fully responsible for stolen memory management.

Graphics Mode Select (GMS), or stolen memory, can be set to any of the sizes listed in the table below. Smaller sizes limit the framebuffer size during firmware boot. Larger sizes marginally increase surface allocation performance for the graphics driver.

Table 20. GMS Settings

Chipset	GMS Settings
Intel®US15W/US15WP/WPT	64 MB, 128 MB, 256 MB
Intel® Atom™ Processor E6xx	64 MB, 128 MB, 256 MB

3.9.3 AGP (Accelerated Graphics Port) Aperture Size

The AGP Aperture size controls the total amount of graphics memory that can be mapped in the AGP Aperture. This value can be set from 64 MB up to 256 MB, depending on the chipset. Refer to specific chipset details for information on the valid range.

3.10 VBIOS and Driver Configuration

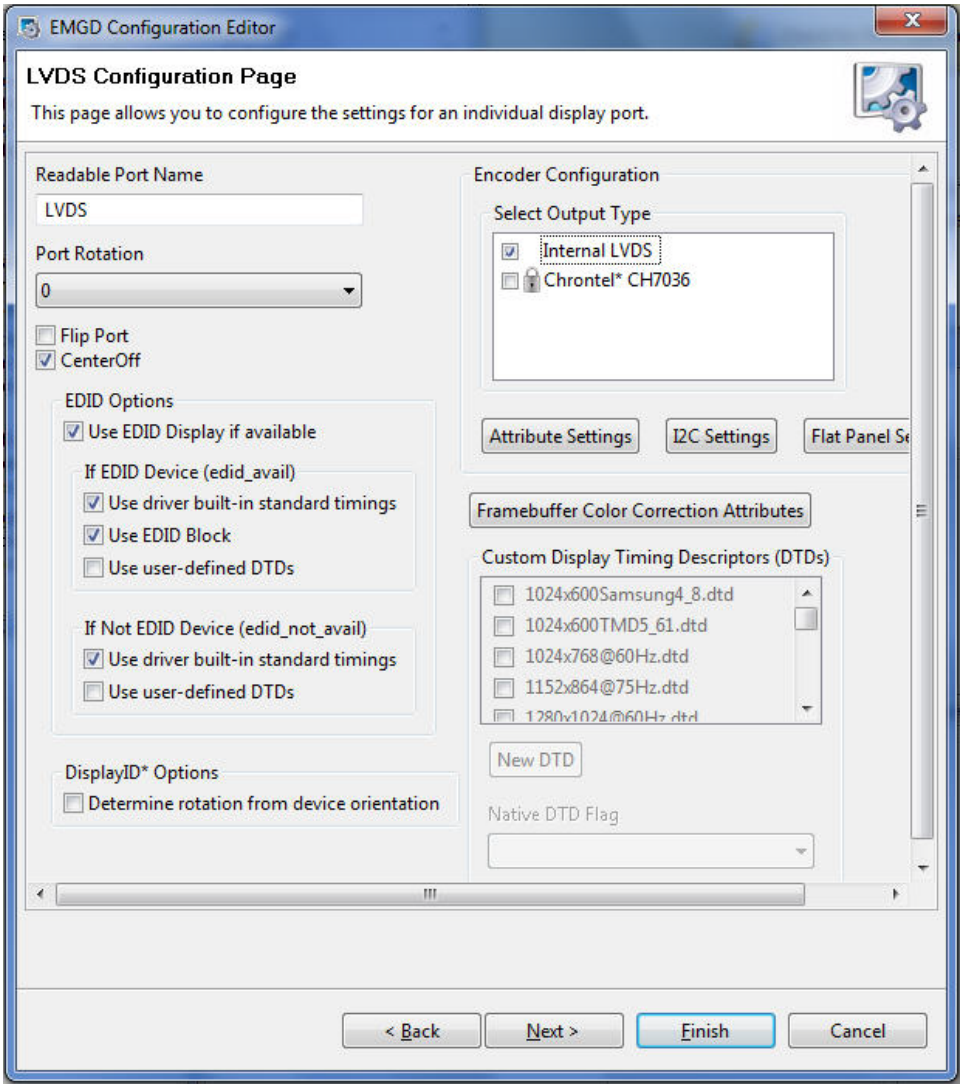
The Intel Embedded Graphics Suite allows user configuration of both the VBIOS and graphics driver as well as programming of Detailed Timing Descriptors (DTDs) for EDID-less panels for both the VBIOS and graphics driver. This is accomplished using CED, which offers several ways to input DTDs, each associated with a potential target panel and display mode for the system. CED generates DTD and configuration settings used by the Intel® EMGD VBIOS, Linux, and/or Windows drivers.

The following example is for a system setup with just an internal LVDS and sample timing parameters for illustration purposes only. You can use this example to set up DTD timings that are specific to your non-standard panels and then activate the panels using a custom mode.

To create a configuration and configure the LVDS options:

1. Create a custom DTD as described in Section 3.4, “Creating a New Customized DTD” on page 23.
2. From the CED main screen, select **New Configuration**.
3. Enter a name for the configuration in the text box provided, for example, *LVDS_test*.
4. Select the platform chipset.
5. In the list of available ports, select **LVDS** and then click **Next**.
6. On the LVDS Configuration Page, clear the checkbox for **Use EDID Display if available**, which disables all the selections under **If EDID Device (edid_avail)**. The screen will be similar to the example below.
7. Select the checkbox for **Use user-defined DTDs**.
8. In the Encoder Configuration section, select **Internal LVDS**.
9. In the Custom Display Timing Descriptors (DTDs) list, select the DTD you created in Section 3.4, “Creating a New Customized DTD” on page 23 for example, *test_LVDS*.

Figure 22. LVDS Configuration Page

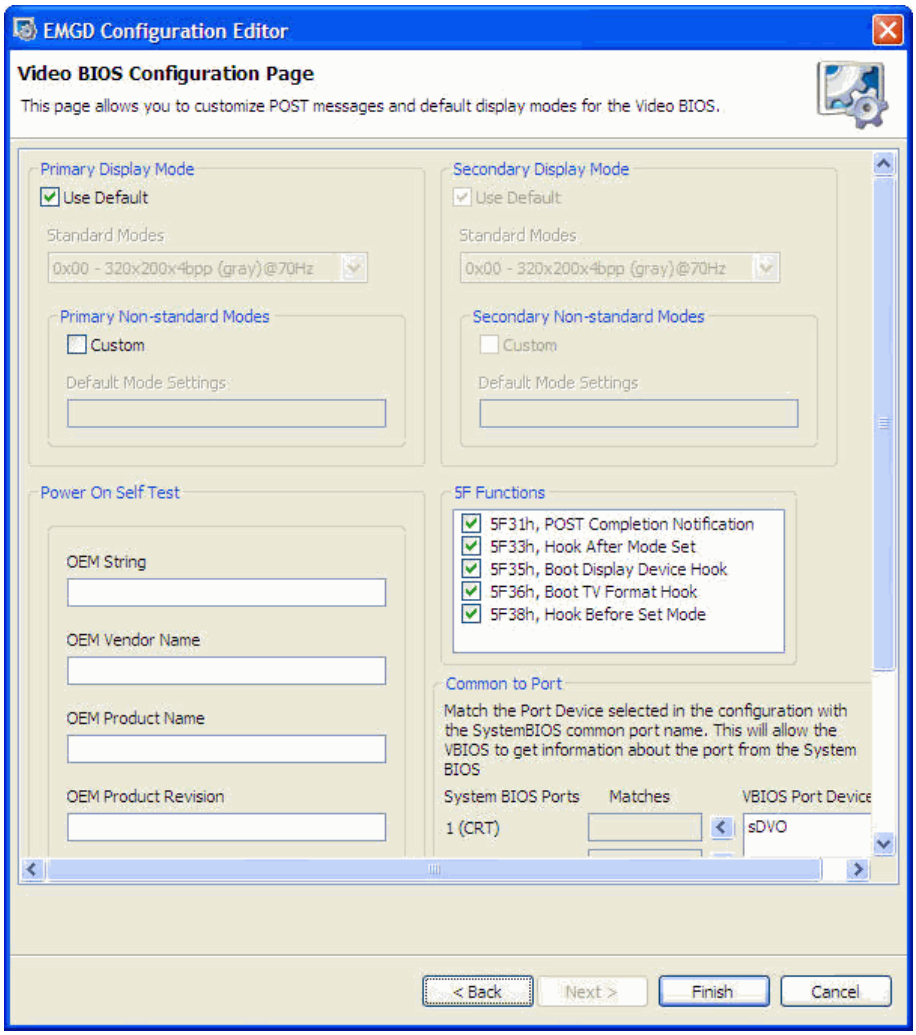


10. Click **Next**.
11. (Optional) Configure Fastboot options as described in "Configuring Fastboot" on page 37.
12. Click **Next**.

To set the custom mode:

1. From the Intel® EMGD CED screen (similar to the example below), in the Primary Display Mode section, clear the **Use Default** checkbox.
2. In the Primary Non-standard Modes section, select the checkbox for **Custom**.
3. In the Primary Non-standard Modes section, enter 0x120 in the Default Mode Settings text box. (See a description of the custom modes.)

Figure 23. Intel® EMGD Configuration Editor Page



Custom Modes

The custom modes begin with 0x120 (0x121 and 0x122 are the same modes in different pixel formats). If there was a second custom mode entered it would begin with 0x123 to 0x125.

From the above DTD 200x200 example, this is what the custom modes represent:

0x120 200x200@8bpp

0x121 200x200@16bpp

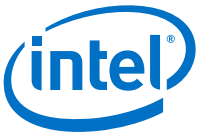
0x122 200x200@32bpp

And if the second custom mode was a 400x400 panel, its custom modes would be:

0x123 400x400@8bpp

0x124 400x400@16bpp

0x125 400x400@32bpp



3.11 Configuration Options

The table below describes available Intel® EMGD settings. The gray rows are block headings and the non-gray rows that follow each heading are settings within the block. Some of these block headings are contained within prior block headings.

Table 21. Parameter Configuration Format (Sheet 1 of 7)

Name	Range / Value	Description
ConfigID	Integer (1-15)	Optional keyword used to specify which configuration is used. The config ID specified here must match one of the configuration IDs defined with CED. If this keyword is omitted, all configurations specified in the config file are used. Note that this keyword is not required for Linux OS and VBIOS configurations.
Config	Integer (1-15)	More than one configuration is valid.
Comment		A quoted string used to identify the origin of the .bin or .inf file.
Name		A quoted string used to identify the configuration name. Name is a required field for VBIOS configuration.
General		Settings that are generic to the configuration.
DisplayConfig	1 – Single 2 – Clone 8 – Extended Default: 8	Used to configure initial state of attached displays. 1 – Single. A single display. 2 – Clone. Primary and secondary displays enabled and configured with separate timing pipes. This allows different timings to be applied to each display. Resolutions can be different on both displays. 8 – Extended. Configures separate pipes to allow primary and secondary displays to have different resolutions and display different content. Upon first boot after the driver installation, this option will enable only the primary display, as the extended modes must be enabled in the operating system (i.e., **Extended Desktop** in the **Display Properties** sheet within Microsoft Windows).
DisplayDetect	0 - Disable 1 - Enable	Enable or disable Display Detection. Note that this parameter must be Enabled in order to use COMMON_TO_PORT values. Default is 0. Please see Section 3.12, "Display Detection and Initialization" on page 65 for detailed information on this parameter.
PortOrder	PortOrder must be specified as a quoted string containing five digits. The valid values are: 2 - sDVO B port 4 - Integrated LVDS port (mobile chipsets only) Default: 0 for all keys	Search order for detecting attached displays for the Display Detection feature. When Display Detection is enabled, the PortOrder determines which display is primary and which display is secondary. The port search order can be specified to ensure the port device (sDVO device) is found, based on the system integrator's routing choices. Default ordering is chosen by specifying zeros in the PortOrder keys. Default ordering is chipset specific; see Table 41, "Default Search Order" on page 188. Please see Section 3.12, "Display Detection and Initialization" on page 65 for more information on using `PortOrder` in combination with the Display Detect feature.

Table 21. Parameter Configuration Format (Sheet 2 of 7)

Name	Range / Value	Description
CloneWidth CloneHeight	Typical sizes: clonewidth – 800, cloneheight - 600 clonewidth – 1024, cloneheight - 768 clonewidth – 1280, cloneheight - 768 clonewidth – 1400, cloneheight – 1050	Width and height for a cloned display.
CloneRefresh = 60	Typical refresh rates (expressed in Hz): 60 Hz, 75 Hz, 85 Hz	Refresh rate for a cloned display.
OverlayOff	0 - Overlay on (default) 1 - Overlay off	This parameter allows you to disable Overlay support, which is enabled by default. **Note:** This parameter is only for Microsoft Windows*. The Linux* OS configuration for the xorg.conf provides a standard option that performs the same function.
FbBlendOvl	0 - Off (Default) 1 - On	When checked, this enables overlay blending with the framebuffer on both display outputs when display mode resolution is 32-bit XRGB.
No_DFB	0 - Off (Default) 1 - On	This parameter enables the Intel® EMGD to pass the DIB call back to the OS. This is required in certain circumstances to improve performance.
FbBlend2DOverRide	0 - Off (Default) 1 - On	This option applies only to Windows XP and US15W. When checked, it enables an override to the frame buffer overlay blending 2D alpha. ***Note:*** Checking the Frame Buffer Overlay Blending option and running a 3D alpha blending application on overlay [non full screen mode] causes the black icons on the desktop to appear. This is expected behavior as the operating system sets the 2D alpha values. To overcome this behavior, choose **Enable Frame Buffer Overlay Blending 2D Alpha Override** option and then enter the alpha value. This alpha override will cause performance impact when a lot of 2D blitting operations take place.
FbBlendAlphaValue	The valid range is from 0x00 to 0xFF.	The Alpha value used for the frame buffer blend.
OverlayNoClip	0 - Off (Default) 1 - On	Enables/disables the colorkey feature.
vbios		This block contains settings for the Video BIOS. Note that you only need to specify the parameters you are actually using. You do not need to specify all the parameters in this block. If you omit any parameters, the vbios uses the default values.
COMMON_TO_PORT	6 digit value	Maps the ports from the system BIOS to a port number used by the graphics hardware. Please see Section 3.5.4, "Configuring the Video BIOS and EFI" on page 41 for more information on this parameter. Note that the `displaydetect` parameter must be set to Enabled in order for the COMMON_TO_PORT values to be used. The default is all zeroes: 000000

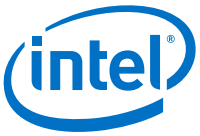


Table 21. Parameter Configuration Format (Sheet 3 of 7)

Name	Range / Value	Description
post_display_msg	0 - disable greater than 0 - enable and display POST message for the specified number of seconds	Enables or disables the POST (Power On Self Test) message. When you specify a value greater than 0, the message is displayed for the specified number of seconds. For example: post_display_msg = 5 This enables the POST message and displays it for approximately 5 seconds. The maximum value that can be entered here is 65535. The default is 1, enable and display the POST message for approximately 1 second.
oem_string	double-quoted string	This string appears on the display when the post_display_msg is enabled and the VBIOS starts up. The maximum string length is 100 characters. The default is " " (two double quotes with a single space in between).
oem_vendor	double-quoted string	This string appears on the display when the post_display_msg is enabled and the VBIOS starts up. The maximum string length is 80 characters. The default is " " (two double quotes with a single space in between).
oem_product_name	double-quoted string	This string appears on the display when the post_display_msg is enabled and the VBIOS starts up. The maximum string length is 80 characters. The default is " " (two double quotes with a single space in between).
oem_product_rev	double-quoted string	This string appears on the display when the post_display_msg is enabled and the VBIOS starts up. The maximum string length is 80 characters. The default is " " (two double quotes with a single space in between).

Table 21. Parameter Configuration Format (Sheet 4 of 7)

Name	Range / Value	Description
int15	5 digits	This parameter allows you to enable or disable the five System BIOS 15h interrupt hooks. The value must be 5 digits in length. Each digit is associated with one of the five System BIOS interrupt 15h hooks as shown below (left to right) 1 - 5F31h, POST Completion Notification Hook 2 - 5F33h, Hook After Mode Set 3 - 5F35h, Boot Display Device Hook 4 - 5F36h, Boot TV Format Hook 5 - 5F38h, Hook Before Set Mode (Please see Appendix C, "Intel® 5F Extended Interface Functions" for more information on 5F functions.) The value of each digit must be a 0 or a 1 as follows: 0 - disable a System BIOS 15h hook 1 - enable a System BIOS 15h hook For example, `int15 = 11001` Enables 5F31h, 5F33h, and 5F38h hooks only. The 5F35h and 5F36h hooks are disabled. The default is 11111, enable all five hooks.
port	2 -sDVO B port 4 - Integrated LVDS port (mobile chipsets only)	Used to define port specific settings.
rotation	Degrees / Windows/ Linux / WEC7 0 / 0 / 0x00 90 / 90 / 0x5A 180 / 180 / 0xB4 270 / 270 / 0x10E Default: 0 degrees	Rotation of the display.
flip	Flip / Windows/ Linux / WEC7 off / 0 / 0x00 on / 1 / 0x01 Default: off	Flip of the display.
centeroff	Default: 0 – disabled, allow centering and add compatibility modes 1 – enabled, no centering, no added compatibility modes	When this option is enabled it DISABLES centering. Also, depending on the combination of "edid" + "user-dtd" + connected hardware, Intel® EMGD will add missing compatibility modes (6x4, 8x6, 10x7& 12x10) via centering. Use this option to disable this feature.
edid	0 – Do not read EDID from panel/CRT 1 – Attempt to extract EDID timing data from panel/CRT	If VBIOS/driver reads EDID from panel/CRT.

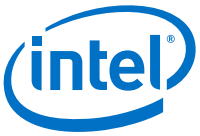


Table 21. Parameter Configuration Format (Sheet 5 of 7)

Name	Range/Value	Description
edid_avail edid_not_avail	Range [16 bits] Valid values (specified in hex): bit 0=0: Do not use built-in standard timings. =1: Use driver built-in standard timings. bit 1=0: Do not use EDID block. =1: Use EDID block and filter modes. (Bit 1 not applicable to edid_not_avail.) bit 2=0: Do not use user-defined DTDs. =1: Use user-defined DTDs. bits 3-15: Reserved for future use.	These two parameters are used to control the available timings for any display. `edid_avail` is used when EDID values are read from the display. If an attempt to read EDID from the display fails or the `edid` parameter is set to 0, then the driver uses the `edid_not_avail` flags. The value for both parameters must be specified as a decimal or hex value, e.g., "3" or "0x3" Defaults: edid_avail: "3" sets Bit 0 = 1, Bit 1 = 1, Bit 2 =0 (Use driver built-in standard timings and EDID block and filter modes.) edid_not_avail: "1" sets Bit 0 = 1, Bit 1 = 0, Bit 2 = 0. (Use driver-built-in standard timings.) Please see Section 3.13, "Advanced EDID Configuration" on page 67 for detailed information.
DIDRotation	Range: 0 - Disable. Do not use rotation/flip setting based on DisplayID file. 1 - Enable. Use rotation/flip setting based on the DisplayID file for display port configuration.	This option provides flexibility to enable/disable display rotation configuration based on DisplayID file. Default: disabled To enable display rotation configuration based on DisplayID file, please ensure you configure following parameters: - DIDRotation =1 - flip =0 - rotation =0 - edid =1 - edid_avail = 2 or 3 or 6 or 7 ***Note:*** If you set the Port Rotation and Flip Port option in CED to a non-zero value, then the graphics will use your setting instead of the DisplayID file configuration.
dvo		sDVO device information.
i2cpin	<0-6>	The GPIO pin pair used on the I^2C bus to read and write to sDVO device registers.
ddcpin	<0-6>	The GPIO pin pair used as DDC bus to read panel EDID data.
i2cdab	<0x00-0xff>	I^2C device address for reading and writing device registers. The device address should be in 8-bit format with the 7-bit slave address assigned to its bits 7:1 and bit 0 set to 0.
ddcdab	<0x00-0xff>	I^2C device address for reading EDID data from display through the DDC bus.
i2cspeed	[10-400]. Units in KHz	Speed of I^2C bus for sDVO device.
ddcspeed	[10-400]. Units in KHz	Speed of I^2C bus for EDID device.
fpinfo		Panel-specific information.

Table 21. Parameter Configuration Format (Sheet 6 of 7)

Name	Range / Value	Description
bkltmethod	Range [0-3] 0 – no backlight 1 – Port Driver 2 – GMCH 3 – ICH ***Note:*** The only supported parameter for internal LVDS is 1 – Port Driver	Instructs which backlight method is required for the panel attached to the given port. If zero is supplied, or the key is not present, then no backlight control is provided.
bkltt1	Range [0 -0xfff]. Units of 1ms => the limit specified in your hardware specifications. For example, the maximum for the CH7307 is 409 ms.	(T1) Time delay between VDD active, and sDVO clock/data active. Zero indicates no delay required.
bkltt2		(T2) Time delay between sDVO clock/data active and backlight enable.
bkltt3		(T3) Time delay between backlight disable and sDVO clock/data inactive.
bkltt4		(T4) Time delay between sDVO clock/data inactive and VDD inactive.
bkltt5		(T5) Minimum delay between VDD inactive, and active.
gpiopinvee	Valid ICH GPIO pin, 0 indexed For example: gpiopinvdd = 3 gpiopinvee = 5 gpiopinenable = 1	GPIO connection for panel power.
gpiopinvdd		GPIO connection for backlight power on/off sequencing signal.
gpiopinbklt		GPIO to enable backlight signal.
UseGMCHClockPin	1 - Flat panel is connected to the clock pin 0 - Flat panel is not connected to the clock pin	This entry is needed when GMCH is selected as backlight control method.
UseGMCHDataPin	1 - Flat panel is connected to the data pin 0 - Flat panel is not connected to the data pin	This entry is needed when GMCH is selected as backlight control method.
dtd		Denotes a Detailed Timing Descriptor (DTD) block. Settings in this section, except for the `flags` parameter, correspond to the Detailed Timing Block described in the VESA standard "Extended Display Identification Data Standard", Version 3, November 13, 1997.
p_clock	Range [0-0x7fffffff]	Pixel clock value in KHz.
h_active	Range 0-4096 [12 bits]	Horizontal Active.
v_active	Range 0-4096 [12 bits]	Vertical Active.
h_sync	Range 0-1024 [10 bits]	Horizontal Sync Offset.
v_sync	Range 0-64 [6 bits]	Vertical Sync Offset.
h_syncp	Range 0-1024 [10 bits]	Horizontal Sync Pulse Offset.
v_syncp	Range 0-64 [6 bits]	Vertical Sync Pulse Width.
h_blank	Range 0-4096 [12 bits]	Horizontal Blanking.
v_blank	Range 0-4096 [12 bits]	Vertical Blanking.
h_border	Range 0-256 [8 bits]	Horizontal Border. Currently not supported.
v_border	Range 0-256 [8 bits]	Vertical Border. Currently not supported.
h_size	Range 0-4096 [12 bits]	Horizontal Size. Currently not supported.

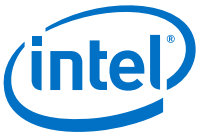


Table 21. Parameter Configuration Format (Sheet 7 of 7)

Name	Range/Value	Description
v_size	Range0-4096 [12 bits]	Vertical size. Currently not supported.
flags	Range [32 bits] Valid values: bit 31 0 - Non-interlaced 1 - Interlaced bit 27 0 - vertical sync polarity active low 1 - vertical sync polarity active high bit 26 0 - horizontal sync polarity active low 1 - horizontal sync polarity active high bit 25 0 - blank sync polarity active high 1 - blank sync polarity active low bit 17 0 - Normal DTD 1 - Panel/display Native DTD All other bits Do not use any other bits; all other bits must be set to 0.	Interlace, Horizontal polarity, Vertical polarity, Sync Configuration, etc. Note that these flags are Intel® EMGD specific and do not correspond to VESA 3.0 flags. For example, to set Interlaced with Horizontal Sync Polarity high (bits 31 and 26), then the flags value = 0x84000000. (Binary = 10000100 00000000 00000000 00000000)
attr	0-0xFFFF	Attribute values that are specific to the sDVO device for the port. See Appendix B, "Port Driver Attributes" for specific attribute IDs and associated values.
id <Attribute ID>	0 -4294967296	id = <value>. Both the Attribute ID and its value should be specified in decimal. For example, to set brightness to 50, you specify id 0 = 50 See Appendix B, "Port Driver Attributes".

3.12 Display Detection and Initialization

The Display Detection and Initialization feature, when enabled, automatically detects displays and allocates ports without the need to change any configuration files. This feature is off by default and can be enabled either through CED or by directly editing the `iegd.inf` file for Microsoft Windows or the `xorg.conf` file for the Linux OS.

To enable the feature via CED, select the `DisplayDetect` option on the CED Chipset Configuration page. Please see Section 3.5, "Creating a New Configuration" on page 27 or CED online help for more information.

Alternatively, you can enable the feature in Microsoft Windows by entering the following line in the `iegd.inf` section [`iegd_SoftwareDeviceSettings_xxx`] (where xxx = chipset/processor code name, for example: `tnc` for Atom E6xx, `plb` for US15W, etc.):

HKR, All\<ConfigID>\General, DisplayDetect, %REG_DWORD%, 1

where <ConfigID> is the configuration ID (without the angle brackets).

To enable the feature in the Linux OS, enter the following line Option setting in the `xorg.conf` file:

Option "Config/<ConfigID>/General/DisplayDetect" "1"

When the display detection feature is enabled, ports are allocated only when the display satisfies the following conditions:

1. The port is not in use (that is, it is not already allocated).
2. The port driver detects the display.

The first port that passes these conditions is allocated. If condition 2 fails for all ports, then the first port in the `PortOrder` setting that passes condition 1 is allocated. If the port is not detectable (specifically the internal LVDS or external LVDS using CH7308), the driver assumes the display is connected. Condition number 2. always passes for these displays.

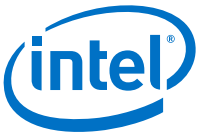
When this feature is disabled, display allocation is done based on `PortOrder` and no display detection is performed.

3.12.1 Display Detect Operation

This section describes the logic of the Display Detection feature and provides several examples.

1. If Display Detect is disabled, the driver uses the first two ports identified in the PortOrder.
2. If Display Detect is enabled and you are using the 1.14 or later version of the VBIOS, the VBIOS performs the display detection. The driver then checks whether the VBIOS returns the display allocations and if it does, the driver does not re-execute the display detection steps.

   If you are not using the version 1.14 or later Legacy VBIOS, then the driver performs display discovery as described in the following steps.
3. The number of displays to be detected is based on the `DisplayConfig` settings in the configuration. If this is set to **Single**, then only one display is detected. If it is set to any other value, a maximum of two displays will be detected.



4. The Intel® EMGD goes through each port in the `PortOrder` settings and attempts to detect a display using the following algorithm:

   a. PortOrder sequence determines display detection. Port allocation shows after the display has been detected. For example:

      PortOrder = “42000” (LVDS, sDVO)

      Displays Connected = LVDS

      Primary display allocation: Searches for a display connected according to the PortOrder sequence. The first detected display is an LVDS, so the Primary display is “LVDS.”

      Secondary display allocation: Searches for a display connected according to the PortOrder sequence. The first non-allocated display detected is sDVO, so the Secondary display is “sDVO.”

   b. With no display detected on any port, then turn off the DisplayDetect option and allocate ports in the order defined by PortOrder. For example:

      PortOrder = “20000”

      Displays Connected = None

      Primary display allocation: Searches for a connected display according to the PortOrder. Because Intel® EMGD detects no displays, the Primary display is set to “sDVO-B.”

   c. The driver cannot detect the presence of a display connected to the Internal LVDS and external LVDS displays connected to some sDVO devices, for example, an LVDS connected to the CH7308. Consequently, the driver assumes that an LVDS display is connected if it is in the PortOrder. If you only want to use the internal LVDS when no external panel (from sDVO) or devices are connected, then put LVDS in the PortOrder after them. For example:

      PortOrder = “24000” (sDVOB, LVDS)

      Display Connected = None

      Primary display allocation: Searches for a display connected according to PortOrder sequence. Since no display is connected and since LVDS is specified in the PortOrder, the driver assumes that an LVDS display is connected. Consequently, set the Primary display to “LVDS.”

   d. When the port drivers do not load for any ports specified in the PortOrder, the driver enables port 4 (LVDS) only. For example:

      PortOrder = “20000” (sDVO)

      PortDrivers = “” (None)

      Primary display allocation: Searches for displays connected according to the PortOrder. Since no port drivers are available for the specified ports, LVDS port 4 is enabled. Consequently, set the Primary display to “LVDS.”

3.12.2 Detectable Displays

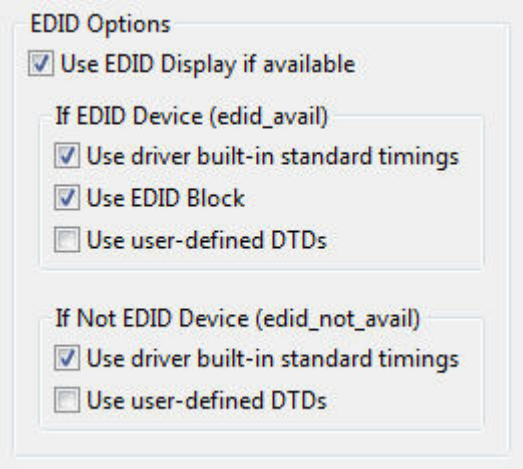
The table below provides a list of displays that are detectable by the Intel® EMGD.

Table 22. Detectable Displays

Transmitter	Display Type	Detectable by Intel® EMGD?
GMCH Integrated LVDS	LVDS	No (assumed attached)
CH7022	VGA Bypass	Yes
CH7036	LVDS Bypass and/or VGA/ HDMI (VGA and HDMI are mutually exclusive)	Yes
CH7307	DVI	Yes
CH7308	LVDS	No (assumed attached)
CH7315	HDMI/DVI	Yes
CH7317b	VGA Bypass	Yes
CH7319	DVI	Yes
CH7320	DVI	Yes
SiI 1362	DVI	Yes
SiI 1364	DVI	Yes

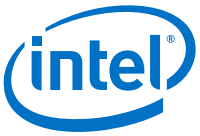
3.13 Advanced EDID Configuration

Shown in the following EDID Options example, the If EDID Device (`edid_avail`) and If Not EDID Device (`edid_not_avail`) options in CED are found on the CRT, sDVO, LVDS, and TV Out configuration pages.



These options control the available timings for any display. Use the `edid_avail` parameter when EDID information is read from the display. If the driver cannot read EDID information from the display or if the `edid` parameter is set to "`0`" (disable), then use the settings of the `edid_not_avail` parameter.

The default behavior of `edid_avail` is to use the driver's built-in standard timings and EDID block and filter modes. The default for `edid_not_avail` uses the driver's built-in standard timings. Please see Table 21 in Section 3.11 for more information on these parameters.



The Intel® EMGD supports three different types of EDID display modes:

1. **Built-in display modes**. These modes are hard-coded in the Intel® EMGD. These modes can be filtered based on the EDID block.
2. **EDID-DTDs**: These are Detailed Timing Descriptors read from the EDID block. EDID can have these DTDs along with other information about the display.
3. **User-specified DTDs** defined in CED. See Section 3.13.2, "User-Specified DTDs" on page 69.

The Advanced EDID Configuration supports different possible combinations of display modes when an EDID display is present along with user-specified DTDs.

3.13.1 Sample Advanced EDID Configurations

The table below presents various EDID configurations and the EDID settings in CED used for those configurations.

Table 23. Sample Advanced EDID Configurations

Configurations	CED Settings	Description
1. Use only filtered built-in and any EDID-DTDs when the display has EDID information 2. Use all built-in modes when the display does not contain EDID information	edid = 1 edid_avail = 3 edid_not_avail = 1	Default values.
1. Use only filtered built-in modes and EDID-DTDs when the display has EDID. 2. Use only user-DTDs otherwise.	edid = 1 edid_avail = 3 edid_not_avail = 4	This configuration allows the Intel® EMGD to use its built-in display modes and the modes provided by the display. If the Intel® EMGD is unable to read EDID information from the display, then the Intel® EMGD uses the user-DTDs defined in CED.
1. Use only user-DTDs regardless of connected display. (Typically used for a custom panel that only supports user-defined DTDs.) 2. Use limited set of timings when a panel EDID is present, but the Intel® EMGD cannot read the EDID information.	edid = 0 edid_avail = (any value) edid_not_avail = 4	Only user-DTDs defined in CED are used.
1. Use EDID-DTDs for an EDID display. 2. Use user-DTDs for a non-EDID display.	edid = 1 edid_avail = 2 edid_not_avail = 4	This configuration uses the EDID-DTDs when detecting an EDID display and EDID information comes from the display. If the driver detects a non-EDID display, then the Intel® EMGD uses user-DTDs defined in CED.
1. Use only EDID-DTDs and user-DTDs for an EDID display. 2. Use user-DTDs only for a non-EDID display.	edid = 1 edid_avail = 6 edid_not_avail = 4	This configuration uses both EDID-DTDs and user-DTDs when the Intel® EMGD detects an EDID display. If the driver detects a non-EDID display, then the Intel® EMGD uses user-DTDs defined in CED.

3.13.2 User-Specified DTDs

CED provides the ability to input DTD data directly. There are numerous sources of DTD data: VESA, panel manufacturers, etc. See Creating a New Customized DTD for more information.

3.14 Using an External PCIe Graphics Adapter as the Primary Device on Windows XP

Note: Intel® EMGD supports this feature only on Intel® Atom™ Processor E6xx and Windows XP.

Intel® EMGD can be configured to work with an external PCIe graphics adapter card as the primary graphics adapter device with the Intel internal graphics device (GMCH) as the secondary graphics device. You can configure your system to boot with a PCIe graphics adapter in the System BIOS (note: check with your BIOS vendor as not all BIOS supports the option). When designating an external PCIe graphics adapter as the primary graphics adapter, the Intel GMCH becomes the secondary graphics device.

Note: The term *secondary* adapter refers to the adapter that is not the *boot-up*, or VGA-Compatible, adapter. The secondary adapter is not necessarily the secondary display as assigned by the OS.

You can configure an external PCIe card to work with the Intel® EMGD as follows:

- The external PCIe card as the primary graphics adapter and the GMCH internal graphics device as the secondary.
- The external PCIe card as the secondary graphics adapter and the GMCH internal graphics device as the primary.

Note: When using an ATI card, first install the ATI driver and do not reboot the machine. Reboot after installing Intel® EMGD.

Intel® EMGD lets you specify which display is primary, secondary, and tertiary. It allows Clone configurations on the internal graphics device when the external PCIe display is the primary graphics adapter. It also allows Clone configurations on the internal graphics device when the external PCIe device is the secondary graphics adapter.

An external PCIe graphics driver runs independently without sharing resources with Intel® EMGD.

The following figures show several configurations when an external PCIe adapter is the primary graphics device and when it is the secondary graphics device.

Figure 24 shows an External PCIe card as the primary graphics adapter card and Intel® EMGD as the secondary. The drivers do not share hardware resources. The OS decides the framebuffer content and handles that by drawing to the respective driver independently.

Figure 24. External PCIe Graphics Card as Primary Driver and Intel® EMGD as Secondary Driver

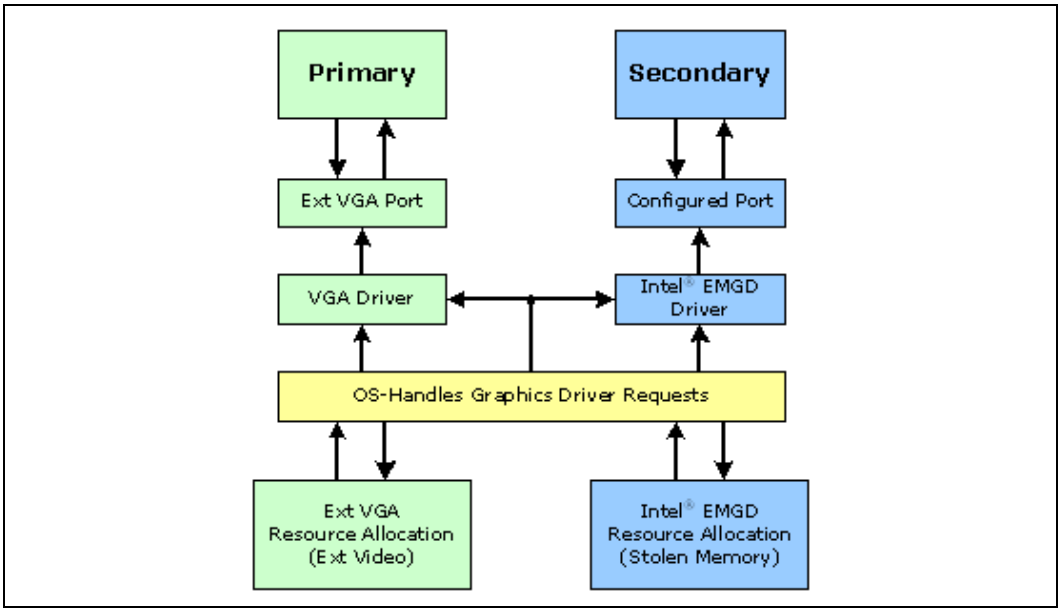


Figure 25 shows the interaction between Intel® EMGD and the External VGA driver when Intel® EMGD is booted as the primary driver. Again, the drivers do not share hardware resources. The OS decides the framebuffer content and handles it by drawing to the respective driver independently.

Figure 25. Intel® EMGD as Primary Driver and External PCIe Graphics Card as Secondary Driver

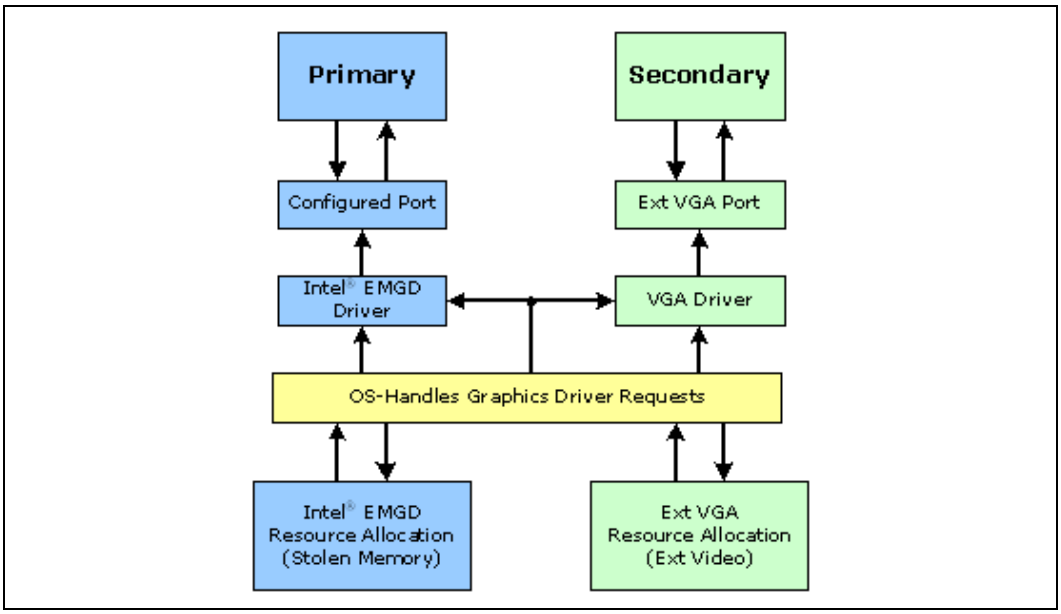
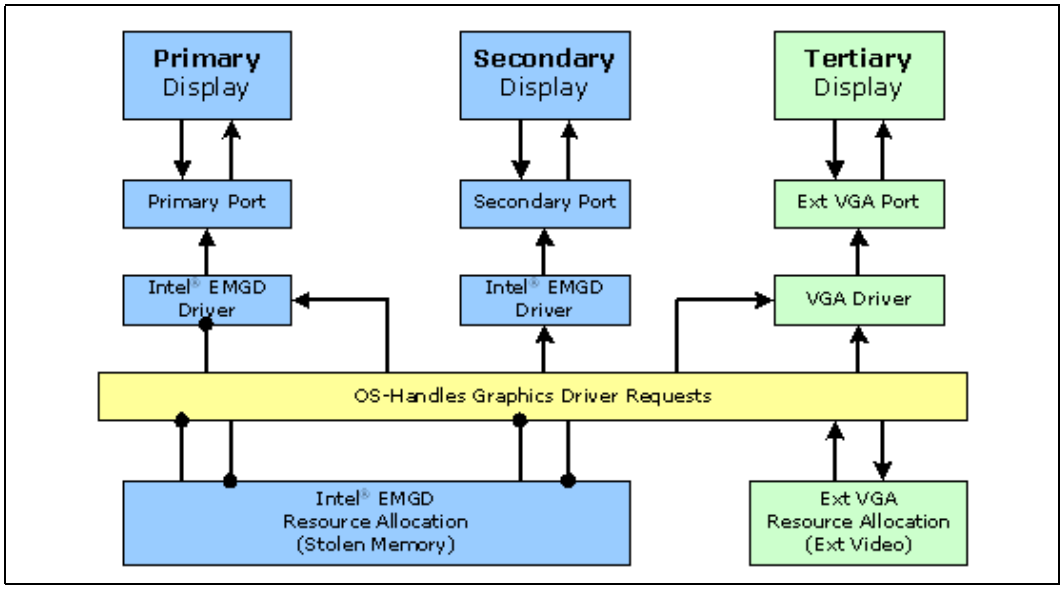


Figure 26 shows a sample configuration where the internal graphics device is primary and configured to use two ports to drive two displays while an external PCIe graphics adapter is used to drive a tertiary display. Note that regardless of the number of ports being assigned to a driver, the external PCIe graphics run independently without sharing resources with Intel® EMGD.

Figure 26. Intel® EMGD as Primary Driver with Two Displays and External PCIe Driving a Tertiary Display



3.15 Enhanced Clone Mode Support

The Enhanced Clone Mode feature lets you specify a clone display size that is different from the primary display. It also allows you to change the clone display size at runtime using the Intel® EMGD Runtime GUI (see Section 5.6, “Viewing and Changing the Driver Configuration from Microsoft Windows” on page 96 or Section 7.5, “Runtime Configuration GUI” on page 162 for Linux systems).

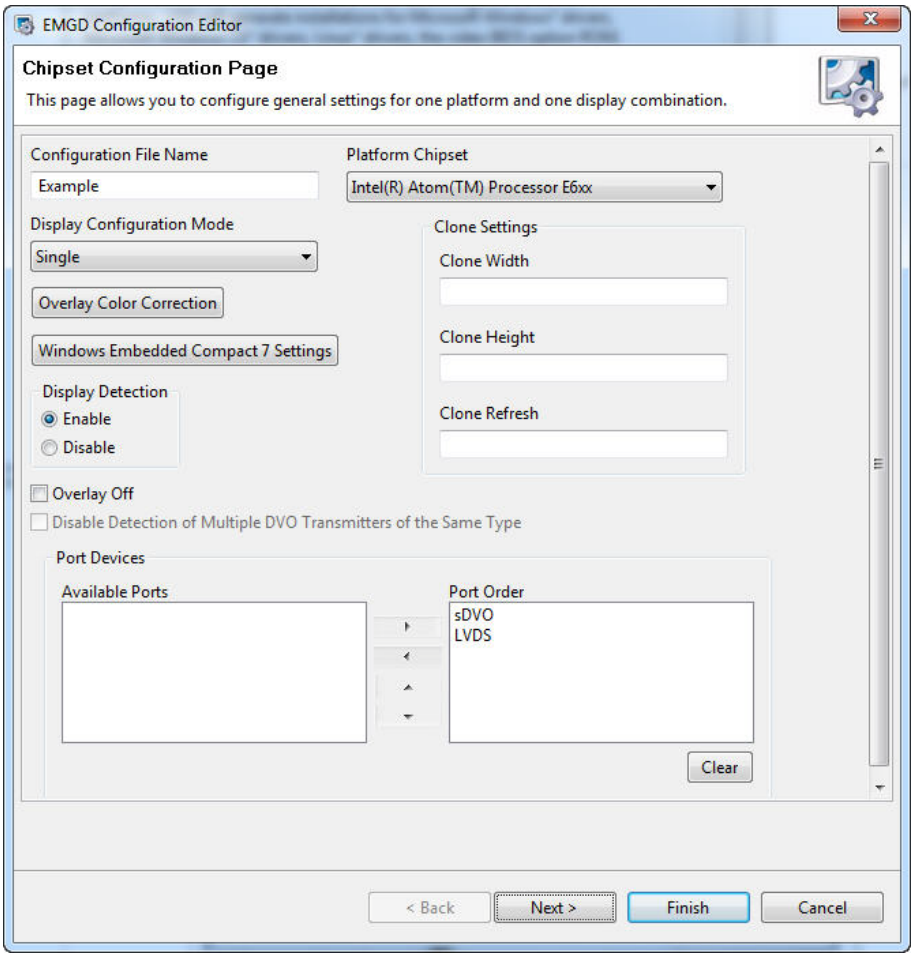
In Clone mode, the framebuffer is always allocated to match the primary display size. On the clone display (secondary display) the image is centered if the display is bigger than the framebuffer. Centering happens only if the requested resolution and refresh rate are not available for the clone display.

Extended Clone mode uses four CED parameters:

- Clone Width — specifies a width for the clone display
- Clone Height — specifies a height for the clone display
- Clone Refresh — specifies a refresh rate for the clone display
- Enable interlace mode — uses interlace mode for the clone display

3.15.1 Clone Mode CED Configuration

The following CED screenshot shows a sample Clone mode setting configuration.



3.16 Scaling and Centering Configurations

This release supports the following scaling and centering configurations:

- Upscaling for the Chrontel CH7308 LVDS Transmitters
- Internal LVDS Scaling With EDID Panels
- Alignment in Clone mode
- sDVO as Primary
- Render Scaling modes to native panels connected to non-scaling port encoders

See the following topics for configuration details:

- "Upscaling for the Chrontel CH7308 LVDS Transmitters"
- "Internal LVDS Scaling with EDID Panels"
- "Centering Primary Display with Scaling Encoders"
- "Enabling Render Scaling on Port Encoders without Hardware Scaling"
- "Alignment in Clone Mode"

3.16.1 Upscaling for the Chrontel CH7308 LVDS Transmitters

The Intel® EMGD can upscale lower-resolution modes (those smaller than the size of the respective panel) to the native size of the panel connected to a Chrontel CH7308* LVDS transmitter.

The Intel® EMGD uses a user-supplied DTD with the native flag set (also known as native DTD) as native timing for the panel connected to either a CH7308 transmitter.

If the user does not supply a native DTD, the Intel® EMGD takes the first available matching `FP info width` and `height` timings as native timing for the panel if standard timings were selected as part of `edid_avail` or `edid_not_avail` flags.

To support upscaling, the LVDS transmitters require setting the pipe to native timing of the panel despite the user-selected resolution. It also requires finding the native timing (also known as native DTD) of the panel based on user-supplied configuration information.

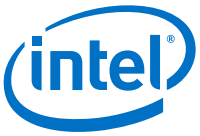
The CH7308 (sDVO) port drivers limit the list of supported modes to the size of panel. The port drivers also mark one of the timings as native DTD as follows (it goes to the next step only if native DTD is not found in the current step).

1. It finds the timing with the user-defined DTD with the native DTD flag set. This becomes the native DTD for the panel.
2. If the panel is an EDID panel and user selected to use EDID DTDs, then the port driver marks the EDID DTD as native DTD.
3. If the user supplies a DTD without the native DTD flag set, then the port driver marks this one as the native DTD.
4. If none of the above steps works, the port driver finds the first matching timing for FP width, height and marks it as native DTD.

If none of the above steps work, then there is no native DTD and no upscaling is performed.

3.16.2 Internal LVDS Scaling with EDID Panels

The Internal LVDS connected to an EDID Panel supports scaling of modes other than native mode. To support this, the port driver exports information to the EDID parser that it can scale. The EDID parser does not remove other modes (that is, non-native modes) from the mode table. It only marks the native mode. When the Intel® EMGD queries the port driver on which modes are supported, the port driver then removes any modes that cannot be scaled (up or down depending on the port's hardware capability). When mode-setting occurs, the second display in Clone mode can indeed support non-native modes even though the panel had EDID. This occurs only if a native mode can be found the port driver can scale. Otherwise, the port driver ignores the scaling information and the Intel® EMGD proceeds normally.



The driver also supports Internal LVDS Scaling on EDID-less panels. The steps that enable this are the same as those described for the scaling of Chrontel LVDS transmitters in Section 3.16.1.

3.16.3 Centering Primary Display with Scaling Encoders

In Clone mode, the Intel® EMGD expects the primary display to have a framebuffer size (OS Aware mode) that matches the display's native size of panel timings. When the user designates a display as primary in a Clone mode configuration and wants to center it (as explained in Section 3.16.5), they may want this setup to align a primary display on a scaling encoder with a secondary one that can only center. This will not work by default for certain port encoders such as the internal LVDS, which default to hardware scaling. But Intel® EMGD has a mechanism to override hardware scaling, thus forcing centering.

When possible, the Intel® EMGD allows centering of 640x480, 800x600, and 1024x768 resolutions on the primary display. In some cases (depending on panels), the image may appear on the top-left. It may also produce unusable output on some displays (such as a TV). Therefore, this type of configuration is more appropriate for LVDS panels.

To disable hardware scaling and force centering for a primary display on the above modes, users only need to set the "Panel_Fit" attribute ("0x12") to "0" (zero).

3.16.4 Enabling Render Scaling on Port Encoders without Hardware Scaling

The Intel® EMGD Render Scaling feature allows the driver to support any one of the standard modes (640x480, 800x600, 1024x768 or 1280x1024) as a drawable framebuffer size output to a native panel and connected via a port encoder that does not hardware scale. To achieve this, the GPU engine repeats all rendering operations twice (from the original OS-targeted back buffer) to a separate front buffer, which is rendered via the 3D engine for scaling. This feature is enabled by turning on the "Panel-Fit" attribute ("0x12") on a port driver that does not support that attribute. But this only happens if there is a native mode timing (see Section 3.16.1 for information about how native mode timing is determined).

Users should be aware that this feature can impact performance and produce scaled output which is inferior in quality to hardware encoder scaling.

3.16.5 Alignment in Clone Mode

In Clone mode, both can be configured with separate timings and different resolutions. Both displays show the same content. In the case where resolutions are different on the cloned displays, the display identified as primary drives the display mode and framebuffer size. In this situation, three options exist for the cloned displays:

- *Panning*: If the clone display is smaller than the primary display, the displayed image can be off the screen with the display showing only a window into the overall image. Panning moves the window, following the cursor.
- *Centering*: If the clone display is larger than the primary display mode, the display image can be centered in the clone display. Black borders are displayed around the image on the display, known as picture-boxing.

- *Scaling:* There are two types of scaling in Clone mode, as described below.
  - *Hardware Encoder Scaling*: This feature adjusts the resolution of the image from the primary display to fit the resolution of the clone display. This permits scaling up to a larger display (upscaling), or scaling down to a smaller display (downscaling). It also allows the full image to be displayed within the full resolution of the clone display.

    Some systems may have cloned displays that cannot scale but have a primary display that can scale such as an internal LVDS. In non-panning modes, i.e., centering/hardware scaling, this display combination results in the primary display (LVDS) scaling up but the clone display centering. Section 3.16.3 explains how to force the primary display to center — thus allowing both displays to center. Or, use *Render Scaling* to make both displays scale up to full size.
  - *Render Scaling:* For clone display, a situation is possible where the primary display uses a hardware scaling port encoder and the secondary display uses a non-scaling port encoder. Assuming both displays are output via native panels, the resulting output should see the primary scaling of any smaller mode to full panel size. But the secondary display will center the smaller modes. *Hardware Encoder Scaling* explains how to align both displays to be centered. Using the Render Scaling feature, the opposite can be achieved. Ensure the non-scaling encoder is primary and enable Render Scaling on that port (see Section 3.16.4). This will make the GPU render-scale the smaller mode and achieve the full panel size. The clone display, now the scaling encoder, will take the render-scaled image as its input (and output) to the clone display panel. This feature will be upgraded in the future so that the clone display can independently take in the original framebuffer image as its input.

§ §

4.0 Video Firmware

4.1 Overview

The Intel Embedded Video BIOS incorporates many of the features and capabilities of the Intel® Embedded Media and Graphics Driver. The 1.16 version of the VBIOS includes support for the following chipsets:

- Intel® Atom™ Processor E6xx
- Intel® System Controller Hub US15W/US15WP/WPT chipset

Enabling the SMSW instructions used when Intel® EMGD VBIOS sets up its caching functions increases the boot speed during POST and system boot. Linux* distributions fall back to text mode as a side effect of the Linux Virtual X86 Engine, which does not work well with SMSW. Caching is vital for the Intel® EMGD VBIOS and it uses SMSW by design. Changes to the Intel® Embedded Media and Graphics Driver VBIOS cannot happen without affecting its performance.

4.2 System Requirements

The new Video BIOS can be built on a host Microsoft Windows* system and moved to the target system. The host system must have a 32-bit Microsoft Windows operating system installed with the capability to execute DOS commands from a command line window.

The target system must contain one of the following Intel chipsets:

- Intel® Atom™ Processor E6xx
- Intel® System Controller Hub US15W/US15WP/WPT chipset

The target system must contain a minimum of 64 MB of RAM.

4.3 Configuring and Building the VBIOS with CED

The Intel® Embedded VBIOS is built with the Intel Configuration Editor (CED). The VBIOS will use the configuration that you specify in CED. The VBIOS is selected to be built when you specify the Video BIOS as a Target OS in your package configuration. After specifying the Video BIOS, follow all CED prompts, and be sure to select “Generate VBIOS” when available. The VBIOS will then be built when you select “Generate Installation” in CED.

Before building your VBIOS, you must set up your DOS environment with the steps below.

1. Download the Open Watcom* C/C++ compiler from http://www.openwatcom.com. The User Build System for the VBIOS relies on the Open Watcom C/C++ compiler to be able to build a 16-bit DOS binary required for the BIOS. The VBIOS has been tested with version 1.7a of the Open Watcom compiler.
2. Install the Open Watcom* C/C++ compiler using the full or complete option. **Do not use the default installation option as it may cause errors when creating the BIOS in CED.**
3. Set up directory paths.
   You must set up the PATH environment variable in DOS to be able to execute the Watcom compiler. If Watcom was installed with its default path, CED will by default be able to use it.

When you generate a VBIOS, CED produces the following folders and files:

- Compiled_VBIOS folder
  - `iegdtsr.exe` (Terminate and Stay Resident executable)

    Note: `iegdtsr.exe` is available only for the US15W platform. Any description about `iegdtsr.exe` in subsequence paragraphs and sections is applicable to the US15W platform only.
  - `VGA.BIN` (Option ROM)
- `IEMGD_HEAD_VBIOS.zip` (this file is generated by the build system)

The `iegdtsr.exe` can be copied to any folder on the target machine. To run the TSR, boot the target machine with DOS, and then run the `iegdtsr.exe` from the DOS command line.

The `VGA.bin` file is the binary option ROM that can be merged with your system BIOS per the instructions provided by your system BIOS vendor.

The `IEMGD_HEAD_VBIOS.zip` file contains default builds of the TSR executable and Option ROM for the various chipsets. The filenames are `iegdtsr-def.exe` and `vga-def.bin` and are located in the `tsr` or `orom` folder of the specific chipset folder (see Figure 27). Refer to Appendix E, “Using the AMI* Video BIOS Utility” for instructions on using the command line utility to merge the VBIOS ROM image with the hardware platform system BIOS ROM image.

For further VBIOS build guidelines, see Section 4.3.3, “Building the VBIOS” on page 80.

See also the following topics:

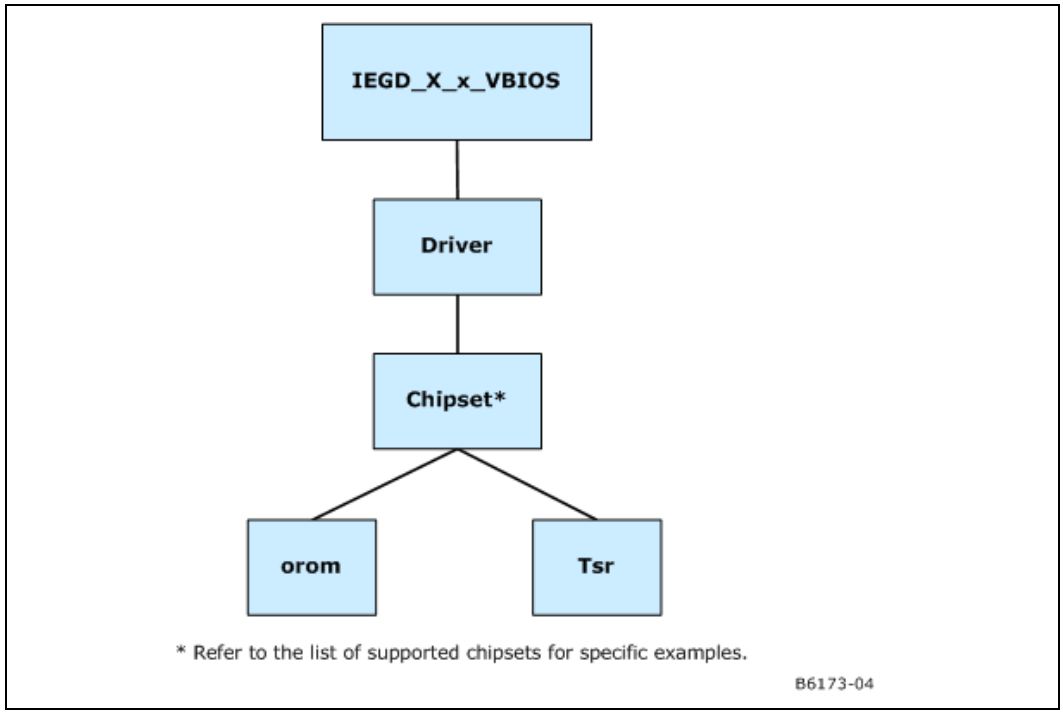
- “Selecting the Build Folder”
- “Configuring the Video BIOS”
- “Building the VBIOS”

4.3.1 Selecting the Build Folder

The 1.16 version of the VBIOS contains specific folders used for creating a VBIOS that is either an option ROM (OROM) that can be merged with the system BIOS, or an executable Terminate and Stay Resident (TSR) program for debugging purposes. There are also separate directories for the different chipsets that are supported. CED will build both the TSR and OROM.

Figure 27 shows the directory structure for the Video BIOS libraries contained within CED.

Figure 27. Video BIOS Directory Structure



4.3.2 Configuring the Video BIOS

Use CED to configure the VBIOS. Display settings will be used the same way as for the driver.

4.3.2.1 COMMON_TO_PORT

This setting allows you to associate standard display names used in most system BIOSs to specific ports that are recognized by Intel® Embedded Media and Graphics Driver (e.g., LVDS, sDVO-B). The VBIOS makes this association when the VBIOS calls the System BIOS Intel® 5F interrupt functions.

This setting is a six digit number, where each digit is associated with one of the system BIOS displays (from left to right):

1 : CRT - Standard analog CRT

2 : TV1 - TV Output 1

3 : EFP1 - DVI Flat Panel 1

4 : LFP - Local Flat Panel (Internal LVDS display)

5 : TV2 - TV Output 2

6 : EFP2 - DVI Flat Panel 2

The example values above show the typical displays and corresponding order used by a system BIOS. However, this may vary depending on how your system BIOS has implemented the displays and the Intel 5F interrupt functions.

The value in each setting associates with the port number. Using the typical settings above, set COMMON_TO_PORT to be 500400, if you want to associate CRT in the system BIOS with the internal CRT (port 5) and LFP in the system BIOS with internal LVDS (port 4) in the VBIOS.

Warning: This feature must be compatible with the system BIOS. If the system BIOS does not properly implement the Intel 5F functions, then using the `COMMON_TO_PORT` feature could cause unpredictable results with the displays. If you are unsure, set `COMMON_TO_PORT` to all zeros (000000) to disable this feature.

Note: The `displaydetect` parameter must be set to `Enabled` in order for the `COMMON_TO_PORT` values to be used.

4.3.2.2 post_display_msg

This setting is a binary setting that enables (1) or disables (0) POST messages to the display.

4.3.2.3 OEM Vendor Strings

The following settings are string values that allow you to set the values that are returned from the Intel 4F interrupt functions.

`oem_string`

`oem_vendor_name`

`oem_product_name`

`oem_product_rev`

4.3.2.4 Default Mode Settings

These settings establish the default VGA or VESA mode to use for the primary (0) and secondary (1) displays. The values should be set to a valid standard VGA or VESA mode (in hexadecimal format, for example, 0x117). Note that a VGA mode can only be set on one display and a second display is disabled unless the `DisplayConfig` parameter is set to `clone` mode.

`default_mode_0`

`default_mode_1`

4.3.2.5 Default Refresh Settings

These settings allow you to specify which refresh rate to use for certain VESA modes on the primary and secondary displays. For example, mode 0x117 specifies refresh rates of 60 Hz, 75 Hz, and 85 Hz. This setting allows use to specify which of those three rates to use (specified in decimal, e.g., default_refresh_0=60).

`default_refresh_0`

`default_refresh_1`

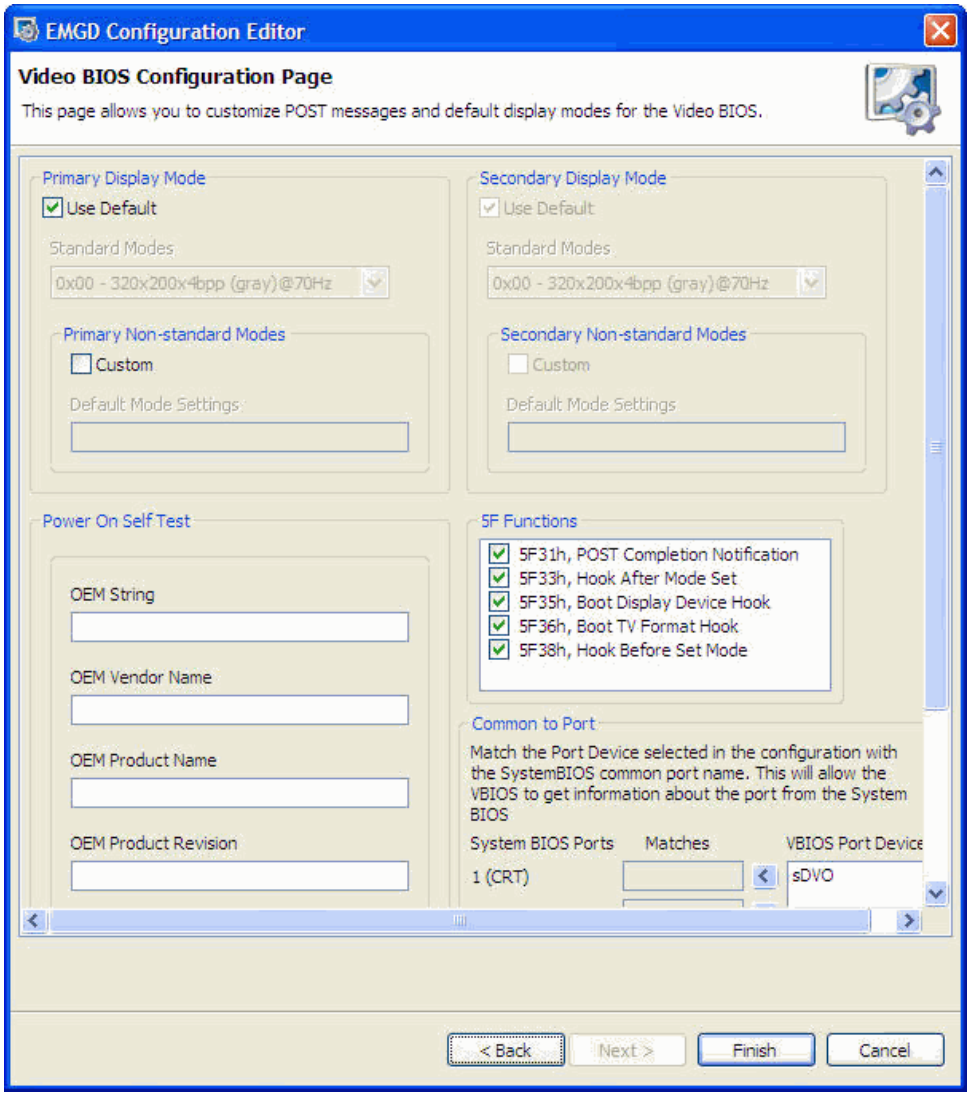
4.3.2.6 default_vga_height

This setting allows you to specify which resolution to use for certain VGA modes. Because only one VGA mode can be supported on both displays, this setting applies to the primary display mode (default_mode_0). For example, mode 3 specifies three possible resolutions: 640x200, 640x350, and 720x400. In this example, setting default_vga_height=350 indicates the resolution 640x350.

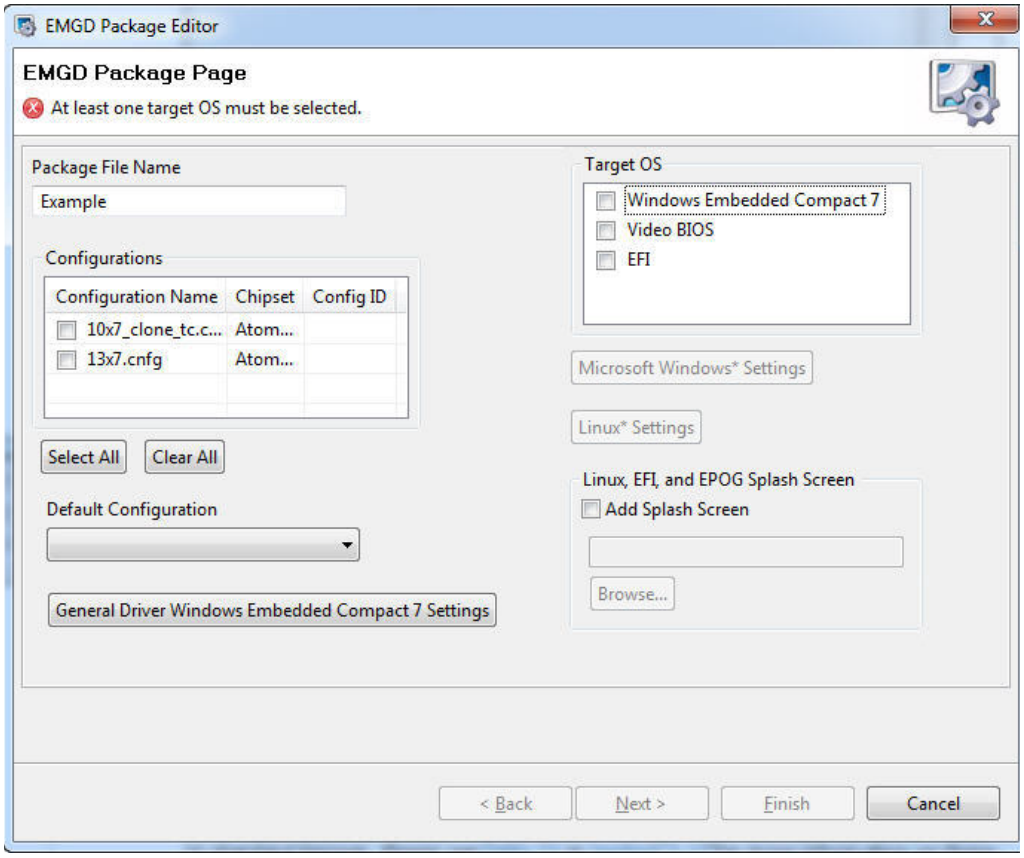
4.3.3 Building the VBIOS

CED is used to build the VBIOS. The following steps and screenshots outline a typical CED VBIOS build procedure.

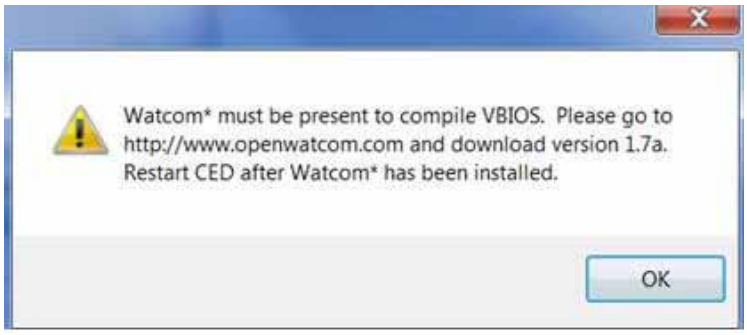
1. Define your configuration via CED, being sure to complete the Video BIOS Configuration Page.

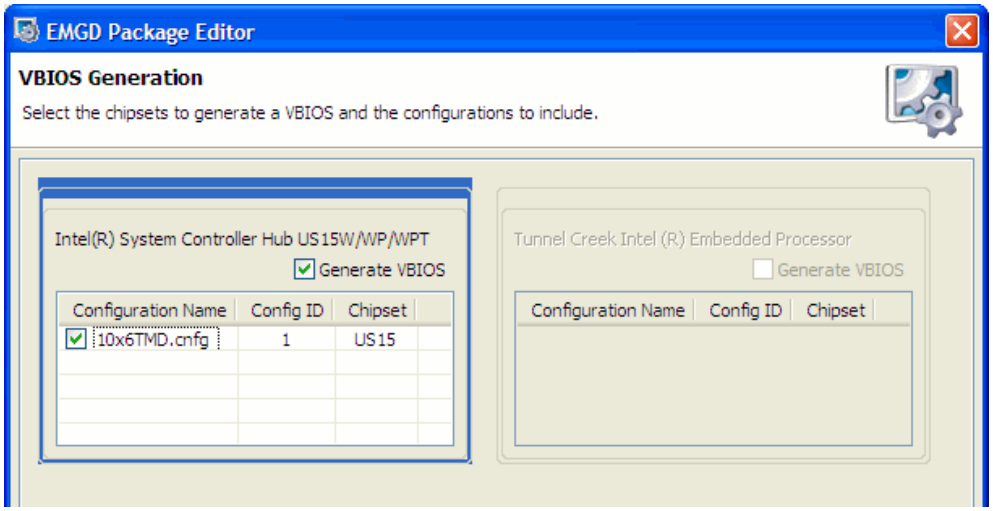


2. When defining the package, be sure to select **Video BIOS** as Target OS.

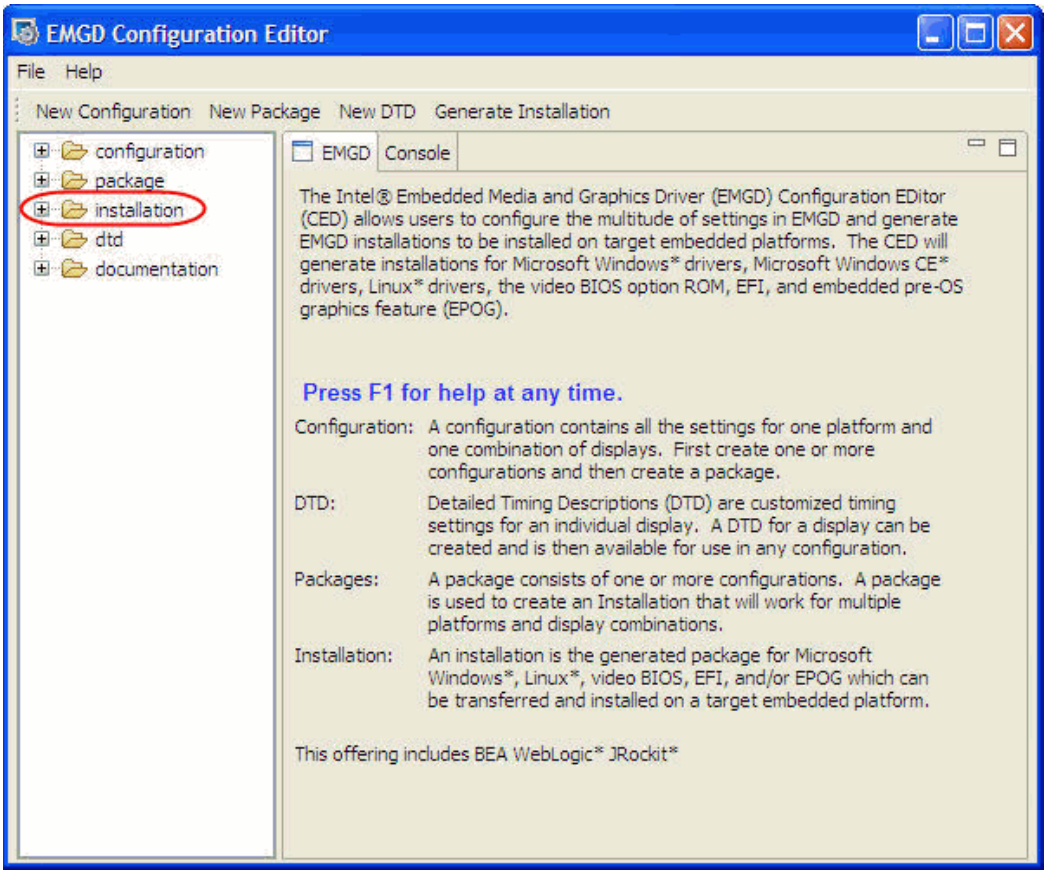


3. Generate the installation. The following message will appear if the Open Watcom* C/C++ compiler has not been installed on the user build system.





4. Generated files should now be in your CED Installation folder.



4.4 VBIOS, Driver Compatibility, and Data Dependencies

The Intel® Embedded Media and Graphics Driver does not depend on any data from the VBIOS, and will either use driver settings or select default values for the attached displays. This allows the driver to properly operate with incompatible BIOS or BIOS replacements.

The Intel® Embedded Media and Graphics Driver will retrieve settings, such as panel ID and other display settings from the Embedded VBIOS. The Embedded VBIOS can configure display timings that can also be used for the Intel® Embedded Media and Graphics Driver.

In the current release, Intel® EMGD supports only pre-configured 10x7 resolution and EDID-detected 13x7 resolutions on an internal LVDS panel.

4.4.1 VESA and VGA Video Modes

The VBIOS supports many VESA and standard VGA modes. See Table 24 and Table 25 for the VGA and VESA modes and vertical refresh rates that are supported by the VBIOS.

Note: Although IBM labeled certain EGA modes with a (*) suffix and the VGA modes with a (+) suffix (such as mode 3, 3* and 3+), the VGA modes are so common that this document does not use the (+) suffix to refer to them.

The actual availability of any particular mode depends on the capabilities of the display device, the amount of memory installed, and other system parameters.

Table 24. Supported VGA Video Display Modes (Sheet 1 of 2)

Video Mode	Pixel Resolution	Color Depth (bpp)	Mode Type	Display Adapter	Font Size	Character Resolution	Dot Clock (MHz)	Horiz. Freq. (KHz)	Vert Freq (Hz)	Video Memory (KB)
00h	320 x 200	16 (gray) (4 bpp)	Text	CGA	8 x 8	40 x 25	25	31.5	70	256
	320 x 350	16 (gray) (4 bpp)		EGA	8 x 14	40 x 25	25	31.5	70	256
	360 x 400	16 (4 bpp)		VGA	9 x 16	40 x 25	28	31.5	70	256
01h	320 x 200	16 (4 bpp)	Text	CGA	8 x 8	40 x 25	25	31.5	70	256
	320 x 350	16 (4 bpp)		EGA	8 x 14	40 x 25	25	31.5	70	256
	360 x 400	16 (4 bpp)		VGA	9 x 16	40 x 25	28	31.5	70	256
02h	640 x 200	16 (gray) (4 bpp)	Text	CGA	8 x 8	80 x 25	25	31.5	70	256
	640 x 350	16 (gray) (4 bpp)		EGA	8 x 14	80 x 25	25	31.5	70	256
	720 x 400	16 (4 bpp)		VGA	9 x 16	80 x 25	28	31.5	70	256

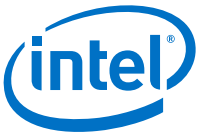


Table 24. Supported VGA Video Display Modes (Sheet 2 of 2)

Video Mode	Pixel Resolution	Color Depth (bpp)	Mode Type	Display Adapter	Font Size	Character Resolution	Dot Clock (MHz)	Horiz. Freq. (KHz)	Vert Freq (Hz)	Video Memory (KB)
03h	640 x 200	16 (4 bpp)	Text	CGA	8 x 8	80 x 25	25	31.5	70	256
	640 x 350	16 (4 bpp)		EGA	8 x 14	80 x 25	25	31.5	70	256
	720 x 400	16 (4 bpp)		VGA	9 x 16	80 x 25	28	31.5	70	256
04h	320 x 200	4	Graph	All	8 x 8	40 x 25	25	31.5	70	256
05h	320 x 200	4 (gray)	Graph	CGA	8 x 8	40 x 25	25	31.5	70	256
	320 x 200	4 (gray)		EGA	8 x 8	40 x 25	25	31.5	70	256
	320 x 200	4		VGA	8 x 8	40 x 25	25	31.5	70	256
06h	640 x 200	2	Graph	All	8 x 8	80 x 25	25	31.5	70	256
07h	720 x 350	Mono	Text	MDA	9 x 14	80 x 25	28	31.5	70	256
	720 x 350	Mono		EGA	9 x 14	80 x 25	28	31.5	70	256
	720 x 400	Mono		VGA	9 x 16	80 x 25	28	31.5	70	256
08h-0Ch	Reserved			-		-				
0Dh	320 x 200	16 (4 bpp)	Graph	E/VGA	8 x 8	40 x 25	25	31.5	70	256
0Eh	640 x 200	16 (4 bpp)	Graph	E/VGA	8 x 8	80 x 25	25	31.5	70	256
0Fh	640 x 350	Mono	Graph	E/VGA	8 x 14	80 x 25	25	31.5	70	256
10h	640 x 350	16 (4 bpp)	Graph	E/VGA	8 x 14	80 x 25	25	31.5	70	256
11h	640 x 480	2 (4 bpp)	Graph	VGA	8 x 16	80 x 30	25	31.5	60	256
12h	640 x 480	16 (4 bpp)	Graph	VGA	8 x 16	80 x 30	25	31.5	60	256
13h	320 x 200	256 (8 bpp)	Graph	VGA	8 x 8	40 x 25	25	31.5	70	256

Table 25. VESA Modes Supported by Video BIOS (Sheet 1 of 2)

Video Mode	Pixel Resolution	Colors (bpp)	Mode Type	Display Adapter	Vertical Frequency (Hz)	Video Memory (MB)
101h	640 x 480	256 (8 bpp)	Graph	VGA	60	0.5
	640 x 480	256 (8 bpp)	Graph	VGA	75	0.5
	640 x 480	256 (8 bpp)	Graph	VGA	85	0.5
103h	800 x 600	256 (8 bpp)	Graph	SVGA	60	1
	800 x 600	256 (8 bpp)	Graph	SVGA	75	1
	800 x 600	256 (8 bpp)	Graph	SVGA	85	1
105h	1024 x 768	256 (8 bpp)	Graph	XVGA	60	1
	1024 x 768	256 (8 bpp)	Graph	XVGA	75	1
	1024 x 768	256 (8 bpp)	Graph	XVGA	85	1
107h	1280 x 1024	256 (8 bpp)	Graph	SXGA	60	2
	1280 x 1024	256 (8 bpp)	Graph	SXGA	75	2
	1280 x 1024	256 (8 bpp)	Graph	SXGA	85	2
111h	640 x 480	64K (16 bpp)	Graph	VGA	60	1
	640 x 480	64K (16 bpp)	Graph	VGA	75	1
	640 x 480	64K (16 bpp)	Graph	VGA	85	1
114h	800 x 600	64K (16 bpp)	Graph	SVGA	60	2
	800 x 600	64K (16 bpp)	Graph	SVGA	75	2
	800 x 600	64K (16 bpp)	Graph	SVGA	85	2
117h	1024 x 768	64K (16 bpp)	Graph	XVGA	60	2
	1024 x 768	64K (16 bpp)	Graph	XVGA	75	2
	1024 x 768	64K (16 bpp)	Graph	XVGA	85	2

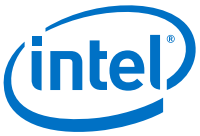


Table 25. VESA Modes Supported by Video BIOS (Sheet 2 of 2)

Video Mode	Pixel Resolution	Colors (bpp)	Mode Type	Display Adapter	Vertical Frequency (Hz)	Video Memory (MB)
11Ah	1280 x 1024	64K (16 bpp)	Graph	SXGA	60	4
	1280 x 1024	64K (16 bpp)	Graph	SXGA	75	4
	1280 x 1024	64K (16 bpp)	Graph	SXGA	85	4
112	640 x 480	16M (32 bpp)	Graph	VGA	60	2
	640 x 480	16M (32 bpp)	Graph	VGA	75	2
	640 x 480	16M (32 bpp)	Graph	VGA	85	2
115	800 x 600	16M (32 bpp)	Graph	SVGA	60	4
	800 x 600	16M (32 bpp)	Graph	SVGA	75	4
	800 x 600	16M (32 bpp)	Graph	SVGA	85	4
118	1024 x 768	16M (32 bpp)	Graph	XVGA	60	4
	1024 x 768	16M (32 bpp)	Graph	XVGA	75	4
	1024 x 768	16M (32 bpp)	Graph	XVGA	85	4
11B	1280 x 1024	16M (32 bpp)	Graph	SXGA	60	8
	1280 x 1024	16M (32 bpp)	Graph	SXGA	75	8
	1280 x 1024	16M (32 bpp)	Graph	SXGA	85	8

Notes: Clone mode is not supported in VBIOS for Atom E6xx.

A single config ID can have multiple port drivers. However, only one display will be activated based on port order priority.

§ §

5.0 Configuring and Installing Microsoft Windows Drivers

5.1 Editing the Microsoft Windows INF File

This section describes the driver-level information (`iegd.inf`) for the Microsoft Windows* operating system, which includes the following[1]:

- Microsoft Windows* XP SP3
- Microsoft Windows* XP Embedded with Embedded Standard 2009

Note: Configuration and Installation information for the Microsoft Windows CE operating system is described in Chapter 6.0, "Configuring and Building Intel® EMGD for Microsoft Windows* Embedded Compact 7."

5.2 Configuration Information

5.2.1 Universal INF Configuration

One INF file can specify multiple display configurations. A ConfigId parameter uniquely identifies each configuration.

The driver reads the PanelId from the System BIOS during initialization and uses the configuration whose ConfigId matches the PanelId. If the System BIOS does not set a valid PanelId (for example, panelId = 0), the driver reads a configuration using ConfigId = 1. (A ConfigId value of 0 is invalid.)

Note: When setting up a multiple display configuration to be used with the PanelID, do not set a default configuration. To have no default configuration, select **None** from the Default Configuration drop-down menu on the EMGD Package Page. See Section 3.6, "Creating a New Package" on page 44 for details.

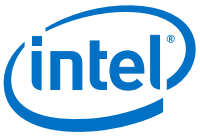
You can override the default behavior by specifying a ConfigId parameter as follows:

```
HKR,, ConfigId, %REG_DWORD%, %DEFAULT_CONFIG_ID%
```

In this case, the driver ignores the PanelId returned by the System BIOS. Instead, the Intel® Embedded Media and Graphics Driver uses the configuration information using the specified ConfigId.

The `PcfVersion` key is generated automatically by the CED utility and is placed in the `[iegd_SoftwareDeviceSettings]` section of the `.inf` file. The default `iegd.inf` file already contains the `PcfVersion` key. Please see Appendix A, "Example INF File" to view a sample `.inf` file.

1. These versions of the drivers are not WHQL (Windows Hardware Quality Labs) certified.



5.2.2 Dual Panel Configuration

Below are the settings required to set the INF file to enable extended display configurations. Typically, these settings are output from the CED utility. However, the INF file may also be edited directly. See Table 26 for a description of these settings.

```
HKR, Config\%DEFAULT_CONFIG_ID%\General, DisplayConfig, %REG_DWORD%, 8
HKR, Config\%DEFAULT_CONFIG_ID%\General, PortOrder, %REG_SZ%, "52000"
```

5.2.3 Chipset Dual Display Example

The table below presents the dual display example for an Intel chipset.

Table 26. Example of Chipset Dual Display Parameter Setting

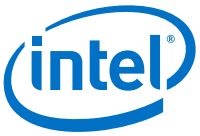
Dual Display Combination	Port Order
Internal LVDS + sDVOB	"42000"
sDVOB + Internal LVDS	"24000"

5.2.4 Creating Registry Settings for Graphics Driver INF File

Use CED to configure the driver settings. It generates the following output, which is then inserted into the graphics driver INF file before driver installation. CED simply translates the configuration options to the INF file. See Table 21 for details on the specific settings and values, which also apply to the settings and values of the INF file. The values of the INF file may also be directly modified. See the example below for syntax and usage. Also, see Appendix A, "Example INF File" for a complete sample INF file.

```
;=============================================================================
[iegd_plb.SoftwareSettings]
AddReg = iegd_SoftwareDeviceSettings_plb
AddReg = iegd_ICDSoftwareSettings

[iegd_tnc.SoftwareSettings]
AddReg = iegd_SoftwareDeviceSettings_tnc
AddReg = iegd_ICDSoftwareSettings

;-----------------------------------------------------------------------------
[iegd_SoftwareDeviceSettings_plb]
HKR,, InstalledDisplayDrivers, %REG_MULTI_SZ%, iegddis
HKR,, MultiFunctionSupported, %REG_MULTI_SZ%, 1
HKR,, VgaCompatible, %REG_DWORD%, 0
HKR,, PcfVersion,    %REG_DWORD%, 0x0700

HKR,, ConfigId, %REG_DWORD%, 1

HKR, ALL\1 , name, %REG_SZ%, "US15"
HKR, ALL\1\General , DisplayConfig, %REG_DWORD%, 1
HKR, ALL\1\General , DisplayDetect, %REG_DWORD%, 1
HKR, ALL\1\General , PortOrder, %REG_SZ%, "24000"
HKR, ALL\1\General , DxvaOptions, %REG_DWORD%, 1
HKR, ALL\1\Port\4\General , name, %REG_SZ%, "LVDS10x7"
HKR, ALL\1\Port\4\General , Rotation, %REG_DWORD%, 0
HKR, ALL\1\Port\4\General , Edid, %REG_DWORD%, 1
HKR, ALL\1\Port\4\General , EdidAvail, %REG_DWORD%, 3
HKR, ALL\1\Port\4\General , EdidNotAvail, %REG_DWORD%, 4
HKR, ALL\1\Port\4\General , CenterOff, %REG_DWORD%, 1
HKR, ALL\1\Port\4\FpInfo , bkltmethod, %REG_DWORD%, 1
HKR, ALL\1\Port\4\FpInfo , BkltT1, %REG_DWORD%, 60
HKR, ALL\1\Port\4\FpInfo , BkltT2, %REG_DWORD%, 200
HKR, ALL\1\Port\4\FpInfo , BkltT3, %REG_DWORD%, 200
```

```
HKR, ALL\1\Port\4\FpInfo , BkltT4, %REG_DWORD%, 50
HKR, ALL\1\Port\4\FpInfo , BkltT5, %REG_DWORD%, 400
HKR, ALL\1\Port\4\Dtd\1 , PixelClock, %REG_DWORD%, 65000
HKR, ALL\1\Port\4\Dtd\1 , HorzActive, %REG_DWORD%, 1024
HKR, ALL\1\Port\4\Dtd\1 , HorzSync, %REG_DWORD%, 24
HKR, ALL\1\Port\4\Dtd\1 , HorzSyncPulse, %REG_DWORD%, 136
HKR, ALL\1\Port\4\Dtd\1 , HorzBlank, %REG_DWORD%, 320
HKR, ALL\1\Port\4\Dtd\1 , VertActive, %REG_DWORD%, 768
HKR, ALL\1\Port\4\Dtd\1 , VertSync, %REG_DWORD%, 3
HKR, ALL\1\Port\4\Dtd\1 , VertSyncPulse, %REG_DWORD%, 6
HKR, ALL\1\Port\4\Dtd\1 , VertBlank, %REG_DWORD%, 38
HKR, ALL\1\Port\4\Dtd\1 , Flags, %REG_DWORD%, 0x20000
HKR, ALL\1\Port\4\Attr , 27, %REG_DWORD%, 0
HKR, ALL\1\Port\4\Attr , 26, %REG_DWORD%, 18
HKR, ALL\1\Port\4\Attr , 60, %REG_DWORD%, 1
HKR, ALL\1\Port\2\General , name, %REG_SZ%, "SDVOB"
HKR, ALL\1\Port\2\General , Rotation, %REG_DWORD%, 0
HKR, ALL\1\Port\2\General , Edid, %REG_DWORD%, 1
HKR, ALL\1\Port\2\General , EdidAvail, %REG_DWORD%, 3
HKR, ALL\1\Port\2\General , EdidNotAvail, %REG_DWORD%, 4
HKR, ALL\1\Port\2\General , CenterOff, %REG_DWORD%, 1

HKR,, No_D3D, %REG_DWORD%, 0

HKR,, PortDrivers, %REG_SZ%, "sdvo lvds"
HKR, ALL\1\General, DxvaOptions, %REG_DWORD%, 0x00000001

HKCU,"Software\Microsoft\Avalon.Graphics",,,""
HKCU,"Software\Microsoft\Avalon.Graphics",DisableHWAcceleration,%REG_DWORD%, 1

;-------------------------------------------------------------------------------
[iegd_SoftwareDeviceSettings_tnc]
HKR,, InstalledDisplayDrivers, %REG_MULTI_SZ%, iegddis
HKR,, MultiFunctionSupported, %REG_MULTI_SZ%, 1
HKR,, VgaCompatible, %REG_DWORD%, 0
HKR,, PcfVersion,    %REG_DWORD%, 0x0700

HKR,, ConfigId, %REG_DWORD%, 1

HKR, ALL\1 , name, %REG_SZ%, "Atom_E6xx_13X7"
HKR, ALL\1\General , DisplayConfig, %REG_DWORD%, 1
HKR, ALL\1\General , DisplayDetect, %REG_DWORD%, 1
HKR, ALL\1\General , PortOrder, %REG_SZ%, "24000"
HKR, ALL\1\General , DxvaOptions, %REG_DWORD%, 1
HKR, ALL\1\Port\4\General , name, %REG_SZ%, "LVDS13x7"
HKR, ALL\1\Port\4\General , Rotation, %REG_DWORD%, 0
HKR, ALL\1\Port\4\General , CenterOff, %REG_DWORD%, 1
HKR, ALL\1\Port\4\General , Edid, %REG_DWORD%, 1
HKR, ALL\1\Port\4\General , EdidAvail, %REG_DWORD%, 3
HKR, ALL\1\Port\4\General , EdidNotAvail, %REG_DWORD%, 4
HKR, ALL\1\Port\4\FpInfo , bkltmethod, %REG_DWORD%, 1
HKR, ALL\1\Port\4\FpInfo , BkltT1, %REG_DWORD%, 60
HKR, ALL\1\Port\4\FpInfo , BkltT2, %REG_DWORD%, 200
HKR, ALL\1\Port\4\FpInfo , BkltT3, %REG_DWORD%, 200
HKR, ALL\1\Port\4\FpInfo , BkltT4, %REG_DWORD%, 50
HKR, ALL\1\Port\4\FpInfo , BkltT5, %REG_DWORD%, 400
HKR, ALL\1\Port\4\Dtd\1 , PixelClock, %REG_DWORD%, 72300
HKR, ALL\1\Port\4\Dtd\1 , HorzActive, %REG_DWORD%, 1366
HKR, ALL\1\Port\4\Dtd\1 , HorzSync, %REG_DWORD%, 48
HKR, ALL\1\Port\4\Dtd\1 , HorzSyncPulse, %REG_DWORD%, 32
HKR, ALL\1\Port\4\Dtd\1 , HorzBlank, %REG_DWORD%, 160
HKR, ALL\1\Port\4\Dtd\1 , VertActive, %REG_DWORD%, 768
HKR, ALL\1\Port\4\Dtd\1 , VertSync, %REG_DWORD%, 3
HKR, ALL\1\Port\4\Dtd\1 , VertSyncPulse, %REG_DWORD%, 5
HKR, ALL\1\Port\4\Dtd\1 , VertBlank, %REG_DWORD%, 22
HKR, ALL\1\Port\4\Dtd\1 , Flags, %REG_DWORD%, 0x20000
HKR, ALL\1\Port\4\Attr , 27, %REG_DWORD%, 0
```



```
HKR, ALL\1\Port\4\Attr , 26, %REG_DWORD%, 18
HKR, ALL\1\Port\4\Attr , 60, %REG_DWORD%, 1
HKR, ALL\1\Port\4\Attr , 70, %REG_DWORD%, 100
HKR, ALL\1\Port\4\Attr , 71, %REG_DWORD%, 20300
HKR, ALL\1\Port\2\General , name, %REG_SZ%, "SDVOB"
HKR, ALL\1\Port\2\General , Rotation, %REG_DWORD%, 0
HKR, ALL\1\Port\2\General , Edid, %REG_DWORD%, 1
HKR, ALL\1\Port\2\General , EdidAvail, %REG_DWORD%, 3
HKR, ALL\1\Port\2\General , EdidNotAvail, %REG_DWORD%, 4
HKR, ALL\1\Port\2\General , CenterOff, %REG_DWORD%, 1

HKR, ALL\2 , name, %REG_SZ%, "Atom_E6xx_10X7"
HKR, ALL\2\General , DisplayConfig, %REG_DWORD%, 1
HKR, ALL\2\General , DisplayDetect, %REG_DWORD%, 1
HKR, ALL\2\General , PortOrder, %REG_SZ%, "24000"
HKR, ALL\2\General , DxvaOptions, %REG_DWORD%, 1
HKR, ALL\2\Port\4\General , name, %REG_SZ%, "LVDS10x7"
HKR, ALL\2\Port\4\General , Rotation, %REG_DWORD%, 0
HKR, ALL\2\Port\4\General , CenterOff, %REG_DWORD%, 1
HKR, ALL\2\Port\4\General , Edid, %REG_DWORD%, 1
HKR, ALL\2\Port\4\General , EdidAvail, %REG_DWORD%, 3
HKR, ALL\2\Port\4\General , EdidNotAvail, %REG_DWORD%, 4
HKR, ALL\2\Port\4\FpInfo , bkltmethod, %REG_DWORD%, 1
HKR, ALL\2\Port\4\FpInfo , BkltT1, %REG_DWORD%, 60
HKR, ALL\2\Port\4\FpInfo , BkltT2, %REG_DWORD%, 200
HKR, ALL\2\Port\4\FpInfo , BkltT3, %REG_DWORD%, 200
HKR, ALL\2\Port\4\FpInfo , BkltT4, %REG_DWORD%, 50
HKR, ALL\2\Port\4\FpInfo , BkltT5, %REG_DWORD%, 400
HKR, ALL\2\Port\4\Dtd\1 , PixelClock, %REG_DWORD%, 65000
HKR, ALL\2\Port\4\Dtd\1 , HorzActive, %REG_DWORD%, 1024
HKR, ALL\2\Port\4\Dtd\1 , HorzSync, %REG_DWORD%, 24
HKR, ALL\2\Port\4\Dtd\1 , HorzSyncPulse, %REG_DWORD%, 136
HKR, ALL\2\Port\4\Dtd\1 , HorzBlank, %REG_DWORD%, 320
HKR, ALL\2\Port\4\Dtd\1 , VertActive, %REG_DWORD%, 768
HKR, ALL\2\Port\4\Dtd\1 , VertSync, %REG_DWORD%, 3
HKR, ALL\2\Port\4\Dtd\1 , VertSyncPulse, %REG_DWORD%, 6
HKR, ALL\2\Port\4\Dtd\1 , VertBlank, %REG_DWORD%, 38
HKR, ALL\2\Port\4\Dtd\1 , Flags, %REG_DWORD%, 0x20000
HKR, ALL\2\Port\4\Attr , 27, %REG_DWORD%, 0
HKR, ALL\2\Port\4\Attr , 26, %REG_DWORD%, 18
HKR, ALL\2\Port\4\Attr , 60, %REG_DWORD%, 1
HKR, ALL\2\Port\4\Attr , 70, %REG_DWORD%, 100
HKR, ALL\2\Port\4\Attr , 71, %REG_DWORD%, 20300
HKR, ALL\2\Port\2\General , name, %REG_SZ%, "SDVOB"
HKR, ALL\2\Port\2\General , Rotation, %REG_DWORD%, 0
HKR, ALL\2\Port\2\General , Edid, %REG_DWORD%, 1
HKR, ALL\2\Port\2\General , EdidAvail, %REG_DWORD%, 3
HKR, ALL\2\Port\2\General , EdidNotAvail, %REG_DWORD%, 4
HKR, ALL\2\Port\2\General , CenterOff, %REG_DWORD%, 1

HKR,, No_D3D, %REG_DWORD%, 0

HKR,, PortDrivers, %REG_SZ%, "sdvo lvds"
HKR, ALL\1\General, DxvaOptions, %REG_DWORD%, 0x00000001

;-------------------------------------------------------------------------------
[iegd_ICDSoftwareSettings]
HKLM, "SOFTWARE\Microsoft\Windows NT\CurrentVersion\OpenGLDrivers\iegddis", DLL,
%REG_SZ%, iegdglga
HKLM, "SOFTWARE\Microsoft\Windows NT\CurrentVersion\OpenGLDrivers\iegddis",
DriverVersion, %REG_DWORD%, 0x00000001
HKLM, "SOFTWARE\Microsoft\Windows NT\CurrentVersion\OpenGLDrivers\iegddis", Flags,
%REG_DWORD%, 0x00000001
HKLM, "SOFTWARE\Microsoft\Windows NT\CurrentVersion\OpenGLDrivers\iegddis",
Version, %REG_DWORD%, 0x00000002

;===============================================================================
```

```
[Strings]

;-------------------------------------------------------------------------
; Localizable Strings
;-------------------------------------------------------------------------
Intel="Intel Corporation"
DiskDesc="Embedded Installation"

i900G0="US15 Intel® Embedded Media and Graphics Driver Function 0"
iTNC0="Atom™ E6xx Intel® Embedded Media and Graphics Driver Function 0"
iTNC1="Atom™ E6xx Intel® Embedded Media and Graphics Driver Extension"

;-------------------------------------------------------------------------
; Non Localizable Strings
;-------------------------------------------------------------------------
SERVICE_BOOT_START     = 0x0
SERVICE_SYSTEM_START   = 0x1
SERVICE_AUTO_START     = 0x2
SERVICE_DEMAND_START   = 0x3
SERVICE_DISABLED       = 0x4

SERVICE_KERNEL_DRIVER  = 0x1

SERVICE_ERROR_IGNORE   = 0x0; Continue on driver load fail
SERVICE_ERROR_NORMAL   = 0x1; Display warn, but continue
SERVICE_ERROR_SEVERE   = 0x2; Attempt LastKnownGood
SERVICE_ERROR_CRITICAL = 0x3; Attempt LastKnownGood, BugCheck

REG_EXPAND_SZ = 0x00020000
REG_MULTI_SZ  = 0x00010000
REG_DWORD     = 0x00010001
REG_SZ        = 0x00000000
```

5.2.5 Dynamic Port Driver Configuration

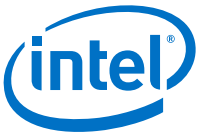
The Intel® Embedded Media and Graphics Driver supports many third-party digital transmitters connected to the sDVO ports of the SCH though device drivers called port drivers. These port drivers are dynamically loaded at startup. The driver configuration can be modified to add or remove availability of specific port drivers.

This section describes the portions of the `iegd.inf` file that can be modified to either add or remove a port driver for the Microsoft Windows version of the Intel® Embedded Media and Graphics Driver.

5.2.5.1 iegd.PortDrvs_xxx

The first step in either adding or removing a port driver is to identify the family of the chipset you are using. Next locate the appropriate [iegd.PortDrvs_xxx] section for your graphics family. Below are the default settings for the blocks of associated port drivers for a particular graphics chipset family:

```
[iegd.PortDrvs_plb]
sdvo.sys
lvds.sys

[iegd.PortDrvs_tnc]
sdvo.sys
lvds.sys
```



To remove one or more port drivers, delete the associated line from the `iegd.PortDrvs_xxx` block. To add a port driver, add the associated line into the appropriate `iegd.PortDrvs_xxx` block. For example, to add a new port driver for a device named “NewPD”, add the following line to the `iegd.PortDrvs_alm` block:

```
NewPD.sys
```

5.2.5.2 SourceDisksFiles

To either add or remove a port driver, identify the specific port driver file names in the `SourceDisksFiles` blocks. The default settings are as follows:

```
[SourceDisksFiles]
iegdmini.sys    = 1
iegdckey.vp     = 1
iegdmsys.vp     = 1
iegdcagt.cpa    = 1
iegdcagt.vp     = 1
iegddis.dll     = 1
iegd3dga.dll    = 1
iegdglga.dll    = 1
libGLES_CM.dll = 1
libGLESv2.dll  = 1
lvds.sys        = 1
sdvo.sys        = 1
sdvo.vp         = 1
lvds.vp         = 1
```

To remove a port driver, delete the associated line in the `[SourceDisksFiles]` block. To add a port driver, add the associated line to the block. For example, to add a port driver for a device whose driver is named `NewPD.sys`, add the following line:

```
NewPD.sys   = 1
```

5.2.5.3 PortDrivers Registry Key

Modify the registry key in the appropriate `[iegd_SoftwareDeviceSettings_xxx]` section that defines the list of available port drivers. Below are the default values of this registry key in the `iegd.inf` file:

For the `[iegd_SoftwareDeviceSettings_tnc]` block:

```
HKR,, PortDrivers, %REG_SZ%, “sdvo lvds”
```

Remove or add port driver names as appropriate to the list of port drivers specified within the quoted string. For example, to add support for a new port driver named “NewPD”, the registry key would be defined as follows:

```
HKR,, PortDrivers, %REG_SZ%, "lvds NewPD"
```

5.2.6 Changing Default Display Mode

After installing the Intel® Embedded Media and Graphics Driver, Microsoft Windows selects a default display mode for the initial startup of the system. This is an 800 x 600 resolution in 8-bit, 16-bit, or even 32-bit color mode.

In some cases, particularly with EDID-less LVDS displays, the 640 x 480 resolution may not be supported, so the default mode selected by Microsoft Windows must be changed. Otherwise, the display may not work after installation of the Intel® Embedded Media and Graphics Driver.

This default mode can be changed by adding the following registry keys to the `[iegd_SoftwareDeviceSettings]` section of the `iegd.inf` file:

```
HKR,, DefaultSettings.XResolution, %REG_DWORD%, 1024
HKR,, DefaultSettings.YResolution, %REG_DWORD%, 768
HKR,, DefaultSettings.BitsPerPel,  %REG_DWORD%, 32
HKR,, DefaultSettings.VRefresh,    %REG_DWORD%, 60
```

The example above makes the default resolution 1024 x 768, with a 32-bit color depth and a refresh rate of 60 MHz.

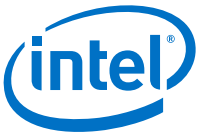
5.2.7 Creating an .sld file for Microsoft Windows XP Embedded Systems

Microsoft Windows XP Embedded* operating systems require the use of an `.sld` (system level definitions) file. The following steps detail how to create such a file for Intel® EMGD from your custom `iegd.inf` file that you created using CED.

1. Run Component Designer.
2. In the **File** menu, select **Import**.
3. In the **Choose File for Import** dialog, select **Setup Information files (*.inf)**. in the **File of type** drop-down list.
4. Select **iegd.inf** from installation directory.
5. In the **Inf Processing Options** dialog, select **Automatic** in the **Parsing Options** dialog and click **OK**.
6. Click **Start** in the **Import File** dialog box. Close the dialog on completion. There should not be any errors.
7. If there are no errors, **Save** the `.sld` file.
8. Run Component Database Manager and import the `.sld` file created above.

Note: Multiple versions will be created.

9. To move the binaries, copy the Intel® EMGD/driver files into the root repository: `\Windows Embedded Data\Repository`
10. In Target Designer, all Intel® EMGD files are found under `Hardware\Devices\Display Adapters` and can be selected by dragging and dropping into your build.



5.3 Installing the Intel® Embedded Media and Graphics Driver on Microsoft Windows

You can install and uninstall the Intel® Embedded Media and Graphics Driver on a Microsoft Windows system by using the `setup.exe` program located in the `IEMGD_HEAD_Windows\Utilities` folder. The following procedure shows how to install the Intel® Embedded Media and Graphics Driver. Section 5.4, "Uninstalling the Current Version of the Driver" on page 94 provides instructions for uninstalling the current version of the Intel® Embedded Media and Graphics Driver.

5.3.1 Silent Installation

Intel® Embedded Media and Graphics Driver supports silent installation through an option in `setup.exe`. With command line installation, add the parameter "/s" (case insensitive), for example **setup.exe /s** at the command prompt. When this option is used, the installation does not display any messages or splash screen except the warning messages about Intel® EMGD not being WHQL compliant. After the silent installation, a message box prompts the user to reboot the system.

Note: To disable the Windows WHQL compliance warning messages, use the Windows **System Properties** -> **Hardware** -> **Driver Signing** -> **Ignore** option.

To allow silent installation with an automatic reboot but without the reboot dialog box stopping the installation, use the option "/r" following the `setup.exe` command, for example, **setup.exe /r**. When the driver installation completes, the system will automatically reboot without any prompts.

To allow silent installation without the reboot dialog box stopping the installation, use the option "/nr" following the setup.exe command, for example, **setup.exe /nr**. This will allow you to do any additional installation or other steps before rebooting. Your install process or the end user will be responsible to do their own reboot without being prompted that a reboot is necessary.

5.4 Uninstalling the Current Version of the Driver

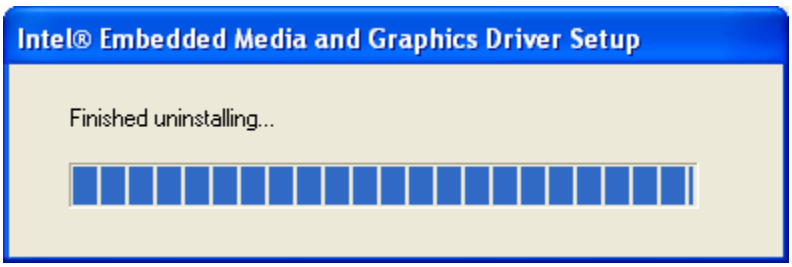
You can use the `setup.exe` Microsoft Windows GUI program to remove the driver from your system. When you run the uninstaller program, it removes the following items from the system:

- The Intel® Embedded Media and Graphics Driver
- The `.inf` and `.pnf` files from the `windows\inf` folder.
- The `DisplayPage.dll` and `qt-mt332.dll` from the `windows\system32` folder
- Data registry items by running `regsvr32.exe` with the uninstall option.

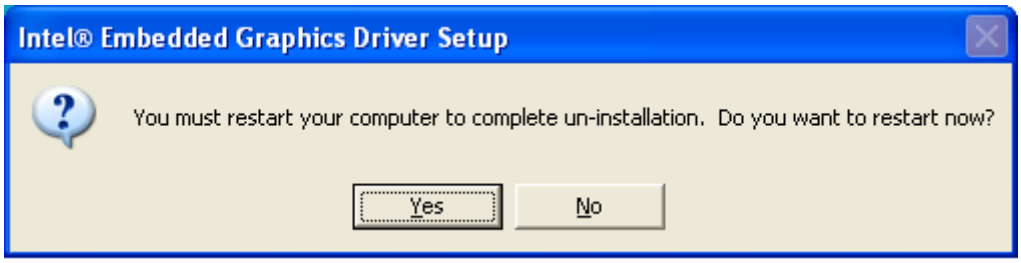
Warning: If you have a previous version of the Intel® Embedded Media and Graphics Driver installed on your system, you must remove it. Do not use the current version of the Intel® Embedded Media and Graphics Driver Install program to uninstall previous versions of the driver. If you do, unpredictable results may occur. You can use this program only to uninstall the driver from the current version. Each version of the driver has its own version of the installer/uninstaller utility.

1. Click the `setup.exe` icon located in the `Utilities` subfolder of the `Windows` folder.
2. In the dialog box, select **Uninstalls driver and application files**, and then click **Next**. The following prompt appears:

3. Click **Yes** to remove the driver. A progress bar displays and when the driver has been removed, the following screen appears.



4. To complete the uninstallation, you must restart your system. If you want to restart your system now, click **Yes** in the following dialog box.



5.5 Runtime Operation

Resolution, refresh rate, and color bit depth can be changed after installation and reboot via a Microsoft Windows display property sheet. Other operations such as enabling and disabling ports (display output), rotation, port configuration, and attribute control are accessible via the standard display driver escape protocol.

5.6 Viewing and Changing the Driver Configuration from Microsoft Windows

Note: For correct display, `EMGDGUI` requires the MS Sans Serif(8) font to be installed in the system font folder.

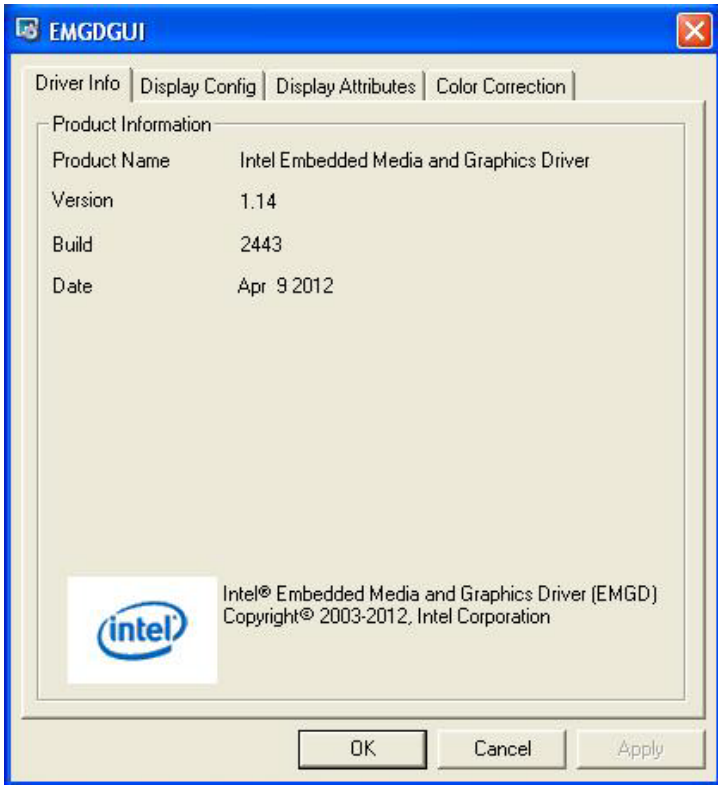
You can change certain configuration attributes of the Intel® Embedded Media and Graphics Driver using the `emgdgui.exe` program located in the `\Utilities` folder. On Microsoft Windows XP systems, you can access the Intel® Embedded Media and Graphics Driver configuration on the display properties setting tab. This program launches the Intel® Embedded Media and Graphics Driver Configuration GUI that consists of the following four tabs:

- **Driver Info** — Contains the driver information.
- **Display Config** — Contains current display information and allows configuration of display configurations, display resolutions and bit depth for primary and secondary displays, flip, rotation, and enabling/disabling for a given port.
- **Display Attributes** — Contains the supported Port Driver (PD) attributes and allows configuration of PD attributes.
- **Color Correction** — Contains color-correction information for the framebuffer and overlay. Using this tab, you can change the framebuffer and overlay color settings.

To view or change the driver settings using the GUI interface, follow this procedure.

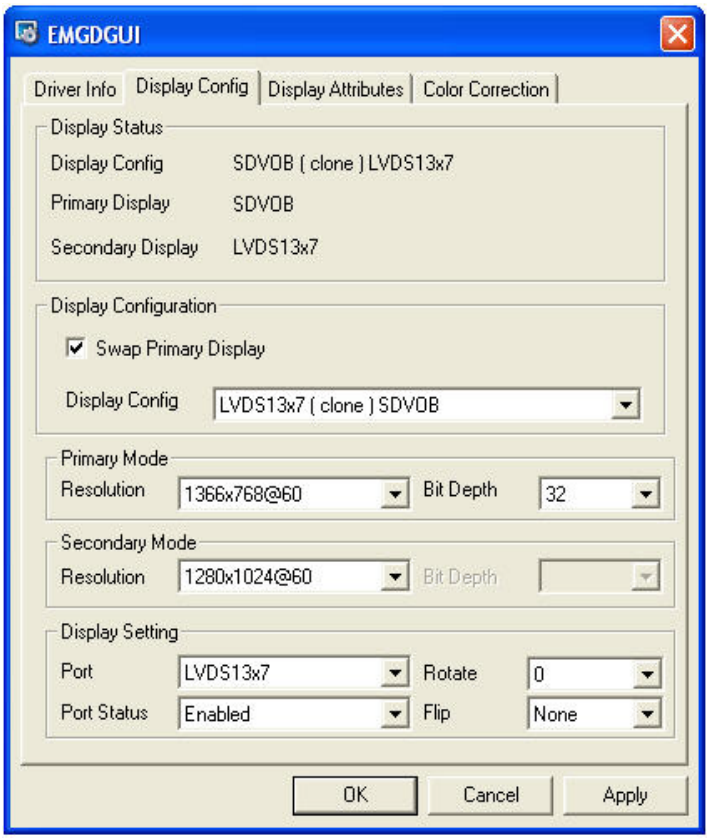
1. Double-click the `emgdgui.exe` icon in the `Utilities` folder.
   To change display configuration, mode, and display setting, select **Display Config**.

Figure 28. Example Runtime Configuration GUI — Driver Info Tab



2. Click the **Display Config** tab to show the current configuration.

Figure 29. Example Runtime Configuration GUI — Display Config Tab



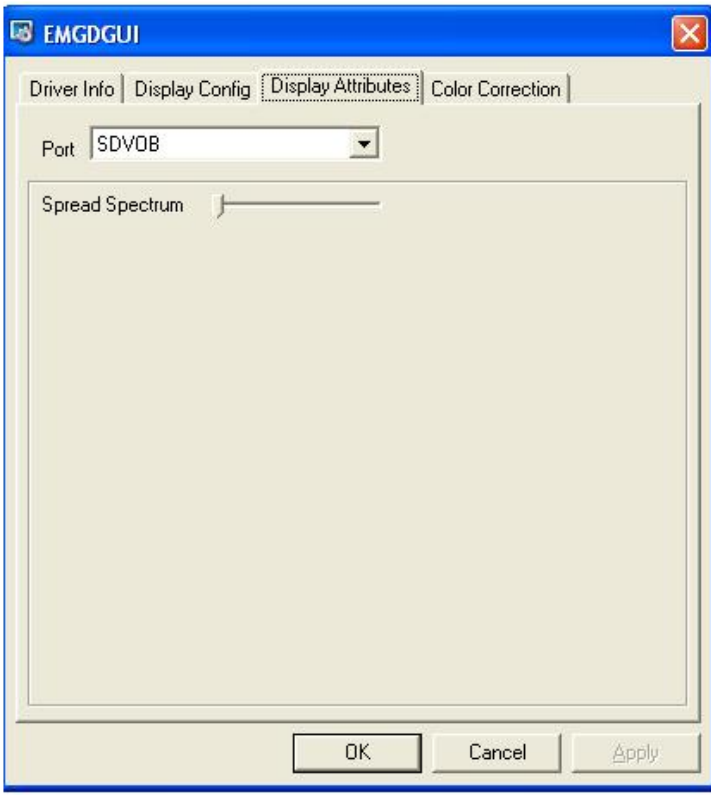
The **Display Status** section of the above dialog shows the current configuration for the **Primary** and **Secondary** displays.

3. The “Swap Primary Display” option allows you to swap the port order for primary and secondary displays.
4. In the **Display Configuration** section of the dialog, select the required display configuration in the **Display Config** drop-down list. This allows the user to choose between Single, Clone, and Extended for all connected ports. A maximum of two ports per display configuration is currently allowed.
5. In the **Primary Mode** and **Secondary Mode** sections of the dialog, change resolution and bit depth of the primary and secondary displays via the **Resolution** and **Bit Depth** drop-down lists.
6. In the **Display Settings** section of the dialog, view and change the settings for a port, rotate and flip the display via the appropriate drop-down lists:
   - **Port**: Allows you to select the required port.
   - **Port Status**: Allows you to enable or disable the selected port. May not be available if there is only one currently active port.
   - **Rotate**: You can rotate the display 0, 90, 180, and 270 degrees.
   - **Flip**: Inverts the display horizontally.

Note: If you change any configuration settings in the **Display Config** dialog box, click **Apply** for the changes to take effect.

7. Click the **Display Attributes** tab to view and change the attributes for a port. The screen that appears depends upon the port drivers used.

Figure 30. Example Runtime Configuration GUI — Display Attributes Tab



The figure above shows the attributes that can be changed for the selected port in the **Port** drop-down list. You can change the Port Driver by selecting the appropriate one for your device. The attributes that appear on this tab depend upon the selected port driver. Please see Appendix B, "Port Driver Attributes," for a complete list of port driver attributes.

8. Click the **Color Correction** tab to view and change color corrections. Figure 31 shows a sample Color Correction tab screen.

   Color Correction is available for both overlays and framebuffers.

Figure 31. Example Runtime Configuration GUI — Color Correction Tab

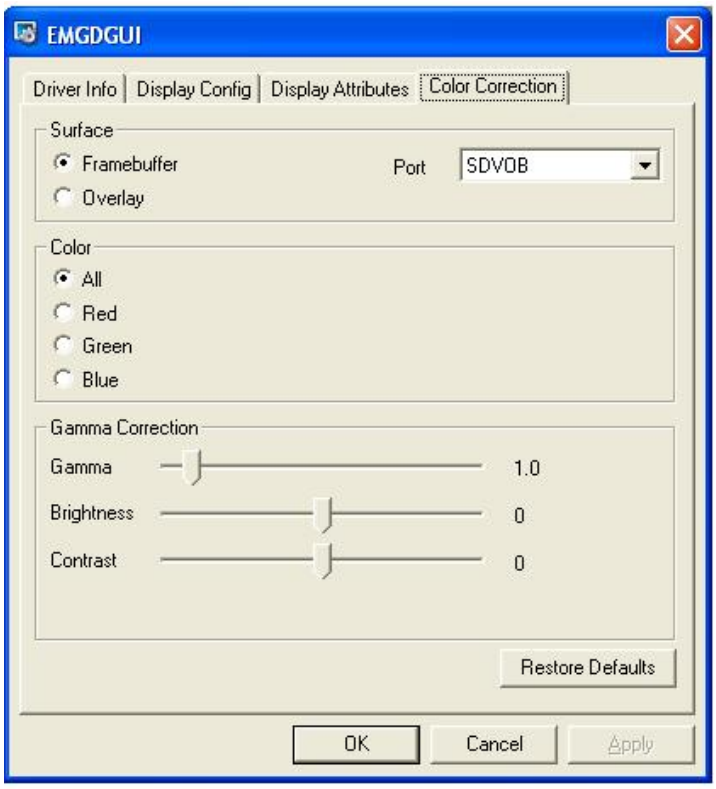
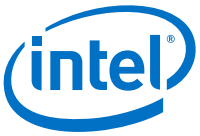


Table 27. Framebuffer Color Correction Values (applies to R, G, B color)

Gamma:	0.6 to 6.0 (default value is 1)
Brightness:	-127 to 127 (default value is 0)
Contrast:	-127 to 127 (default value is 0)

Table 28. Overlay Color Correction Values (applies to ALL color)

Gamma:	0.6 to 6.0 (default value is 1)
Brightness:	0 to 200 (default value is 100)
Contrast:	0 to 200 (default value is 100)
Saturation:	0 to 200 (default value is 100)



The following sub-steps present an example color-correction procedure:

a. Select **Framebuffer** in the **Surface** section and select the appropriate port for the color correction to be applied to or select **Overlay** in the Surface section for color correction to be applied to the overlay.

b. Select the required color to be corrected in the **Color** section.

c. Select the required color attribute to be corrected in the **Gamma Correction** section.

d. Click **Restore Defaults** to restore the default values.

Note: If you make any changes to the color-correction settings, click **Apply** to have the changes take effect.

Note: The hardware does not support brightness, saturation, and contrast of the overlay and second overlay with RGB pixel format.

§ §

6.0 Configuring and Building Intel® EMGD for Microsoft Windows* Embedded Compact 7

6.1 Microsoft Windows* Embedded Compact 7 Installation

The following sections describe how to install Intel® EMGD on the Microsoft Windows* Embedded Compact 7 operating system.

6.1.1 Prerequisites

The development system should have the following software installed:

- Visual Studio* 2008 and Visual Studio Professional Service Pack 1
- Windows* Embedded Compact 7
- Board Support Package (BSP) - version 3.1 (see Section 6.1.2.1, “Installation and Setup” on page 102 for download location)

The target system must contain one of the following Intel chipsets:

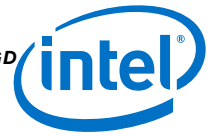
- Intel® Atom™ Processor E6xx

Notes: When using a platform based on the Intel® Atom™ Processor E6xx, for proper driver operation you must:

1. Replace the default VBIOS with the latest EMGD VBIOS.
2. Install the latest Intel® EMGD XP/Linux/Windows Embedded Compact 7 software package as described in Section 6.1.3.2, “Installing Intel® EMGD DirectShow Codecs” on page 105.

6.1.2 Platform Builder Requirements

You must configure your Platform Builder parameters specific to the options that the system and image require, for example, options for the operating system. A Board Support Package (BSP) is also required however, configuration steps for the BSP are beyond the scope of this procedure. An Intel® BSP can be used or the Windows Embedded Compact 7 PC PSP that is included with Platform Builder.



6.1.2.1 Installation and Setup

Note: The installation sequences are crucial for compilation success.

1. Install Visual Studio 2008 Professional.
2. Install Visual Studio 2008 Professional Service Pack 1.
3. Install Windows Embedded Compact 7 platform builder.
   In the SETUP dialog during installation, select **x86** in the processor architecture section.
4. Install Board Support Package (BSP); v3.1 is required. Download the BSP from BSquare at:
   http://www.bsquare.com/software-downloads.aspx

   or Adeneo at:
   http://www.adeneo-embedded.com/Products/Board-Support-Packages/Intel

Note: You need to register before you are able to download.

5. Download Intel® EMGD for XP/Linux/Windows Embedded Compact 7 from the Intel EDC website: edc.intel.com. For assistance on using CED, refer to Section 3.0.
6. Generate a driver for your platform.
7. Unzip and copy the contents of the Driver folder from the zip to `C:\Driver`.
8. Copy and unzip the codecs to a folder on your hard drive. For illustration purposes, assume the folders have been copied to `C:\Driver\Codecs`

Note: See Section 6.1.3.2 for instructions to request the Codecs through QuAD.

9. After installation is complete, change directories:
   cd C:\WinCE700\Platform\Intel_CS
10. Edit `Intel_cs.bat`:
    a. Change BSP_DISPLAY_FLAT= 1
       to
       BSP_DISPLAY_FLAT=
    b. On a new line below `set BSP_DISPLAY_FLAT=`
       add a new line:
       `set BSP_DISPLAY_EMGD=1`
    c. Save and close the file.
11. Change directories:
    cd C:\WinCE700\Platform\Intel_CS\Files
12. Edit `Platform.reg`:
    a. From the `C:\WinCE700\platform\INTEL_CS\FILES` folder, open the file `platform.reg`.
    b. Find the lines `ENDIF BSP_DISPLAY_RAGEXL` and `ENDIF BSP_NODISPLAY !`. Between these two lines, paste the following code:

```
IF BSP_DISPLAY_EMGD
[HKEY_LOCAL_MACHINE\System\GDI\DisplayCandidates]
    "Candidate6"="Drivers\\Display\\Intel"
[$(PCI_BUS_ROOT)\Template\EMGD]
    "DisplayDll"="ddi_emgd.dll"
    "Class"=dword:03
    "SubClass"=dword:00
```

```
    "ProgIF"=dword:00
    "VendorID"=multi_sz:"8086", "8086", "8086", "8086",
"8086", "8086", "8086", "8086", "8086", "8086", "8086",
"8086", "8086", "8086", "8086", "8086", "8086", "8086",
"8086", "8086", "8086", "8086", "8086", "8086", "8086",
"8086", "8086", "8086"
    "DeviceID"=multi_sz:"3582", "2572", "2562", "357B",
"3577", "1132", "7125", "7123", "7121", "2582", "2782",
"2592", "2792", "2772", "2776", "27A2", "27A6", "2982",
"2983", "29A2", "29A3", "2992", "2993", "2972", "2973",
"2A12", "8108", "4108"

; include the path to the emgd.reg file in the release
package
#include "C:\Driver\emgd.reg"
#include "C:\Driver\Codecs\emgd_filters.reg"

ENDIF BSP_DISPLAY_EMGD
```

13. Edit `Platform.bib` and at the end of the file, add the following lines. (Note that indented lines below indicate that lines have wrapped; however, they should be entered into `Platform.bib` as one line.)

```
IF BSP_DISPLAY_EMGD
ddi_emgd.dll  C:\<EMGD driver path>\ddi_emgd.dll NK SH
libegl.dll  C:\<EMGD driver path>\libegl.dll NK SH
isr_emgd.dll  C:\<EMGD driver path>\isr_emgd.dll NK SH
lvds.dll C:\<EMGD driver path>\lvds.dll NK SH
sdvo.dll C:\<EMGD driver path>\sdvo.dll NK SH
libGLESv2.dll  C:\<EMGD driver path>\libGLESv2.dll NK SH
libGLES_CM.dll  C:\<EMGD driver path>\libGLES_CM.dll NK SH
libOpenGL.dll  C:\<EMGD driver path>\libOpenGL.dll NK SH
libOpenVG.dll  C:\<EMGD driver path>\libOpenVG.dll NK SH
aac_dec_filter.dll C:\<EMGD driver
path>\Codecs\aac_dec_filter.dll NK SH
ac3_dec_filter.dll  C:\<EMGD driver
path>\Codecs\ac3_dec_filter.dll NK SH
mp3_dec_filter.dll  C:\<EMGD driver
path>\Codecs\mp3_dec_filter.dll NK SH
h264_dec_filter.dll C:\<EMGD driver path>\Codecs\
   h264_dec_filter.dll NK SH
mpeg2_dec_filter.dll C:\<EMGD driver path>\Codecs\
   mpeg2_dec_filter.dll NK SH
mpeg2_spl_filter.dll C:\<EMGD driver path>\Codecs\
   mpeg2_spl_filter.dll NK SH
mpeg4_dec_filter.dll C:\<EMGD driver path>\Codecs\
   mpeg4_dec_filter.dll NK SH
mpeg4_spl_filter.dll C:\<EMGD driver path>\Codecs\
   mpeg4_spl_filter.dll NK SH
ENDIF BSP_DISPLAY_EMGD
```

where `<EMGD driver path>` is replaced with the actual path for Intel® EMGD, for example, `C:\Driver\....` This tells the BSP where to find the EMGD driver files.

14. Check that all the paths edited in steps 12 and 13 are valid. If not, do a manual search for the file concerned and adjust the path accordingly.
15. Intel® EMGD does not support compositor in Windows Embedded Compact 7, however, the default setting in Platform Builder has compositor enabled. To disable it, after creating a new project in Platform Builder, use the Catalog Items View search function to find and disable the following settings IN ORDER:
    - SYSGEN_VIDEO_PLAYER
    - SYSGEN_PHOTO_VIEWER
    - SYSGEN_COMPOSITION
    - SYSGEN_DSHOW_MPEG2DEMUX
    - SYSGEN_DSHOW_MP4DEMUX
16. Your image is ready to be built in Visual Studio 2008.

6.1.3 Integrating Intel® EMGD DirectX DirectShow Codecs

6.1.3.1 Intel® EMGD DirectShow Codecs Overview

Microsoft's DirectX DirectShow infrastructure provides a standardized interface for middleware audio-video codec software libraries to expose features for accelerating video and audio processing. This infrastructure does not differentiate between hardware and software acceleration but the middleware codec libraries have the choice of employing either method. For the purpose of enabling hardware accelerated video decode on Windows Embedded Compact 7, the Intel® EMGD Windows Embedded Compact 7 DirectShow filters are provided in the form of middleware codec libraries (DLLs) that will interface with the Intel® EMGD Windows Embedded Compact 7 driver to operate.

The Intel® EMGD DirectShow package includes the following Windows Embedded Compact 7 codecs that are DirectShow transform filters in .dll binary form:

- mpeg2_dec_filter.dll
- mpeg2_spl_filter.dll
- mpeg4_dec_filter.dll
- mp3_dec_filter.dll
- mpeg4_spl_filter.dll
- h264_dec_filter.dll
- aac_dec_filter.dll
- ac3_dec_filter.dll

The codecs with “spl” are splitter codecs.

Notes: Intel® EMGD DirectShow codecs are supported only on the Windows Embedded Compact 7 operating system.

Intel® EMGD splitter filters can connect with most source filters but have been verified to connect only with Intel® EMGD transform filters on its downstream pins. The same case is true with respect to Intel® EMGD transform filter connection with upstream splitter filters.

Important: Intel® EMGD audio and video codec filters work only with Intel® EMGD splitter filters. If these codecs are installed properly into the Windows Embedded Compact 7 OS image (via registry changes), the vplayer.exe is able to load and use Intel® EMGD codecs without any help.

6.1.3.2 Installing Intel® EMGD DirectShow Codecs

Prerequisites:

- At least 512 MB RAM for the target system. The hardware video decode performance depends on what other processes are being run on the system.
- The target system must contain the chipset or processor that supports the video engine.
- Include Intel® EMGD Graphics Driver in the Windows Embedded Compact 7 OS image per the appropriate installation instructions in Section 6.1.2.1, “Installation and Setup” on page 102.

For the latest EVALUATION ONLY versions of the Intel® EMGD DirectShow codecs, contact your Intel FAE or open a QuAD case and request the codecs.

After you have the codec package, follow these steps to set up the Intel® EMGD DirectShow codecs:

1. Ensure that the Intel® EMGD DirectShow codecs are included in the Windows Embedded Compact 7 OS image. You do this by including it into either the `platform.bib` or `project.bib` file. See Section 6.1.2.1, step 13 for an example.
2. Ensure that the `emgd_filters.reg` file is included into the image registry. You do this by including it into either the `platform.reg` or `project.reg` file. See Section 6.1.2.1, end of step 12 for an example.
3. Set the backbuffers required for Intel® EMGD Codecs on the Microsoft video renderer filter for smoother performance by changing the following registry key:

   [HKEY_LOCAL_MACHINE\Software\Microsoft\DirectX\DirectShow\Video Renderer]

   “MaxBackBuffers”=dword:X

   where X is the current value that you need to change to equal to or greater than 5.
4. For smoother playback and lower CPU utilization, ensure you use interrupts with Intel® EMGD if available. See Section 6.1.2.1, “Installation and Setup” on page 102 for details.

6.2 Microsoft Windows* Embedded Compact 7 Configuration

The following sections describe how to configure the Intel® EMGD on the Microsoft Windows* Embedded Compact 7 operating system. All the Intel® EMGD-specific registry keys are located within the path
`[HKEY_LOCAL_MACHINE\DRIVERS\Display\Intel]`

All keys use one of the following syntax:

`"<keyname>"=dword:<value>`,

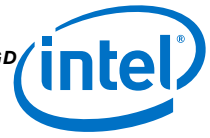
or

`"<keyname>" = <value>`

where <*value*> in the second case is a string in double quotes.

Note: Unless specified otherwise, the “value” field is in hex format.

The `emgd.reg` file contains display configuration registry entries for the Intel® EMGD. A sample `emgd.reg` file is provided along with the driver package. The content of this file may be included through the “#include” directive in `platform.reg` (see Section



6.1.2.1, "Installation and Setup" on page 102), or it may be copied into the proper section in `platform.reg`.

6.2.1 Basic Driver Configuration

This section discusses basic driver configuration keys located in `[HKEY_LOCAL_MACHINE\Drivers\Display\Intel\ALL\1\General]`.

The table below lists the keys in the "`Intel`" folder.

Table 29. [HKLM\DRIVERS\Display\Intel] Registry Keys

Registry Entry	Description	Possible Ranges
PCFversion	Specifies the version of the current configuration file.	400 or 700
ConfigId	This selects the configuration set.	1, 2, 3, 4, or 5
PortDrivers	List of port drivers to be dynamically loaded when the system boots. The dll's must exist in the `C:\Windows` directory. sDVO transmitter port drivers to load when the system boots.	Space separated string enclosed in quotes, where each port driver name is listed in the string. The default string included with the release has all supported port drivers.

6.2.1.1 Graphics Memory Configuration

The Intel Embedded Graphics Suite (IEGS = VBIOS + Graphics driver) provides the ability to dedicate additional memory for graphics functions on the Microsoft Windows* Embedded Compact 7 platform. This is known as *reserved memory*. Firmware selects the amount of reserved memory. The reservation size is passed to the graphics driver through a scratch register available on the GMCH. Reserved memory helps minimize the amount of memory stolen from the OS for memory-limited, embedded systems. For instance, if firmware uses a 640 x 480, 32-bit framebuffer, a total of 1.2 MB is required. Stolen memory would need to be configured as 8 MB or higher, since the next smaller option is only 1 MB, too small for the 640 x 480, 32-bit framebuffer. In such a case, stolen memory can be programmed to 1 MB. Reserved memory can provide the additional memory required for the framebuffer, removing only a minimum amount of memory from the OS.

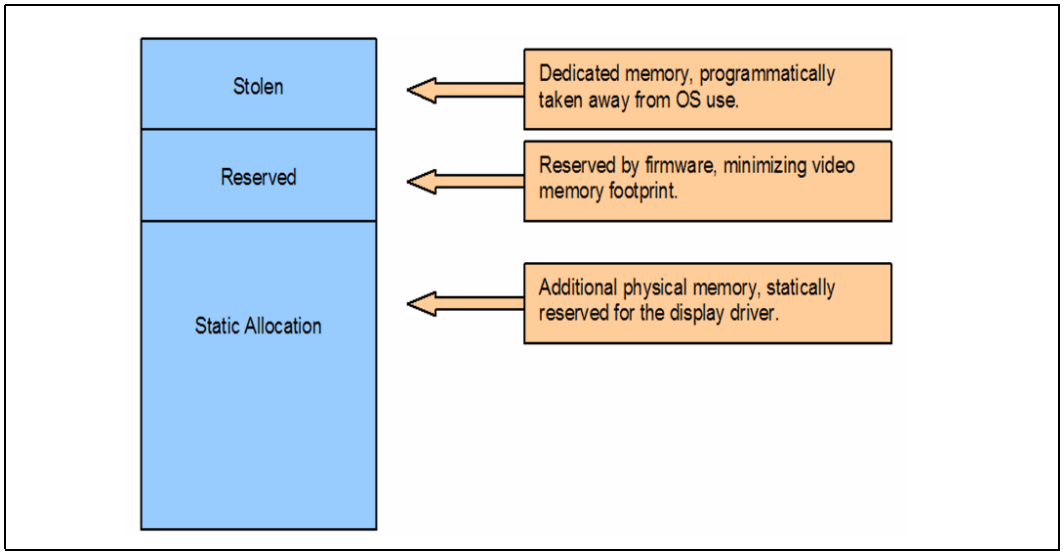
Note: Reserved memory is only available on the Microsoft Windows Embedded Compact 7 operating system, and must be accounted for in the `config.bib` memory layout file.

Additionally, one can configure the Microsoft Windows Embedded Compact 7 display driver for either static or dynamic allocation of video memory. The static model preallocates physical memory for the display driver and provides a more efficient surface allocation scheme. The dynamic model allocates surface memory on demand from the system and will incur a small performance hit. However, the dynamic model has the advantage of deallocation of video memory when not required, thus making it available to other applications.

The static memory model requires a base and size specification registered in the `project.reg` file. The base + size must reach to top of memory (TOM). Since this is not required to be specified in the config.bib memory map, care must be taken not to overlap any other memory arenas with the static allocation. See Section 6.2.1.2, "Defining Graphics Memory Size" on page 107 for further details on how to configure the static memory model.

Figure 32 shows a typical memory map, using a static memory model.

Figure 32. Typical Memory Map Using Static Memory Model



6.2.1.2 Defining Graphics Memory Size

The driver supports the ability to allocate graphics memory dynamically by sharing system resources with the operating system or statically by pre-allocating a block of system memory to be used exclusively by the graphics driver.

To configure the driver to operate using static video memory, two registry settings "ReservedMemoryBase" and "ReservedMemorySize" need to be enabled and defined with valid values. These two registry entries control the start address and size of the memory range pre-allocated for graphics driver use. The pre-allocated memory range should include the stolen memory (BIOS setting). For the Intel chipset or processor, this feature does not reuse the stolen video memory reserved by BIOS. Intel recommends getting BIOS to limit this to the smallest size as this memory is wasted due to some combined OS and hardware limitation.

For example, on a system with 512 MB of system memory and 4 MB of stolen memory (BIOS option), if an additional 14 MB of graphics memory (for a total of 18 MB) is desired, these settings should be used.

```
"ReservedMemoryBase"=dword:1E400000
"ReservedMemorySize"=dword:01C00000
```

These settings indicate that the managed graphics memory pool will begin at physical address 0x1E400000 (484 MB) and will be 18 MB in size. The base address, "ReservedMemoryBase," is the physical system address value and the stolen memory from the BIOS settings is included.

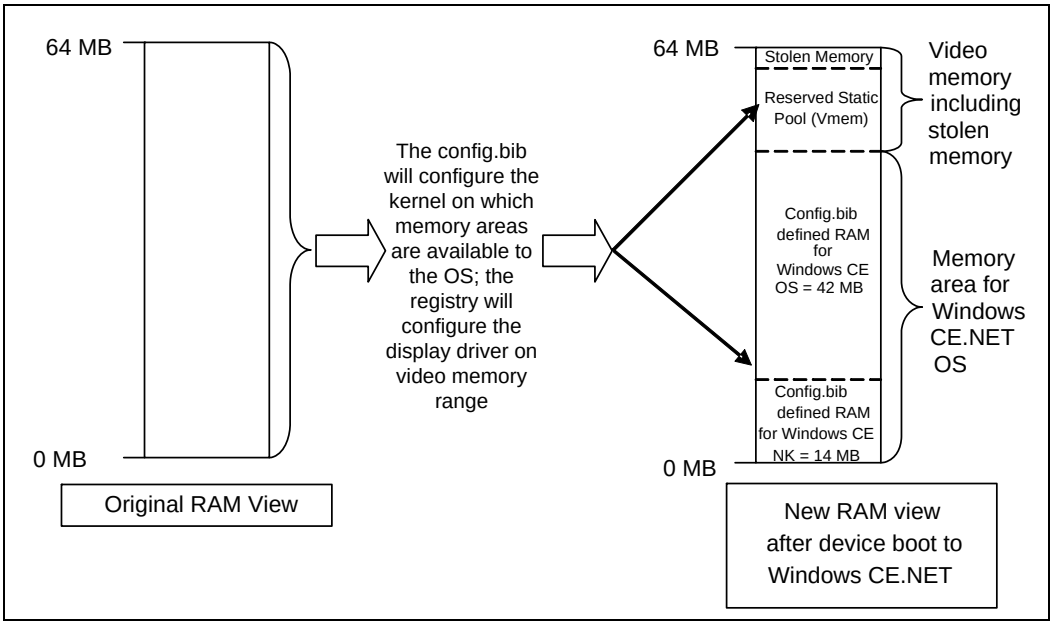
Check the platform you are using to ensure you have all the stolen memory taken into account. For example, in the case of the Cobra board that uses Intel's ACSFL firmware, 2 MB of stolen video memory needs to be included in this configuration. Always remember to include the amount of stolen memory in this number.

Besides the registry entry, the Platform Builder working project also needs to be updated to ensure that the kernel does not try to access this stolen memory. Two items in the config.bib of the project workspace need editing: the NK image/RAM memory partitioning and the memory reservation list. Using the example of the registry configurations above, the kernel would have to be configured not to use the physical memory above the 484 MB mark since that's where the static video memory begins. Thus, the total of the NK image and the system's available RAM must be no more than 484 MB, so you must change your `config.bib` accordingly:

```
NK      80220000 009BE0000  RAMIMAGE ;14 MB for nk.bin + misc.

RAM     80C00000 1DA00000  RAM ;42 MB for RAM
```

The `NK.BIN` image plus the lower conventional memory DMA buffers used by Windows Embedded Compact 7 takes 10 MB; 474 MB is for the RAM. Thus, the memory area above the 484 MB mark is untouched by the kernel and will be used by the display driver.

Overall solution from above example settings in terms of physical system memory viewpoint:



6.2.1.3 Framebuffer and Video Surface Size

Two additional optional registry settings are available to limit the framebuffer size of the display driver and the total size of offscreen video surfaces.

The `MaxFbSize` registry entry will control the maximum size of the framebuffer only. Actual usage will depend on the mode being used.

The `PageReqLimit` registry entry will control the total size in pages (4 KB) of all video surfaces, buffers allocated for any use. Both of these registry configurations apply to both the static as well as dynamic video memory management explained in the previous section. The default below indicates that a maximum of 2 MB are used for the framebuffer and a maximum of 16 MB are permitted for all offscreen videosurface allocations.

```
"MaxFbSize"=dword:200000
"PageReqLimit"=dword:1000
```

In the case of Microsoft Windows Embedded Compact 7, because the OS does not allow for dynamically setting the framebuffer size, the `MaxFbSize` can be changed to match the mode setting being used in order to minimize on video memory waste. The following are different suggested values for `MaxFbSize` for different display modes. These values have not been validated. Note that 640x480 is calculated as 640x512 and 800x600 is calculated as 800x768 for stride alignment purposes.

640x512X16 = A0000
640x512X24 = F0000
640x512X32 = 140000
800x768X16 = 12C000
800x768X24 = 1C2000
800x768X32 = 258000
1024x768X16 = 180000
1024x768X24 = 240000
1024x768X32 = 300000
1280x1024x16 = A000000
1280x1024x32 = A000000

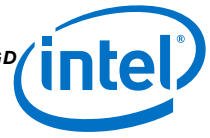
6.2.1.4 Video Surface Allocation Rule

Another two optional registry entries determine a minimum width and height that allow video surface allocations to succeed.

In Windows Embedded Compact 7 GDI, video surface allocations can happen with a REQUIRE_VIDEO _MEMORY or a PREFER_VIDEO_MEMORY flag. The following options will force surface allocations with the PREFER_VIDEO_MEMORY flag to be allocated in system memory if the width and height are lower than stated.

The “MinVidSurfX” registry entry defines the minimum width of a surface allocation for it to succeed with video memory. “MinVidSurfY” defines the minimum height. The surface allocation will succeed if either the width or the height is at the required minimum.

```
"MinVidSurfX"=dword:10
“MinVidSurfY”=dword:10
```



In this example, surfaces allocated with the PREFER_VIDEO_MEMORY where the width and height are both less than 16 pixels are forced to be in system memory.

This option increases performance of the display device as smaller video images, such as icons, would be kept in system memory and only blitted onto the visible frame buffer when they are needed. This ensures optimal use of the display device for larger video surfaces where acceleration makes sense.

6.2.1.5 System to Video Stretch Blit

System to Video Memory stretch blits are not natively supported on Intel GMCH devices. This feature allows you to enable a soft copy of system surfaces to video surfaces to conduct an accelerated stretch blit. The advantage is that the stretch blit uses the blend engine and hardware filtering can be applied. The filtering options are listed in Section 6.2.2.

A value of 1 for the “SysToVidStretch” enables system-to-video stretch blits, as described above, while a value of 0, disables this feature and forwards all system to video stretch blits to the emulator provided by the operating system.

```
[HKEY_LOCAL_MACHINE\Drivers\Display\Intel\ALL\1\General]
"SysToVidStretch"=dword:0
```

6.2.1.6 emgd.reg File Backward Compatibility

Intel® Embedded Media and Graphics Driver expects a configuration file in the PCFVersion 700 format. However, the driver currently supports backward compatibility with version 4.0. This support is not guaranteed, and will be discontinued at a later release. This support is implemented through the `PcfVersion` key as shown below:

```
[HKEY_LOCAL_MACHINE\Drivers\Display\Intel\ALL\1\General]
"PcfVersion"=dword:400
```

Intel® EMGD uses this key to determine the format of the configuration file. When this key is present, Intel® EMGD parses the configuration file using the format specified by the key (400 or 700). If this key is not present, then Intel® EMGD assumes 4.0 format.

6.2.2 Configuration Sets

The Intel® Embedded Graphics Drivers allows multiple configuration sets for OEMs who want to use the same `emgd.reg` file across different platforms. There can be up to 16 instances of configurations. The registry key described in the previous section, ConfigId, ensures the display driver selects the right instance. Each instance may contain multiple groups of per-config and per-config+per-port platform customizations.

The configuration sets are defined in the registry tree as

```
[HKEY_LOCAL_MACHINE\Drivers\Display\Intel\<platform>\<config id>],
```

Where <config id> is the configuration number. The “ConfigID” key described in the previous section selects the active configuration set.

6.2.3 General Configuration

Registry keys described in this section can be found in [`HKEY_LOCAL_MACHINE\Drivers\Display\Intel\<platform>\<config id>\`], where <config id> is the configuration number, and where <*platform*> is Atom E6xx. The driver first attempts to find the configuration or platform on which it is booted, but if the configuration for that platform is not present, the driver uses the ALL platform setting.

Table 30. [HKLM\Drivers\Display\Intel\<platform>\<config id>\]Registry Keys (Sheet 1 of 2)

Registry Entry	Description	Possible Ranges
Width	Width of the display	Width and Height must be expressed as hexadecimal values. For example: 1024 x 768: 400 x 300 800 x 600: 320 x 258 640 x 480: 280 x 1E0
Height	Height of the display	See above.
Depth	Color depth in bpp (bits per pixel)	Depth must be expressed as a hexadecimal number and must be one of the following values: 8bpp: 8 16bpp: 10 24bpp: 18 32bpp: 20 (Note that the Intel 915 chipsets do not support 24 bpp.)
Refresh	The refresh rate of the display.	Refresh rate must be in hex: 60 : 3c 70 : 46 75 : 4b 85 : 55 etc... This value can be any valid refresh rate as long as the display port supports it. A refresh of '0' takes the first refresh that matches `width, height` and `depth`.
NO_D3D	Specify whether to enable D3D.	0 = Enable D3D 1 = Disable D3D Default is 0.
ReservedMemoryBase ReservedMemorySize	Video memory can be statically reserved or dynamically allocated on demand. If both *ReservedMemoryBase* and *ReservedMemorySize* are non-zero, then Video memory allocation uses the static model.	The ReservedMemoryBase plus the ReservedMemorySize must extend to the TOM (Top Of Memory) and not conflict with other reserved memory arenas in config.bib. Default for both base and size is zero, indicating a dynamic allocation model. Default behavior disables static memory model.
MaxFbSize	Maximum size of the expected framebuffer. By providing this hint, the display driver can more efficiently organize GART memory, leading to a smaller video memory consumption.	Must be greater than or equal to the expected size of framebuffer. Units are in bytes. Specifying zero causes the default framebuffer reservation sizing. Default: All other chipsets: 16 MB
MinVidSurfX MinVidSurfY	In pixels, the minimum width and height of surfaces in order to be acceptable for allocation in Video memory. Due to hardware restrictions that optimize memory access, it is advisable to reserve video memory for larger surfaces and allow GDI and DirectDraw* to allocate small surfaces from system memory.	No limitations. Suggested values for both width and height are 10. Default value for both width and height is 1. Default: MinVidSurfX = 1 MinVidSurfY = 1
ReUseStolenMemory	The dynamic memory option allows the user to choose whether they want to use the memory stolen by the BIOS or if they want to scrap that memory and re-allocate memory dynamically.	dword:0 = Disabled dword:1 = Enabled Default: dword:1

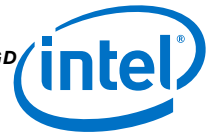


Table 30. [HKLM\Drivers\Display\Intel\<platform>\<config id>\]Registry Keys (Sheet 2 of 2)

Registry Entry	Description	Possible Ranges
SysToVidStretch	Enables system-to-video memory stretch blit operations to take advantage of hardware-accelerated filtering. Normally, it is more efficient to allow GDI to conduct system-to-video stretch blits, but the default filtering used by GDI is Nearest.	0 = Disabled 1 = Enabled Default: 0
BlendFilter	Provides selection of hardware-accelerated filtering methods for stretch blit operations.	0 = Nearest 1 = Bilinear 2 = Anisotropic Default: 2
TearFB	If enabled, all blit operations to the framebuffer are synchronized with video sync to eliminate any visible tearing or flickering on the display screen. Disabling this feature achieves a performance gain.	0 = Disabled, tearing allowed 1 = Enabled, no visible tearing Default: 1
OverlayDualVext	Provides selection for enabling two hardware overlay planes (one for each screen) to display independent video stream on each overlay plane. This selection only applicable in Vertical Extended Mode on Atom E6xx. Note that the hardware overlay plane for each display locks on that screen; the overlay fails to display if it is crossed into the wrong screen.	0 = Disabled 1 = Enabled Default: 0
DisplayConfig	The “DisplayConfig” key sets the display configuration to be in Single, Clone, or Vertical Extended modes. (Unlike Microsoft Windows* XP, Microsoft Windows* Embedded Compact 7 does not support Extended mode). It does not, however, dictate what type of display ports will be used.	1 (single), 2 (clone), 5 (vertical extended)
DisplayDetect	The “DisplayDetect” key allows the user to enable a display port only if a display device is connected. Displays without EDID will not be detected.	0 = disable 1 = enable Default: 0
PortOrder	The PortOrder setting ensures the correct display port types are used based on user selection.	See Section 6.2.3.1.

6.2.3.1 PortOrder Information

PortOrder specifies the actual ports that are used for the Primary and Secondary display. As shown in the table below, the port numbers are slightly different among the supported chipsets.

Table 31. PortOrder Information

Port Number	Chipsets
2	sDVO B Port/RGBA Port
4	Internal LVDS Port

The driver attempts to use the ports in the order specified by “PortOrder”. For example, “PortOrder” = “42000” will assign the internal LVDS port to the primary display and the SDVO port to the secondary display (if any), assuming all the ports are present and detected. Suppose port “4” is not present, in that case the driver tries to assign the next port (2, in this case) in line to the primary display, resulting in SDVO port for primary.

Setting PortOrder to “00000” causes the driver to use default internal settings.

```
*************
[HKEY_LOCAL_MACHINE\Drivers\Display\Intel\ALL\1\General]
;-------------------------------------
; Select Port Order
;-------------------------------------
“PortOrder”=“54320”

;  PortOrder specifies the actual port
;  that will be taken for the Primary /
;  Secondary ports if there are duplicates
;  of the same type. For example, if both
;  Primary and Secondary are digital, then
;  port order will determine which sDVO ports
;  will be first and second. The section below
;  gives the port order numbers for various chipsets.
;  Specify value “0000” to use default settings.
;  On i915 chipsets:
;  ================================
;    1 - Integrated TV Encoder
;    2 - sDVO B port/RGBA port
;    3 - sDVO C port
;    4 - Internal LVDS port
;    5 - Analog port
;
```

6.2.3.2 Vertical Extended Mode

The Windows* Embedded Compact 7 Intel® EMGD driver supports Vertical Extended display mode, which is one large framebuffer that extends across two displays by doubling the height of resolution. The top half of the framebuffer is on the first pipe and the bottom half is on the second pipe. The Windows Embedded Compact 7 operating system is unaware of the two displays. This feature is supported only on the dual-pipelined chipsets, which is every supported platform stated in Section 6.1.1.

This feature is enabled through the `DisplayConfig` key in the `project.reg` file. The resolution, bit depth, and refresh rates of both displays must be the same. Vertical and horizontal panning are *not* supported. DirectDraw is supported on both pipes, but DirectDraw 3D must be disabled when Vertical Extended Display mode is enabled.

6.2.4 Per Port Platform Customization

Intel® EMGD provides what is considered the most useful tools to the embedded market — per port platform customizations. This includes the following:

- Defining custom DTD panel timings
  : `PixelClock, HorzActive, HorzSync` etc...
- Customized GPIO pin selection for I^2C and DDC communication with sDVO encoders and panels.
  :`I2cPin, I2cDab, I2cSpeed` etc...
- Flat Panel width and height limitations and power and/or backlight control mechanisms
  :`BkltMethod, BkltT1, BkltT2, GpioPinVdd` etc...
- Port driver specific attribute settings for initialization at boot time.
  : `Brightness, Contrast, H-Position` etc...

All of the above can be set for each individual port depending on the maximum number of ports the chipset supports. Also, you can have multiple instances of these configurations to allow different settings per configuration.

The usage model for this per-config, per-port platform customizations follows after the same options available in the INF registry settings for the Intel Embedded Graphics Drivers for Microsoft Windows* XP. Please see Figure 6.2.7, “Sample emgd.reg File” on page 117 or to the provided registry sample file in the Intel® EMGD Windows * Embedded Compact 7 driver package for examples. The following sections provide information on these configurations.

6.2.4.1 Per Port Customization — General Port Configuration

This section describes port-specific general configuration options. These options are located under
`[HKEY_LOCAL_MACHINE\Drivers\Display\Intel\ALL\1\Port\1\General]`

- Edid
  This boolean key enables (set to 1) or disables (set to 0) the EdidAvail and EdidNotAvail keys.
- EdidAvail and EdidNotAvail
  These two 16-bit keys control the available timings for the display. If an EDID is successfully read from the display device, then Intel® EMGD uses the EdidAvail flag to determine what timings are available. Otherwise, if an EDID cannot be read, then Intel® EMGD uses the EdidNotAvail key.

Bit #	Value (0 or 1)
0	Disable/Enable driver built-in timings
1	Disable/Enable EDID timings. (Only valid for the EdidAvail flag)
2	Disable/Enable DTD
3-15	Reserved

- CenterOff
  If the selected frame buffer size is smaller than what the Intel® EMGD hardware can support, by default the frame buffer will be centered with a black border around it. To explicitly turn off this feature, the user may set the “CenterOff” key to “1”.

- Rotation and Flip
  Intel® EMGD supports desktop rotation through the “Rotation” key in Single and Clone mode. Rotation is not supported in Vertical Extended Mode.
  The “Rotation” key can be set to one of the four following values.

Degrees	Key Value
0	0 (default)
90	5A
180	B4
270	10E

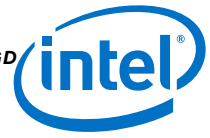
  So, “Rotation”=dword:5A will rotate the frame buffer 90 degrees.
  The “Flip” key flips the desktop horizontally, displaying a mirror image. “Flip” is a boolean value: 1 to enable, 0 to disable.
- Scale
  Intel® EMGD can scale the desktop to the output panel using the panel’s DTD or EDID (in that order). Scaling (attribute ID “18”) is a boolean value, “18”=dword:1 to enable, 0 to disable.

6.2.4.2 Per Port Customization — Custom DTD Timings

For each configuration, each port can be added with up to 255 customized DTD modes.

The following is an example of adding 800x640 mode to the LVDS port when ConfigId=1 is used.

[HKEY_LOCAL_MACHINE\Drivers\Display\Intel\ALL\1\Port\4\DTD\1]
“PixelClock”=dword:9c40
“HorzActive”=dword:320
“HorzSync”=dword:28
“HorzSyncPulse”=dword:80
“HorzBorder”=dword:0
“HorzBlank”=dword:100
“HorzSize”=dword:0
“VertActive”=dword:280
“VertSync”=dword:1
“VertSyncPulse”=dword:4
“VertBorder”=dword:0
“VertBlank”=dword:1c
“VertSize”=dword:0
“Flags”=dword:1e



6.2.4.3 Per Port Customization — Custom Flat Panel Controls

Similarly, the flat panel native resolution and power and backlight sequencing controls can also be configured here.

```
;[HKEY_LOCAL_MACHINE\Drivers\Display\Intel\ALL\1\Port\1\FPInfo]
      ;   "BkltMethod"=dword:0
      ;   "BkltT1"=dword:0
      ;   "BkltT2"=dword:0
      ;   "BkltT3"=dword:0
      ;   "BkltT4"=dword:0
      ;   "BkltT5"=dword:0
      ;   "GpioPinVdd"=dword:0
      ;   "GpioPinVee"=dword:0
      ;   "GpioPinBklt"=dword:0
      ;   "BkltEnable"=dword:0
      ;   "UseGMCHClockPin"=dword:0
      ;   "UseGMCHDataPin"=dword:0
```

Note: For Per-Config, Per-Port configuration, the subkey path includes the correct "Config" and "Port" numbers

6.2.4.4 Per Port Customization — Attribute Initialization

Attributes are also per config and per port. However, the actual keys are dependent on the port driver being used. Below are examples of registry keys associated with initializing attributes for the Chrontel Port Driver.

For complete information on port driver attributes, refer to Appendix B.

Note: For Per-Config, Per-Port configuration, the subkey path includes the correct "Config" and "Port" numbers.

The following example sets the port driver attributes using the attribute IDs. Please see Table 21, "Parameter Configuration Format" on page 58 for a list of attribute IDs and their meanings.

```
[HKEY_LOCAL_MACHINE\Drivers\Display\Intel\ALL\1\Port\1\Attr]
      "0"=dword:32
      "1"=dword:4
      "3"=dword:1
      "8"=dword:1
      "12"=dword:0
      "14"=dword:1
      "19"=dword:1
```

6.2.5 Framebuffer Direct Access

Intel® EMGD provides an option for direct access to the framebuffer during runtime through an escape function call INTEL_ESCAPE_GET_FB_INFO. This feature allows you to directly access the framebuffer for displaying a customized screen or image, such as for showing a customized blue screen.

Refer to the *Intel® Embedded Graphics Drivers, Intel® Embedded Media and Graphics Driver, EFI Video Driver, and Video BIOS API Reference Manual* for a detailed description of the INTEL_ESCAPE_GET_FB_INFO function call.

6.2.6 Miscellaneous Configuration Options

This section covers registry settings not in [HKEY_LOCAL_MACHINE\Drivers\Display\Intel].

6.2.6.1 Text Anti-Aliasing

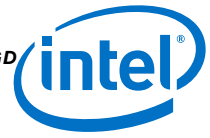
The Microsoft Windows* Embedded Compact 7 driver supports text anti-aliasing. To switch it on, add these registry settings:

```
[HKEY_LOCAL_MACHINE\System\GDI\Fontsmoothing]
[HKEY_LOCAL_MACHINE\System\GDI]
   "ForceGRAY16"=dword:1
```

Note: Always turn on Text Anti-Aliasing when using a TV display device.

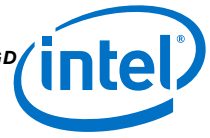
6.2.7 Sample emgd.reg File

```
;-------------------------------------------------------------------------
; Filename: emgd.reg
; $Revision: 1.17 $
;-------------------------------------------------------------------------
; INTEL CONFIDENTIAL
; Copyright (2002-2012) Intel Corporation All Rights Reserved.
; The source code contained or described herein and all documents related
; to the source code (“Material”) are owned by Intel Corporation or its
; suppliers or licensors. Title to the Material remains with Intel
; Corporation or its suppliers and licensors. The Material contains trade
; secrets and proprietary and confidential information of Intel or its
; suppliers and licensors. The Material is protected by worldwide copyright
; and trade secret laws and treaty provisions. No part of the Material may
; be used, copied, reproduced, modified, published, uploaded, posted,
; transmitted, distributed, or disclosed in any way without Intel's prior
; express written permission.
;
; No license under any patent, copyright, trade secret or other
; intellectual property right is granted to or conferred upon you by
; disclosure or delivery of the Materials, either expressly, by
; implication, inducement, estoppel or otherwise. Any license under such
; intellectual property rights must be express and approved by Intel in
; writing.
;
;-------------------------------------------------------------------------

;************** BEGIN INTEL DISPLAY DRIVER REGISTRY ENTRY ***************
;************************************************************************
```



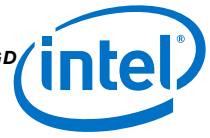
```
[HKEY_LOCAL_MACHINE\Drivers\BuiltIn\PCI\Template\EMGD]
    "Dll"="isr_emgd.dll"
    "Class"=dword:03
    "SubClass"=dword:00
    "ProgIF"=dword:00
    "VendorID"=multi_sz:"8086", "8086"
    "DeviceID"=multi_sz:"8108", "4108"
       ; US15 is the only chipset supporting interrupts
    "Prefix"="IGD"
    "IsrDll"="isr_emgd.dll"
    "IsrHandler"="isr_handler"

[HKEY_LOCAL_MACHINE\System\GDI\Drivers]
    "Display"="ddi_emgd.dll"
;MKM May 2010: IClass for Power MGT - ENABLE FOR TESTING ONLY!
;   "IClass"="{EB91C7C9-8BF6-4a2d-9AB8-69724EED97D1}"
;   "DeviceName"="DISPLAY"
;[HKEY_LOCAL_MACHINE\System\GDI\Drivers]
;    "MainDisplay"="ddi_emgd.dll"
[HKEY_LOCAL_MACHINE\System\GDI\Drivers]
   "D3DMOverride"="ddi_emgd.dll"
[HKEY_LOCAL_MACHINE\System\D3DM\Drivers]
   "RemoteHook"="ddi_emgd.dll"

;*************************************************************************
; The Following Sections Provide
; General Driver-Wide Registry Settings
;*************************************************************************
[HKEY_LOCAL_MACHINE\Drivers\Display\Intel]

;-------------------------------------------------------------------------
; Following registry entry for
; pcf version used
;  400 : EMGD 4.0 version
;-------------------------------------------------------------------------
   "PcfVersion"=dword:700

;-------------------------------------------------------------------------
; This value dictates the configuration to select for Per-Port settings
; from port specific registry. The settings mirror Windows XP EMGD drivers
; implementation Refer to the EMGD User Guide.
;-------------------------------------------------------------------------
   "ConfigId"=dword:1

;-------------------------------------------------------------------------
; Provide a list of port drivers to attempt to load upon boot time
;-------------------------------------------------------------------------
```

```
 #if defined BSP_TUNNEL_CREEK || defined BSP_TOPCLIFF || defined
BSP_PCH_EG20T
    "PortDrivers"="lvds sdvo"
 #else
    "PortDrivers"="analog ch7009 ch7017 fs454 lvds ns2501 ns387 sii164 ti410
th164 sdvo hdmi tv"
 #endif


;-----------------------------------------------------------------------
; Turn on to use stolen memory space. Leave off to re-allocate gart memory
;-----------------------------------------------------------------------
    "ReUseStolenMemory"=dword:0

;***********************************************************************
; The Following Sections Provide Per-Config configuration
;***********************************************************************
[HKEY_LOCAL_MACHINE\Drivers\Display\Intel\ALL\1\General]

;-----------------------------------------------------------------------
; Following registry entries for display settings:resolution, bit depth and
; refresh rate
;
; Width & Height values must be hex, for example
;    1400x1050 : 578h x 41Ah
;    1280x1024 : 500h x 400h
;     1024x768 : 400h x 300h
;      800x600 : 320h x 258h
;      640x480 : 280h x 1E0h
;      etc...
;
; In vertical extended mode, height is doubled
;      640x960 : 280h x 3c0
;      800x600 : 320h x 4b0h
;      etc...;
;-----------------------------------------------------------------------
#if defined BSP_TUNNEL_CREEK || defined BSP_TOPCLIFF || defined
BSP_PCH_EG20T
    ;1366x768 is the highest resolution for the display panel we are using
    "Width"=dword:556
    ; width = 1366
    "Height"=dword:300
     ; height = 768
#else
    "Width"=dword:400
    "Height"=dword:300
#endif


;-----------------------------------------------------------------------
```



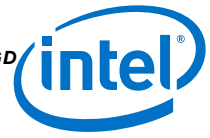
```
; Bit depth must be one of:
;     8bpp  : 8
;     16bpp : 10
;     24bpp : 18
;     32bpp : 20
;(all current EMGD 6.0 & above chipsets do not support 24 bpp)
;-------------------------------------------------------------------------
   "Depth"=dword:20


;-------------------------------------------------------------------------
; Refresh rate must be in hex:
;     60 : 3c
;     70 : 46
;     75 : 4b
;     85 : 55
;     etc...
; any refresh rate as long as the display port supports it refresh of '0'
; will take the first refresh that matches width, height and bpp
;-------------------------------------------------------------------------
   "Refresh"=dword:3c

;-------------------------------------------------------------------------
; Display Mode flags:
;
; Default       : 0
; Interlace     : 80000000
;
; Currently interlace is the only supported display flag
;-------------------------------------------------------------------------
;  "Flags"=dword:0

;-------------------------------------------------------------------------
; Following is registry entry for controlled configuration of video memory
; usage / location
;
; The following settings are for a 64M platform, where the video memory is
; 14M at the top the above settings are assuming there is no system bios /
; firmware that has stolen memory from top of memory. If it does exist
; reduce ReservedMemorySize avoiding overlap exception for ACSFL, memory
; area is reused.
;
; NOTE: CURRENTLY THESE SETTINGS ARE REMARKED FOR DYNAMIC VIDEO MEMORY
;       CONFIGURATION
;-------------------------------------------------------------------------
;   "ReservedMemoryBase"=dword:03200000
;   "ReservedMemorySize"=dword:00E00000


;-------------------------------------------------------------------------
```

```
; Below is Maximum Frame Buffer Size used to limit the maximum size in bytes
; of the main frame buffer
;-------------------------------------------------------------------------
   "MaxFbSize"=dword:800000

;-------------------------------------------------------------------------
; Independent DTDs with mouse restrictions to within screen
;-------------------------------------------------------------------------

    "MouseRestrict"=dword:0

;-------------------------------------------------------------------------
; Page Request Limit is used to control the max allocations of offscreen
; video surfaces, buffers etc.. value is in number of pages (4K).
; this is independent of dynamic or static memory configuration.
;
; The max for 845, 855, 852 = 128MB = 0x8000
; The max for 915s, 945s    = 256MB = 0x10000
;-------------------------------------------------------------------------
   "PageReqLimit"=dword:0

;-------------------------------------------------------------------------
; Above settings are to define a minimum width and height that would allow
; for video surface allocations to succeed, eg: surfaces with width < 16 are
; forced to be in system-mem, surfaces with height < 16 are forced to be in
; system-mem only affects allocations of surfaces with
; GPE_PREFER_VIDEO_MEMORY flag
;-------------------------------------------------------------------------
   "MinVidSurfX"=dword:10
   "MinVidSurfY"=dword:10

;-------------------------------------------------------------------------
; Following are the registry entries for acceleration configuration
;-------------------------------------------------------------------------
; Set SysToVidStrech to '1' enables driver to perform System to Video
; stretch blits
;-------------------------------------------------------------------------
   "SysToVidStretch"=dword:0
;-------------------------------------------------------------------------
; Blend filtering method selection
; The hardware must be capable of
; support, else, emulation is done.
; Possible blend methods are:
;   BlendFilter 0 == NEAREST
;   BlendFilter 1 == BILINEAR
;   BlendFilter 2 == ANISOTROPIC
;   BlendFilter 3 == 4X4
;-------------------------------------------------------------------------
   "BlendFilter"=dword:2
;-------------------------------------------------------------------------
```



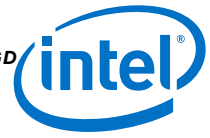
```
; Option for enabling/disabling TEARING - Default is OFF
;-------------------------------------------------------------------------
; Set '1' to enable the NOTEARING option
   "TearFB"=dword:1

;-------------------------------------------------------------------------
; Specify whether to enable d3d
; NO_D3D Value: 0(default)
;             : 0 --> Enable D3D
;             : 1 --> Disable D3D
;-------------------------------------------------------------------------
   "NO_D3D"=dword:0

;-------------------------------------------------------------------------
; Specify the dc (Display Configuration Definition)
;
; The display configuration (dc) is a unique 32-bit identifier that fully
; describes all displays in use and how they are attached to planes and
; pipes to form Single, Clone, Twin and Extended display setups.
;
; The DC is treated as 8 nibbles of information (nibble = 4 bits). Each
; nibble position in the 32bit DC corresponds to a specific role as
; follows:
;
;   0x12345678
;     ||||||||-- Legacy Display Configuration (Single, Twin, Clone, Ext)
;     |||||||--- Port Number for Primary Pipe Master
;     ||||||---- Port Number for Primary Pipe Twin 1
;     |||||----- Port Number for Primary Pipe Twin 2
;     ||||------ Port Number for Primary Pipe Twin 3
;     |||------- Port Number for Secondary Pipe Master
;     ||-------- Port Number for Secondary Pipe Twin 1
;     |--------- Port Number for Secondary Pipe Twin 2
;-------------------------------------------------------------------------
#if defined BSP_TUNNEL_CREEK || defined BSP_TOPCLIFF || defined
BSP_PCH_EG20T
   "dc"=dword:00000041
#else
;  "dc"=dword:00000051
#endif
;-------------------------------------------------------------------------
; Legacy Display configuration, single, twin ...
; When a complex (>2 displays) setup is defined, the legacy configuration
; will describe only a portion of the complete system.
;
; Possible Display Config combo:
;   DisplayConfig 1 == SINGLE
;           (Single is default if none specified)
;   DisplayConfig 4 == TWIN
;      --> (Twin mode: common timing across ports)
```

```
;   DisplayConfig 2 == CLONE
;      --> (Clone mode: distinct timing per port)
;          (845 doesn't support Clone)
;   DisplayConfig 5 == VEXT (vertical extend)
;      --> (Vert Extended modes : "Height"      )
;          (  registry key value must be 2X the )
;          (  intended port timings. Both ports )
;          (  must use the same timings. For    )
;          (  example, for port timings of      )
;          (  800x600, the DisplayConfig should )
;          (  be 5 and the Height=1200 or 0x4b0 )
;          (  Overlay wont work in VEXT mode.   )
;          (845 & 915GV doesn't support Vext)
;-----------------------------------------------------------------------
   "DisplayConfig"=dword:1


;-----------------------------------------------------------------------
; Select Port Order
;  PortOrder specifies the actual port that will be used for the primary and
;  secondary ports. IF specified port is unavailable (port driver failed or
;  display detection failed or port is not available on current chipset),
;  then the next port in the above order will be used. PortOrder must be
;  set, based on chipset specifications:
;  On i915 chipsets:
;  ================================
;    1 - Integrated TV Encoder
;    2 - DVO B port/RGBA port
;    3 - DVO C port
;    4 - Internal LVDS port
;    5 - Analog port
;
;  On i830/835/845/85x/865 chipsets:
;  ================================
;    1 - DVO A port
;    2 - DVO B port/RGBA port
;    3 - DVO C port
;    4 - Internal LVDS port
;    5 - Analog port
;
;    On 835: If RGBA is used (DVO B & C together), then use DVO B number
;    to specify any parameter for it.
;
;  On i81x chipsets:
;  ================
;  Port numbers:
;    3 - DVO port
;    5 - Analog port
;-----------------------------------------------------------------------
```



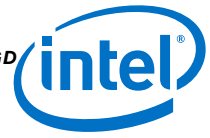
```
#if defined BSP_TUNNEL_CREEK || defined BSP_TOPCLIFF || defined
BSP_PCH_EG20T
    "PortOrder"="42000"
#else
        "PortOrder"="24000"
;   "PortOrder"="52340"
#endif BSP_TUNNEL_CREEK

;-----------------------------------------------------------------------
; Set Clone Port resolutions
;-----------------------------------------------------------------------
;    "CloneWidth"=dword:320
;    "CloneHeight"=dword:258
;    "CloneRefresh"=dword:3c
;   "CloneFlags"=dword:0

;-----------------------------------------------------------------------
; Set "1" to enable Display Detection
; DisplayDetect is to detect display child device before using it
; (panel/tv/etc...).BEWARE, setting this to '1' will mean display for the
; requested port wont be enabled if detection failed. Use this option
; wisely.
;-----------------------------------------------------------------------
  "DisplayDetect"=dword:0


;-----------------------------------------------------------------------
; Set "1" to enable Dual Overlay in Vertical Extended in Windows Embedded
; Compact 7. This is set by the user to enable Dual Hardware Overlays. This
; is a special flag for a specific usage. When two apps request overlays,
; these two will use the two hardware overlays
;-----------------------------------------------------------------------
  "OverlayDualVext"=dword:0

;-----------------------------------------------------------------------
; Overlay Color Correction Settings
;   Gamma:  32-bit integer in 24i.8f format, ranging from 0.6 - 6.0 decimal
;   Brightness: 32-bit integer ranging from 0 to 0xFFFF. 0x8000 = no
;   correction
;   Contrast:   32-bit integer ranging from 0 to 0xFFFF. 0x8000 = no
;   correction
;   Saturation: 32-bit integer ranging from 0 to 0xFFFF. 0x8000 = no
;   correction
;-----------------------------------------------------------------------
;  "OverlayGammaCorrectR"=dword:100
;  "OverlayGammaCorrectG"=dword:100
;  "OverlayGammaCorrectB"=dword:100
;  "OverlayBrightnessCorrect"=dword:8000
;  "OverlayContrastCorrect"=dword:8000
;  "OverlaySaturationCorrect"=dword:8000
```

```
;*************************************************************************
; The sections below are for the more detailed per port
; registry configurations. It follows the same usage model and
; key value meanings as the Windows INF registry configuration
; file. Refer to EMGD User Manual Sections 5.4.1 to 5.4.4
; for details of this.
;*************************************************************************
   ;-------------------------------------------------------------------
   ; Config 1 - DVO-B Port (For Almador)  |
   ;-------------------------------------------------------------------
   ; Following are the registry
   ; entries for port's general config
   ;-------------------------------------------------------------------
   ;
   ;[HKEY_LOCAL_MACHINE\Drivers\Display\Intel\ALL\1\Port\2\General]
   ;
   ;  Advanced Edid Configuration
   ;  ---------------------------
   ;  "Edid"=dword:0
   ;  "EdidAvail"=dword:7    ; STD TIMINGS + EDID TIMINGS + USER TIMINGS
   ;  "EdidNotAvail"=dword:4 ; STD TIMINGS + USER TIMINGS
   ;
   ;      EdidAvail and EdidNotAvail: <only 16 bits used>
   ;      ----------    -------------
   ;      These 2 parameters can be used to control the available timings
   ;      for any display. 'EdidAvail' is used when EDID is read from the
   ;     display device. If an attempt to read EDID is failed or 'Edid = 0'
   ;      then driver uses 'EdidNotAvail' flags.
   ;
   ;      See below bit definitions for both 'EdidAvail' and 'EdidNotAvail'
   ;
   ;       BIT 0:
   ;       ------
   ;         0  - Do not use driver built-in standard timings
   ;         1  - Use driver built-in standard timings
   ;
   ;       BIT1:  <not applicable to EdidNotAvail>
   ;       ------
   ;         0 - Do not use EDID block
   ;         1 - Use EDID block and filter modes
   ;
   ;       BIT2:
   ;       ------
   ;         0 - Do not use user-DTDs
   ;         1 - Use user-DTDs.
   ;
   ;       BIT3-BIT15:
   ;       -----------
   ;           Future use.
```



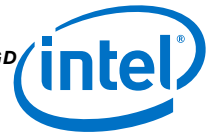
```
;
;       Default behavior:
;       -----------------
;       If user doesn't provide EdidAvail and EdidNotAvail, then
;       EdidAvail     = Use Std timings + Use EDID block and Filter modes
;       EdidNotAvail  = Use Std timings
;
;  Rotation Configuration
;  ----------------------
;  "Rotation"=dword:0
;      Rotation entries must be at a right
;      angle. An invalid entry will be ignored and
;      and no rotation will happen for primary.
;      In clone or twin modes, the secondary
;      port defaults to follow the primary (if set)
;        0   degrees = 0 (not rotated = default)
;        90  degrees = 5A
;        180 degrees = B4
;        270 degrees = 10E
;
;  Flip Configuration
;  ----------------------
;  "Flip"=dword:0
;      Flip has a valid entry of 1 to turn on
;  and 0 to turn off. When turn on the display
;  will be horizontally flip.
;
;  Rendered Scaling Configuration
;  ------------------------------
;  "Scale"=dword:0
;      Scale works as a boolean switch. Valid
;      entries are zero or 1. When "Scale" = 1,
;      EMGD will scale the requested framebuffer
;      resolution to the fixed native panel size
;      indicated by per-port FPInfo, User-DTD or
;      EDID (in that order).
;      In clone or twin modes, the secondary
;      port defaults to follow the primary (if set)

;-------------------------------------------------------------------------
; Following are the registry entries
; for port's DVO I2C settings
;-------------------------------------------------------------------------
;[HKEY_LOCAL_MACHINE\Drivers\Display\Intel\ALL\1\Port\2\DVO]
;  "I2cPin"=dword:2
;  "I2cDab"=dword:70
;  "I2cSpeed"=dword:0
;  "DdcPin"=dword:0
;  "DdcSpeed"=dword:0
;-------------------------------------------------------------------------
```

```
; Following are the registry entries
; for port's flat panel's mode-limits,
; power and backlight control
;-------------------------------------------------------------------------
;[HKEY_LOCAL_MACHINE\Drivers\Display\Intel\ALL\1\Port\2\FPInfo]
;  Only need Width & Height if Panel cannot except other timings
;  "Width"=dword:0
;  "Height"=dword:0
;  "BkltMethod"=dword:3
;  "BkltT1"=dword:1E
;  "BkltT2"=dword:4
;  "BkltT3"=dword:4
;  "BkltT4"=dword:14
;  "BkltT5"=dword:1F4
;  "GpioPinVdd"=dword:27
;  "GpioPinVee"=dword:26
;  "GpioPinBklt"=dword:28
;  "UseGMCHClockPin"=dword:0
;  "UseGMCHDataPin"=dword:0
;-------------------------------------------------------------------------
; Following are the registry entries
; for ports first custom DTD mode to add
;-------------------------------------------------------------------------
;[HKEY_LOCAL_MACHINE\Drivers\Display\Intel\ALL\1\Port\2\DTD\1]
  ;"PixelClock"=dword:6257
  ;"HorzActive"=dword:280
  ;"HorzSync"=dword:8
  ;"HorzSyncPulse"=dword:60
  ;"HorzBlank"=dword:90
  ;"VertActive"=dword:1e0
  ;"VertSync"=dword:2
  ;"VertSyncPulse"=dword:2
  ;"VertBlank"=dword:1d
  ;"Flags"=dword:20000
;-------------------------------------------------------------------------
; Following are the registry entries
; for ports second custom DTD mode to add
; (Up to 255 can be added)
;-------------------------------------------------------------------------
;[HKEY_LOCAL_MACHINE\Drivers\Display\Intel\ALL\1\Port\2\DTD\2]
;  "PixelClock"=dword:9c40
;  "HorzActive"=dword:320
;  "HorzSync"=dword:28
;  "HorzSyncPulse"=dword:80
;  "HorzBorder"=dword:0
;  "HorzBlank"=dword:100
;  "HorzSize"=dword:0
;  "VertActive"=dword:258
;  "VertSync"=dword:1
;  "VertSyncPulse"=dword:4
```



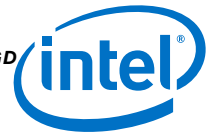
```
;   "VertBorder"=dword:0
;   "VertBlank"=dword:1c
;   "VertSize"=dword:0
;   "Flags"=dword:1e
;-------------------------------------------------------------------------
; Following are the registry
; entries for the port device'
; display attribute parameters
; Use when enabling Port device
; example below is for Conexant
; on Port2 (DVO-B for almador)
; key names depend on port driver
;-------------------------------------------------------------------------
;[HKEY_LOCAL_MACHINE\Drivers\Display\Intel\ALL\1\Port\2\Attr]
;   "Brightness"=dword:32
;   "Contrast"=dword:4
;   "Flicker Filter"=dword:1
;   "Saturation"=dword:4
;   "Hue"=dword:32
;   "Text Filter"=dword:0
;   "Macrovision"=dword:0
;   "Overscan ratio"=dword:1
;   "TV Format"=dword:1
;   "TV Output"=dword:1
;   "Composite and S-Video"=dword:1


;-------------------------------------------------------------------------
; Config 1 - Analog Port (For Any Chipset)
;-------------------------------------------------------------------------
;[HKEY_LOCAL_MACHINE\Drivers\Display\Intel\ALL\1\Port\5\General]
;   "Edid"=dword:1
;   "EdidAvail"=dword:7    ; STD TIMINGS + EDID TIMINGS + USER TIMINGS
;   "EdidNotAvail"=dword:7 ; STD TIMINGS + USER TIMINGS

;[HKEY_LOCAL_MACHINE\Drivers\Display\Intel\ALL\1\Port\5\attr]
;    GAMMA, BRIGHTNESS, CONTRAST
;  "35"=dword:a0a0a0    ; gamma: 3i.5f format for R-G-B, ranging 0.6 to 6
;   "36"=dword:808080     ; brightness:  0 to FF, 0x80 is no correction
;   "37"=dword:808080     ; contrast:    0 to FF, 0x80 is no correction

;[HKEY_LOCAL_MACHINE\Drivers\Display\Intel\ALL\1\Port\5\DTD\1]
;   "PixelClock"=dword:9c40
;   "HorzActive"=dword:320
;   "HorzSync"=dword:28
;   "HorzSyncPulse"=dword:80
;   "HorzBorder"=dword:0
;   "HorzBlank"=dword:100
;   "HorzSize"=dword:0
;   "VertActive"=dword:280
```

```
;  "VertSync"=dword:1
;  "VertSyncPulse"=dword:4
; "VertBorder"=dword:0
;  "VertBlank"=dword:1c
;  "VertSize"=dword:0
;  "Flags"=dword:1e
;[HKEY_LOCAL_MACHINE\Drivers\Display\Intel\ALL\1\Port\5\DTD\2]
;  "PixelClock"=dword:9c40
;  "HorzActive"=dword:320
;  "HorzSync"=dword:28
;  "HorzSyncPulse"=dword:80
;  "HorzBorder"=dword:0
;  "HorzBlank"=dword:100
;  "HorzSize"=dword:0
;  "VertActive"=dword:258
;  "VertSync"=dword:1
;  "VertSyncPulse"=dword:4
;  "VertBorder"=dword:0
;  "VertBlank"=dword:1c
;  "VertSize"=dword:0
;  "Flags"=dword:1e


;-------------------------------------------------------------------------
; Config 1 - Int-LVDS Port (For 855 or 915GM)  |
; Config 1 - Port 4 = Int-LVDS (US15, TUNNEL_CREEK and Mobiles)
;-------------------------------------------------------------------------
 #if defined BSP_TUNNEL_CREEK || defined BSP_TOPCLIFF || defined
BSP_PCH_EG20T
[HKEY_LOCAL_MACHINE\Drivers\Display\Intel\ALL\1\Port\4\General]
 "CenterOff"=dword:1
  "Edid"=dword:0
  "EdidAvail"=dword:0    ; STD TIMINGS + EDID TIMINGS + USER TIMINGS
  "EdidNotAvail"=dword:4 ; STD TIMINGS + USER TIMINGS
  ;"Rotation"=dword:5A
#else
;[HKEY_LOCAL_MACHINE\Drivers\Display\Intel\ALL\1\Port\4\General]
;  "Edid"=dword:1
;  "EdidAvail"=dword:7    ; STD TIMINGS + EDID TIMINGS + USER TIMINGS
;  "EdidNotAvail"=dword:7 ; STD TIMINGS + USER TIMINGS
#endif
```



```
    #if defined BSP_TUNNEL_CREEK || defined BSP_TOPCLIFF || defined
BSP_PCH_EG20T
   [HKEY_LOCAL_MACHINE\Drivers\Display\Intel\ALL\1\Port\4\FPInfo]
    "BkltMethod"=dword:1
     "BkltT1"=dword:3C
     "BkltT2"=dword:C8
     "BkltT3"=dword:C8
     "BkltT4"=dword:32
     "BkltT5"=dword:190
    #else
   ;[HKEY_LOCAL_MACHINE\Drivers\Display\Intel\ALL\1\Port\4\FPInfo]
   ;  Only need Width & Height if Panel cannot except other timings
   ;  "Width"=dword:320
   ;  "Height"=dword:280
   ;  "BkltMethod"=dword:0
   ;  "BkltT1"=dword:0
   ;  "BkltT2"=dword:0
   ;  "BkltT3"=dword:0
   ;  "BkltT4"=dword:0
   ;  "BkltT5"=dword:0
   ;  "GpioPinVdd"=dword:0
   ;  "GpioPinVee"=dword:0
   ;  "GpioPinBklt"=dword:0
   ;  "UseGMCHClockPin"=dword:0
   ;  "UseGMCHDataPin"=dword:0
    #endif
 #if defined BSP_TUNNEL_CREEK || defined BSP_TOPCLIFF || defined
BSP_PCH_EG20T
   [HKEY_LOCAL_MACHINE\Drivers\Display\Intel\ALL\1\Port\4\Attr]
   ;  "27"=dword:0
   ;  Attribute "27" = Dual Channel (boolean)
   ;  "18"=dword:1
   ;  Attribute "18" = Panel Fit Upscale (boolean)
      "26"=dword:12
   ;  Attribute "26" = Panel Depth - 18 or 24
      "60"=dword:1
   ;  Attribute "60" = Fixed timing
      "70"=dword:64
   ;  Attribute "70" = Backlight PWM intensity
      "71"=dword:4F4C
   ;  Attribute "71" = Invertor frequency
#else
   ;[HKEY_LOCAL_MACHINE\Drivers\Display\Intel\ALL\1\Port\4\Attr]
   ;   "27"=dword:1
   ;   Attribute "27" = Dual Channel (boolean)
   ;   "18"=dword:1
   ;   Attribute "18" = Panel Fit Upscale (boolean)
#endif BSP_TUNNEL_CREEK
```

```
 #if defined BSP_TUNNEL_CREEK || defined BSP_TOPCLIFF || defined
BSP_PCH_EG20T
   [HKEY_LOCAL_MACHINE\Drivers\Display\Intel\ALL\1\Port\4\DTD\1]
    "PixelClock"=dword:11A6C
    "HorzActive"=dword:556
    "HorzSync"=dword:30
    "HorzSyncPulse"=dword:20
    "HorzBorder"=dword:0
    "HorzBlank"=dword:A0
    "HorzSize"=dword:0
    "VertActive"=dword:300
    "VertSync"=dword:3
    "VertSyncPulse"=dword:5
    "VertBorder"=dword:0
    "VertBlank"=dword:16
    "VertSize"=dword:0
    "Flags"=dword:20000
#else
   ;[HKEY_LOCAL_MACHINE\Drivers\Display\Intel\ALL\1\Port\4\DTD\1]
   ;  "PixelClock"=dword:9c40
   ;  "HorzActive"=dword:320
   ;  "HorzSync"=dword:28
   ;  "HorzSyncPulse"=dword:80
   ;  "HorzBorder"=dword:0
   ;  "HorzBlank"=dword:100
   ;  "HorzSize"=dword:0
   ;  "VertActive"=dword:280
   ;  "VertSync"=dword:1
   ;  "VertSyncPulse"=dword:4
   ;  "VertBorder"=dword:0
   ;  "VertBlank"=dword:1c
   ;  "VertSize"=dword:0
   ;  "Flags"=dword:1e
 #endif

   ;HKEY_LOCAL_MACHINE\Drivers\Display\Intel\ALL\1\Port\4\DTD\2]
   ;  "PixelClock"=dword:9c40
   ;  "HorzActive"=dword:320
   ;  "HorzSync"=dword:28
   ;  "HorzSyncPulse"=dword:80
   ;  "HorzBorder"=dword:0
   ;  "HorzBlank"=dword:100
   ;  "HorzSize"=dword:0
   ;  "VertActive"=dword:258
   ;  "VertSync"=dword:1
   ;  "VertSyncPulse"=dword:4
   ;  "VertBorder"=dword:0
   ;  "VertBlank"=dword:1c
   ;  "VertSize"=dword:0
   ;  "Flags"=dword:1e
```



```
    ;-------------------------------------------------------------------
    ; Config 1 - SDVO Port-B (For Napa)
    ;-------------------------------------------------------------------
    ;[HKEY_LOCAL_MACHINE\Drivers\Display\Intel\ALL\1\General]
    ;   "name"="EMGD SDVO Configuration File"
    ;[HKEY_LOCAL_MACHINE\Drivers\Display\Intel\ALL\1\Port\2\General]
    ;   "name"="svga"
    ;[HKEY_LOCAL_MACHINE\Drivers\Display\Intel\ALL\1\Port\2\FPInfo]
    ;  For a SDVO driver, sample settings for the panel:1400x1050
    ;  Only need Width & Height if Panel cannot except other timings
    ;   "Width"=dword:578
    ;   "Height"=dword:41A
    ;[HKEY_LOCAL_MACHINE\Drivers\Display\Intel\ALL\1\Port\2\Attr]
    ;   "27"=dword:1
    ;   Attribute "27" = Dual Channel (boolean)

; Optional - Only enable for font anti-aliasing
; Enabling this causes minor performance impact
; Only recommended for TV Output.
;[HKEY_LOCAL_MACHINE\System\GDI\Fontsmoothing]
;
;[HKEY_LOCAL_MACHINE\System\GDI]
;   "ForceGRAY16"=dword:1

;*************** INTEL DISPLAY DRIVER REGISTRY ENTRY END ***************
;***********************************************************************
```

§ §

7.0 Installing and Configuring Linux* OS Drivers

This chapter describes the configuration and installation of the Intel® Embedded Media and Graphics Driver for Linux* systems. The Intel® EMGD supports X servers from the X.Org* organization.

The Intel Linux driver is for use with the integrated graphics of Intel chipsets on the Embedded Intel Architecture roadmap. The driver supports 8-, 16- and 24-bit pixel depths, dual independent head configuration on capable hardware, flat panel, hardware 2D acceleration, hardware cursor, and the XV extension. Stock library files, for example `libva`, can be used with Intel® EMGD. Xinerama is compatible with Intel® EMGD to the extent that it is supported by your Linux distribution although most of EMGD's accelerated video and 3D capabilities will be disabled when running in Xinerama mode.

7.1 Overview

Intel® EMGD includes a kernel-level DRM (Direct Rendering Manager) driver. The Intel® EMGD DRM driver plugs into the kernel's generic DRM framework and is responsible for all hardware access, including display control, memory management, interrupt handling, device initialization, and command queueing.

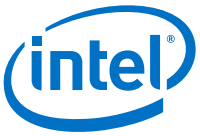
Please refer to "OS and API Support" on page 19 for a list of supported Linux distributions.

Intel® EMGD has been tested with the versions of Xorg that ship with the specified Linux distributions. Interoperability with other versions is not guaranteed.

7.2 Prerequisites

The following lists the prerequisites for installing and configuring Intel® EMGD for Linux*.

- Platform with supported Intel chipset.
- Platform with a minimum of 128 Mbytes RAM.
- Resolution and timing specifications for the display devices that will be configured, either in DTD format or via EDID/Display ID.
- Driver package consisting of directories and files (see the following reduced samples, which are located under the Intel® EMGD `Linux` directory).
  - `Documents/RELNOTES.txt`
  - `License/License.txt`
  - `<Distro>/driver/<xserver-name>/emgd_drv.so` and miscellaneous driver files.
- Linux kernel header package for active running kernel.
  - Direct Rendering support enabled.



- Other system capabilities
  - — DRM packages
- System administration privileges.
- Internet connection.

7.2.1 Supported Hardware

Intel® EMGD supports the following chipsets with integrated graphics:

- Intel® Atom™ Processor E6xx
- Intel® System Controller Hub US15W/US15WP/WPT chipset

7.3 Installation

Intel® EMGD CED runs on Windows to generate a pair of files for Linux that must be copied to the Linux system. Those files are a ".x" containing components of what will become the `xorg.conf` file and a ".tgz" file containing the Intel® EMGD software to be installed and configured on the Linux distribution. The `.x` file can also be generated via CED-Lite in a up-to-date Firefox or Internet Explorer browser. See Linux* OS Configuration Using CED Lite for more information.

Contact your Intel representative for instructions on obtaining the software. You can then install the Intel® EMGD by performing the instructions for your specific distribution in the following sections:

- "Installing Intel® EMGD for MeeGo, IVI Release" on page 134
- "Installing Intel® EMGD for Timesys Fedora Remix 14" on page 136

Note: If you use a Linux distribution different from these, you may need to adapt the instruction steps.

7.3.1 Linux Installer Overview

7.3.1.1 RPM Install

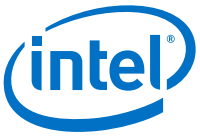
Installation by ROM is supported for both MeeGo and Timesys Fedora Remix 14. The RPM file is located in the respective IEMGD_HEAD_Linux/<Linux Distribution> folder.

7.3.2 Installing Intel® EMGD for MeeGo, IVI Release

1. These instructions assume that you have a fresh install of the MeeGo 1.2 IVI release set up on your system. If that is not the case, you can download the MeeGo 1.2 IVI installer from
   https://meego.com/downloads/releases/1.2/meego-v1.2-in-vehicle-infotainment-ivi
   Installation instructions are available from the site.
2. If the GUI install mode is not working, try using text install mode.

Note: The Intel® EMGD configuration built into the IVI MeeGo 1.2 image defaults to LVDS. If you are using a different main display and you do not have an LVDS connected, you need to prevent MeeGo from booting into its X-based user interface until after you change the `xorg.conf` file to work with your hardware. This can be accomplished by adding a "3" to the boot option in the boot menu when you are presented with the option right at the start of boot. Press "Tab" when prompted, and add a 3 somewhere in the kernel start arguments that will be displayed.

3. After booting, start a terminal unless you are already in console mode. Switch to root by typing
   su
   The default password is **meego**.
4. Install `make` and `gcc` with the command:
   zypper install make gcc
5. Update Kernel and Devel package:
   a. Install the updated MeeGo1.2 Kernel and Development Package for the kernel:
      zypper install kernel-adaptation-intel-automotive-devel kernel-adaptation-intel-automotive
   b. Edit the /boot/extlinux/extlinux.conf and remove the “VGA=current Quiet” from the kernel add-on line. If you are not using an LVDS display (which is the default for the built-in EMGD), also add a 3 to the kernel arguments to prevent the graphics driver from loading and defaulting to LVDS, leaving you with a blank screen. Alternatively, you can also configure the `xorg.conf` for your configuration.
   c. Reboot after installation to enable the updated kernel.
6. Untar the driver package to a convenient location using the following command:
   tar -xvzf <driver package.tgz>
   where <driver package.tgz> is replaced with the actual name of the tar package, which by default is IEMGD_HEAD_Linux.tgz.
7. Change your directory to the MeeGo 1.2 sub-folder in the driver directory and install the Intel® EMGD rpm:
   cd /<your path>/IEMGD_HEAD_Linux/Meego1.2/
   rpm --force -Uvh emgd-*.rpm
8. Remove the old emgd module:
   rm -f /lib/modules/<KernelVersion>/kernel/drivers/staging/emgd/emgd.ko
9. Go to the `common/drm` sub-folder in the driver directory and build the drm module. If you encounter an error concerning an undefined reference to `xen` when building, please do step 9 and come back to retry this step:
   cd /<your path>/IEMGD_HEAD_Linux/common/drm
   tar -xvzf emgd_drm.tgz
   make
   make install
10. If already not done, edit the `/boot/extlinux/extlinux.conf` and remove the “3” and add back “VGA=current Quiet” to the kernel add-on line. After rebooting, this change causes the graphics driver to load with `xorg.conf` configured for your platform.
11. (Optional) This step is necessary only if your compilation is failing in the previous step - otherwise, you may skip to the next step.
    To fix the bug in the MeeGo 1.2 kernel source that contains an undefined reference to the `xen` module:
    a. Edit the file `/usr/src/kernels/<KernelVersion>/arch/x86/include/asm/io.h` to remove the `xen` reference:
       vi /usr/src/kernels/<KernelVersion>/arch/x86/include/asm/io.h
    b. Delete the line referring to `xen` (line 44).
    c. Go back to step 8.



12. By default, MeeGo UX uses Mutter and Mcompositor window manager, which slow down Graphics and Media intensive applications. To change the window manager, follow the installation steps outlined in https://meego.com/sites/all/files/users/admin/xfce_installation.txt

    Note that step 5 in the XFCE install instructions is optional if you want to keep MeeGo UX and not the XFCE4 Desktop.
13. Reboot.
14. Check that you are now using the updated EMGD drivers by opening a terminal and running **emgdgui**.

7.3.3 Installing Intel® EMGD for Timesys Fedora Remix 14

Please note that Intel does not support stock Fedora 14, but has created a custom version of the Fedora 14 distribution, called Timesys Fedora Remix 14, for use with the Atom E6xx processor. It can be obtained from https://linuxlink.timesys.com/intel/linux/. Download the desktop installer and follow the provided instructions from the web site.

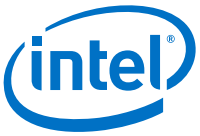
1. These instructions assume that you have a fresh install of the Timesys Fedora Remix 14 release set up on your system. See above for instructions.
2. Log in as root.
3. If yum is having trouble connecting to the Timesys Fedora Remix 14 repositories, you may try the following step:
   Edit `/etc/yum.repos.d/fedora.repo` and `/etc/yum.repos.d/fedora-updates.repo`. Uncomment the first line beginning with `#baseurl` by removing the `#`, and comment out the first line beginning with `mirrorlist` by adding `#` to the beginning of the line.
4. Install the rpm-build package and dependencies:
   yum install rpm-build asciidoc xmlto
5. Disable SELinux:
   a. **vi /etc/selinux/config**
   b. Change
      `SELINUX=enforcing`
      to
      `SELINUX=disabled`
   c. Change `SELinux` to `disabled` and then reboot. The “disabled” state does not become activated until after the reboot.
6. Find your specific kernel version (<kernel-version>) using
   uname -r
7. Download the kernel source for your specific kernel version from the timesys repository:
   http://repository.timesys.com/buildsources/fedora/14/source/SRPMS/

   The file name should be
   `http://repository.timesys.com/buildsources/fedora/14/source/SRPMS/kernel-<kernel-version>.src.rpm`

8. Prepare to install the kernel as follows:
   rpm -Uvh kernel-<kernel-version>.src.rpm
   cd /root/rpmbuild/SPECS
   rpmbuild –bp kernel-<kernel-version>.spec
   cd ../BUILD
   cp -rf linux-<kernel-major-version> /usr/src/kernels/<kernel-version>
   cp /boot/config-<kernel-version> /usr/src/kernels/<kernel-version>/.config

   At the query to overwrite the config, answer **y**.

   cd /usr/src/kernels/<kernel-version>
   make oldconfig prepare scripts
9. Make symbolic links to kernel header:
   cd /lib/modules/<kernel-version>
   ln -sf /usr/src/kernels/<kernel-version>/ build
10. Generate Intel® EMGD for Linux using CED on a Windows system. Bring the compressed tar file and the X configuration file to the Linux system. For example,
    cd /etc/X11
    mkdir emgd
    cd emgd
    and then copy the files to this `/etc/X11/emgd` directory.
11. Untar the driver package to a convenient location using the following command:
    tar -xvzf <driver package.tgz>
    where <driver package.tgz> is replaced with the actual name of the tar package, which by default is `IEMGD_HEAD_Linux.tgz`.
12. If you have installed EMGD modules, please ensure that you have removed existing EMGD modules before installing a newer version.
    cd IEMGD_HEAD_Linux/common/drm
    modprobe emgd && make uninstall
    cd IEMGD_HEAD_Linux/F14
    rpm –e emgd-bin-xxxx-1.1x.i586 –-nodeps
    rpm –e emgd-gui-xxxx-1.1x.i586
13. Change your directory to the F14 sub-folder in the driver directory and install the Intel® EMGD rpm:
    cd /<your path>/IEMGD_HEAD_Linux/F14/
    rpm --force -Uvh emgd-*.rpm
14. Go to the `common/drm` sub-folder in the driver directory and build the drm module:
    cd /<your path>/IEMGD_HEAD_Linux/common/drm
    tar -xvzf emgd_drm.tgz
    make
    make install
15. Copy the `.x` file generated by CED to your `/etc/X11` folder and save it as `xorg.conf`:
    cp /<path>/<filename>.x /etc/X11/xorg.conf
16. Reboot and run `emgdgui` to confirm the driver installation.



7.4 Configuring Linux*

This section describes how to edit the Linux X server configuration file for use with the Intel® EMGD.

7.4.1 Configuration Overview

Intel® EMGD auto-detects all device information necessary to initialize the integrated graphics device in most configurations. However, you can customize the Intel® EMGD configuration for any supported display by editing the X server's configuration file, `xorg.conf`. Please refer to the `Xorg(5x)` man page for general configuration details. This section only covers configuration details specific to the Intel® EMGD.

To configure Intel® EMGD for Linux*, you must edit the X server's configuration file. You can either edit the configuration directly or you can use CED to create configurations that must then be copied into the configuration file. Even if you use CED to create a configuration, you must still edit the Linux configuration file.

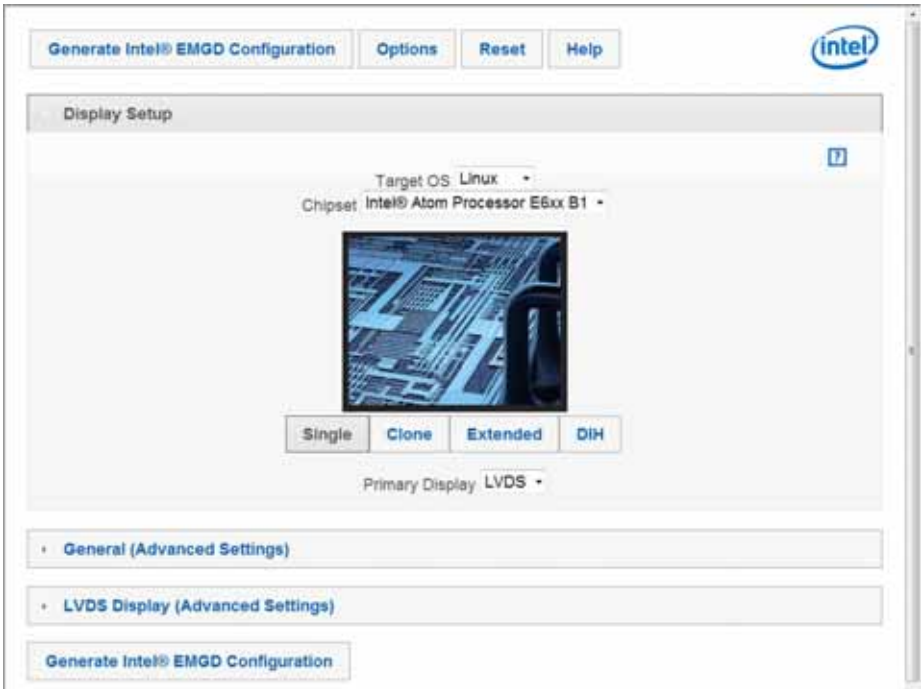
7.4.2 Linux* OS Configuration Using CED

You can configure the Linux* driver settings using CED as described in Section 3.0, "Platform Configuration Using CED" on page 21 or in the CED help.

The output file (`yourbuildnamehere.x`) from CED contains the settings required to configure the Intel® EMGD for Linux systems and can be pasted into the appropriate sections of the `xorg.conf` file.

7.4.3 Linux* OS Configuration Using CED Lite

Intel® EMGD for Linux also comes with a tool that allows users to configure an `xorg.conf` file to suit their needs without having to rebuild the entire EMGD driver with CED. This tool is called CED Lite and is an OS-independent browser-based application that can be found in your `EMGD/Utilities` folder. Unzip the file `CED-Lite.tgz` and open the file `index.html` located inside. CED Lite opens in your browser as shown below. Please note that CED Lite currently supports only the E6xx chipset.



The CED Lite options are identical to those in CED. Click the icon on the CED Lite page or refer to Section 3.0, "Platform Configuration Using CED" on page 21 for help.

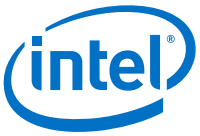
Note that CED Lite generates only an `xorg` file for your convenience in configuring your Linux display settings. It does not rebuild the entire EMGD driver package.

If you experience any problems, please ensure you are using the latest updated version of Internet Explorer* or Firefox*.

7.4.4 Editing the Linux* OS Configuration File Directly

Instead of using CED, you can edit the `xorg.conf` file directly. The following procedure outlines the steps to follow when editing the Linux* configuration file. Section 7.4.5, "The Linux* OS Configuration File" on page 140 provides details on each section of the configuration file.

1. Log in as root and open the configuration file for editing. The configuration file is typically located in the `/etc/X11` directory but may be located elsewhere on your system.
2. In the `Device` section of the configuration file, enter the appropriate information for your driver. The configuration file must have at least one `Device` section. The `Device` section lets you define information about Intel® EMGD. You can use a single `Device` section for single or clone configurations. For Dual Independent Head configurations, you must specify a second `Device` section.
3. In the `Screen` section, enter information for each display in your configuration. The configuration file must have at least one `Screen` section. The `Screen` section binds a `Device` with a `Monitor` and lets you define resolution modes for the display. The Screen section is referenced in the `ServerLayout` section of the configuration file.



4. In the `Monitor` section, define monitor specifications and timings that will be used for the display. You must have a `Monitor` section defined for each display in your configuration. The `Monitor` section is referenced by the `Screen` section.
5. Save your changes to the file. For systems booted to run level 3, **startx** to restart. For systems booted to run level 5, kill X (Ctrl-Alt-Backspace) to restart. Reboot if necessary.

7.4.5 The Linux* OS Configuration File

To configure Intel® EMGD for use with Linux*, you must edit the Linux configuration file (`xorg.conf`). There are several sections within the configuration that must be edited or created, including:

- `Device` Sections
- `Screen` Sections
- `Monitor` Sections
- `ServerLayout` Section (when configuring DIH)
- `ServerFlags` Section (when configuring Xinerama)
- The above Sections are described following the example files. Please see the `xorg.conf` man pages for complete details.

Figure 33. Example xorg.conf File for MeeGo

```
Section "Screen"
       Identifier     "Screen0"
       Device         "IntelEMGD-0"
       Monitor        "Monitor0"
       SubSection     "Display"
       EndSubSection
EndSection

# Primary (First/only) display
Section "Device"
       Identifier "IntelEMGD-0"
       Driver     "emgd"
       VendorName "Intel(R) DEG"
       BoardName  "Embedded Graphics"
       BusID      "0:2:0"
       Screen      0
       Option     "PcfVersion"             "1792"
       Option     "ConfigId"               "1"
       Option     "ALL/1/name"                      "TC_General"
       Option     "ALL/1/General/PortOrder"         "24000"
       Option     "ALL/1/General/DisplayConfig"     "1"
       Option     "ALL/1/General/DisplayDetect"     "1"
       Option     "ALL/1/Port/2/General/name"               "SDVO"
       Option     "ALL/1/Port/2/General/EdidAvail"          "3"
       Option     "ALL/1/Port/2/General/EdidNotAvail"       "1"
       Option     "ALL/1/Port/2/General/Rotation"           "0"
       Option     "ALL/1/Port/2/General/Edid"               "1"
       Option     "ALL/1/Port/4/General/name"               "LVDS"
       Option     "ALL/1/Port/4/General/EdidAvail"          "3"
       Option     "ALL/1/Port/4/General/EdidNotAvail"       "1"
       Option     "ALL/1/Port/4/General/Rotation"           "0"
       Option     "ALL/1/Port/4/General/Edid"               "1"
```

```
EndSection

Section “ServerLayout”
       Identifier     “Default Layout”
       Screen 0       “Screen0” 0 0
       # InputDevice    “Mouse0” “CorePointer”
       # InputDevice    “Keyboard0” “CoreKeybard”
       # InputDevice    “DevInputMice” “SendCoreEvents”
EndSection

Section “InputClass”
       Identifier “evdev keyboard catchall”
       MatchIsKeyboard “on”
       MatchDevicePath “/dev/input/event*”
       Driver “evdev”
EndSection

Section “InputClass”
       Identifier “evdev pointer catchall”
       MatchIsPointer “on”
       MatchDevicePath “/dev/input/event*”
       Driver “evdev”
EndSection

Section “InputClass”
       Identifier “evdev touchscreen penmount”
       MatchIsPointer "on"
       MatchProduct "PenMount USB"
       MatchDevicePath "/dev/input/event*"
       Driver "evdev"
       Option "Calibration" "58 958 116 1020"
EndSection

Section "InputClass"
       Identifier "evtouch touchscreen tsc2007"
       MatchIsTouchscreen “on”
       MatchDevicePath “/dev/input/event*”
       Driver “evtouch”
       Option “MinX” “150”
       Option “MinY” “150”
       Option “MaxX” “3896”
       Option “MaxY” “3696”
       Option “SwapY” “1”
EndSection
```

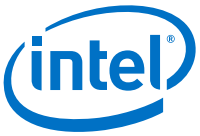
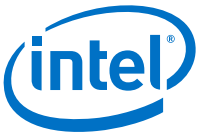


Figure 34. Example xorg.conf File for Timesys Fedora Remix 14

```
##
## X Config options generated from CED
## x11 conf skeleton
## DriverVer=
##

Section "Screen"
       Identifier    "Screen0"
       Device        "IntelEMGD-0"
       Monitor       "Monitor0"
       SubSection    "Display"
       EndSubSection
EndSection

# Primary (First/only) display
Section "Device"
       Identifier "IntelEMGD-0"
       Driver     "emgd"
       VendorName "Intel(R) DEG"
       BoardName  "Embedded Graphics"
       BusID      "0:2:0"
       Screen      0
       Option     "PcfVersion"            "1792"
       Option     "ConfigId"              "1"
       Option     "ALL/1/name"            "TC_General_2023"
       Option     "ALL/1/General/PortOrder"      "24000"
       Option     "ALL/1/General/DisplayConfig"  "1"
       Option     "ALL/1/General/DisplayDetect"  "1"
       Option     "ALL/1/Port/2/General/name"           "SDVO"
       Option     "ALL/1/Port/2/General/EdidAvail"      "3"
       Option     "ALL/1/Port/2/General/EdidNotAvail"   "1"
       Option     "ALL/1/Port/2/General/Rotation"       "0"
       Option     "ALL/1/Port/2/General/Edid"           "1"
       Option     "ALL/1/Port/2/General/CenterOff"      "1"
       Option     "ALL/1/Port/4/General/name"           "LVDS"
       Option     "ALL/1/Port/4/General/EdidAvail"      "3"
       Option     "ALL/1/Port/4/General/EdidNotAvail"   "1"
       Option     "ALL/1/Port/4/General/Rotation"       "0"
       Option     "ALL/1/Port/4/General/Edid"           "1"
       Option     "ALL/1/Port/4/General/CenterOff"      "1"
EndSection

Section "ServerLayout"
       Identifier     "Default Layout"
       Screen 0       "Screen0" 0 0
       # InputDevice    "Mouse0" "CorePointer"
       # InputDevice    "Keyboard0" "CoreKeyboard"
       # InputDevice    "DevInputMice" "SendCoreEvents"
EndSection
```

```
Section "InputDevice"
       Identifier  "Keyboard0"
       Driver      "kbd"
EndSection

Section "InputDevice"
       Identifier  "Mouse0"
       Driver      "mouse"
       Option    "Protocol" "auto"
       Option    "Device" "/dev/input/mice"
       Option    "ZAxisMapping" "4 5 6 7"
EndSection

Section "Monitor"
       Identifier   "Monitor0"
       VendorName   "Monitor Vendor"
       ModelName    "Monitor Model"
EndSection

Section "Device"
       Identifier  "Card0"
       Driver      "emgd"
       VendorName  "Intel Corporation"
       BoardName   "Unknown Board"
       BusID       "PCI:0:2:0"
       Screen   0
       Option   "PcfVersion"   "1792"
       Option   "ConfigId"   "1"
       Option   "ALL/1/name"   "TC_General"
       Option   "ALL/1/General/PortOrder"   "24000"
       Option   "ALL/1/General/DisplayConfig"  "1"
       Option   "ALL/1/General/DisplayDetect"   "1"
       Option   "ALL/1/Port/2/General/name"   "SDVO"
       Option   "ALL/1/Port/2/General/EdidAvail"   "3"
       Option   "ALL/1/Port/2/General/EdidNotAvail"   "1"
       Option   "ALL/1/Port/2/General/Rotation"   "0"
       Option   "ALL/1/Port/2/General/Edid"   "1"
       Option   "ALL/1/Port/4/General/name"   "LVDS"
       Option   "ALL/1/Port/4/General/EdidAvail"   "3"
       Option   "ALL/1/Port/4/General/EdidNotAvail"   "1"
       Option   "ALL/1/Port/4/General/Rotation"   "0"
       Option   "ALL/1/Port/4/General/Edid"   "1"
EndSection

Section "Screen"
       Identifier "Screen0"
       Device     "Card0"
       Monitor    "Monitor0"
       SubSection "Display"
          Viewport   0 0
          Depth     1
       EndSubSection
```



```
        SubSection "Display"
            Viewport    0 0
            Depth     4
        EndSubSection
        SubSection "Display"
            Viewport    0 0
            Depth     8
        EndSubSection
        SubSection "Display"
            Viewport    0 0
            Depth     15
        EndSubSection
        SubSection "Display"
            Viewport    0 0
            Depth     16
        EndSubSection
        SubSection "Display"
            Viewport    0 0
            Depth     24
        EndSubSection
EndSection
```

7.4.5.1 Device Section

The `Device` section provides a description of a graphics device. The Linux* configuration file (`xorg.conf`) must have at least one `Device` section for the graphics driver. If your chipset supports multiple graphics pipelines, you may have multiple `Device` sections, but in most situations, only one is required. If you are creating a Dual Independent Head (DIH) configuration, you must have at least two `Device` sections.

`Device` sections in `xorg.conf` have the following format:

```
Section "Device"
        Identifier "devname"
        Driver "emgd"
        ...
EndSection
```

The `Identifier` field defines the device. This name associates the device with a screen in the `Screen` sections.

`The Driver` field defines the driver to use and is a required field in the `Device` section. The Intel driver, `emgd_drv.o`, must be installed in the `/usr/lib/xorg/modules/drivers` (or the correct path for your system).

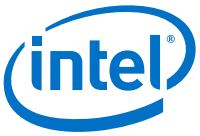
The remainder of the `Device` section can contain Intel® EMGD-specific options. Please see Table 32 on page 148 for a list and description of Intel® EMGD supported options.

DTD IDs for Multiple Ports

While DTD IDs must be unique, if two ports use the same DTD, CED writes to the configuration file twice, once for each port, each with the same ID. **This configuration is correct and should not be changed if you manually edit the configuration file. In most cases you should use CED to configure your system.**

For example, in the Device Section shown below, you see in the first set of option lines in blue that port 2 uses DTD 1 and in the second set of option lines in blue that port 4 also uses DTD 1. The configuration text is correct as written by CED and should not be changed. This situation applies only to Linux configurations.

```
Section "Device"
            Identifier "Intel_Card0"
            Driver     "emgd"
            VendorName "Intel(R) DEG"
            BoardName  "Embedded Graphics"
            BusID      "0:2:0"
            Screen      0
            Option     "PcfVersion"             "1792"
            Option     "ConfigId"               "1"
            Option     "ALL/1/name"                      "dtd_test"
            Option     "ALL/1/General/PortOrder"         "24000"
            Option     "ALL/1/General/DisplayConfig"     "1"
            Option     "ALL/1/General/DisplayDetect"     "0"
            Option     "ALL/1/Port/2/General/name"               "sdvo-b"
            Option     "ALL/1/Port/2/General/EdidAvail"          "7"
            Option     "ALL/1/Port/2/General/EdidNotAvail"       "5"
            Option     "ALL/1/Port/2/General/Rotation"           "0"
            Option     "ALL/1/Port/2/General/Edid"               "1"
            Option     "ALL/1/Port/2/Dtd/1/PixelClock"           "108000"
            Option     "ALL/1/Port/2/Dtd/1/HorzActive"           "1280"
            Option     "ALL/1/Port/2/Dtd/1/HorzSync"             "48"
            Option     "ALL/1/Port/2/Dtd/1/HorzSyncPulse"        "112"
            Option     "ALL/1/Port/2/Dtd/1/HorzBlank"            "408"
            Option     "ALL/1/Port/2/Dtd/1/VertActive"           "1024"
            Option     "ALL/1/Port/2/Dtd/1/VertSync"             "1"
            Option     "ALL/1/Port/2/Dtd/1/VertSyncPulse"        "3"
            Option     "ALL/1/Port/2/Dtd/1/VertBlank"            "42"
            Option     "ALL/1/Port/2/Dtd/1/Flags"                "0xc020000"
            Option     "ALL/1/Port/2/Dtd/2/PixelClock"           "25175"
            Option     "ALL/1/Port/2/Dtd/2/HorzActive"           "640"
            Option     "ALL/1/Port/2/Dtd/2/HorzSync"             "8"
            Option     "ALL/1/Port/2/Dtd/2/HorzSyncPulse"        "96"
            Option     "ALL/1/Port/2/Dtd/2/HorzBlank"            "144"
            Option     "ALL/1/Port/2/Dtd/2/VertActive"           "480"
            Option     "ALL/1/Port/2/Dtd/2/VertSync"             "2"
            Option     "ALL/1/Port/2/Dtd/2/VertSyncPulse"        "2"
            Option     "ALL/1/Port/2/Dtd/2/VertBlank"            "29"
            Option     "ALL/1/Port/2/Dtd/2/Flags"                "0x0"
            Option     "ALL/1/Port/4/General/name"               "lvds"
            Option     "ALL/1/Port/4/General/EdidAvail"          "0"
            Option     "ALL/1/Port/4/General/EdidNotAvail"       "5"
            Option     "ALL/1/Port/4/General/Rotation"           "0"
            Option     "ALL/1/Port/4/General/Edid"               "0"
            Option     "ALL/1/Port/4/Dtd/3/PixelClock"           "65000"
            Option     "ALL/1/Port/4/Dtd/3/HorzActive"           "1024"
            Option     "ALL/1/Port/4/Dtd/3/HorzSync"             "24"
            Option     "ALL/1/Port/4/Dtd/3/HorzSyncPulse"        "136"
            Option     "ALL/1/Port/4/Dtd/3/HorzBlank"            "320"
            Option     "ALL/1/Port/4/Dtd/3/VertActive"           "768"
            Option     "ALL/1/Port/4/Dtd/3/VertSync"             "3"
            Option     "ALL/1/Port/4/Dtd/3/VertSyncPulse"        "6"
```



```
            Option      "ALL/1/Port/4/Dtd/3/VertBlank"          "38"
            Option      "ALL/1/Port/4/Dtd/3/Flags"              "0x20000"
            Option      "ALL/1/Port/4/Dtd/1/PixelClock"         "108000"
            Option      "ALL/1/Port/4/Dtd/1/HorzActive"         "1280"
            Option      "ALL/1/Port/4/Dtd/1/HorzSync"           "48"
            Option      "ALL/1/Port/4/Dtd/1/HorzSyncPulse"      "112"
            Option      "ALL/1/Port/4/Dtd/1/HorzBlank"          "408"
            Option      "ALL/1/Port/4/Dtd/1/VertActive"         "1024"
            Option      "ALL/1/Port/4/Dtd/1/VertSync"           "1"
            Option      "ALL/1/Port/4/Dtd/1/VertSyncPulse"      "3"
            Option      "ALL/1/Port/4/Dtd/1/VertBlank"          "42"
            Option      "ALL/1/Port/4/Dtd/1/Flags"              "0xc000000"
            Option      "ALL/1/Port/4/Dtd/4/PixelClock"         "81230"
            Option      "ALL/1/Port/4/Dtd/4/HorzActive"         "1280"
            Option      "ALL/1/Port/4/Dtd/4/HorzSync"           "48"
            Option      "ALL/1/Port/4/Dtd/4/HorzSyncPulse"      "112"
            Option      "ALL/1/Port/4/Dtd/4/HorzBlank"          "408"
            Option      "ALL/1/Port/4/Dtd/4/VertActive"         "768"
            Option      "ALL/1/Port/4/Dtd/4/VertSync"           "3"
            Option      "ALL/1/Port/4/Dtd/4/VertSyncPulse"      "6"
            Option      "ALL/1/Port/4/Dtd/4/VertBlank"          "34"
            Option      "ALL/1/Port/4/Dtd/4/Flags"              "0x4000000"
            Option      "PortDrivers"                           "sdvo lvds"
EndSection
```

7.4.5.2 Screen Section

The `Screen` section binds a `Screen` with a `Device` and a `Monitor`. It defines resolution modes, color depths, and various other screen characteristics. Please see the `xorg` man page for detailed information.

The `Screen` section has the following format:

```
Section "Screen"
       Identifier  "screenname"
       Device  "devname"
       Monitor  "Monitor0"
       DefaultDepth   24
       Subsection "Display"
          Depth   24
          Modes   "1280x1024" "1024x768" "800x600" "640x480"
       EndSubSection
EndSection
```

7.4.5.3 Monitor Section

Use the Monitor section to define monitor characteristics and timings for a display. You should have one Monitor section for each display your system supports. The Monitor section is referenced in a Screen section and has the following format.

```
Section "Monitor"
      Identifier "Monitor0"
      VendorName "NEC"
      MonitorName "NEC MultiSync LCD"
      HorizSync  30-60
      VertRefresh  50-75
      ...
EndSection
```

7.4.5.4 ServerLayout Section

The ServerLayout section defines the overall layout of the system configuration. Input devices are specified in the InputDevice fields and output devices usually consist of multiple components, such as a graphics board and a monitor, which are bound together in a Screen section. Typically, edit this section only when you are using a DIH configuration. Add a line to reference the second Screen section and specify its relative location to the first screen. In the following sample, the line beginning with Screen 1... is required for DIH configurations.

```
Section "ServerLayout"
      Identifier "Default Layout"
      Screen 0 "Screen0" 0 0
      Screen 1 "Screen1" RightOf "Screen0"
      InputDevice entries...
EndSection
```

7.4.5.5 ServerFlags Section

If you are configuring the Intel® EMGD for Xinerama support, you must set the "Xinerama" option to "True" in the ServerFlags section of the configuration file.

```
Section "ServerFlags"
      Option "Xinerama" "True"
EndSection
```

Note: Timesys Fedora Remix 14 is the only distribution supported by Intel® EMGD that supports Xinerama.

7.4.6 Xorg* Configuration Options

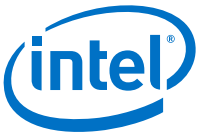
Intel® EMGD provides a format syntax for Linux* configuration options. The syntax is similar to the Microsoft Windows* INF file and is as follows:

```
"All/<ConfigID>/<block name>/<option name>"
```

Intel® EMGD parses the configuration options and looks for "new-style" 4.0 and later options. If it does not find any, then it falls back to processing old-style options.

Device configuration must contain the "pcfversion" option with value "0x700". This indicates to the driver the options format to use. Earlier pcfversions (0 and 0x400) are supported for backward compatibility.

The Intel® EMGD supports multiple sets of installed configuration options that may be selected at runtime.



Configuration ID 0 is used unless otherwise specified in the configuration file or supplied by the system BIOS.

The table below shows the supported driver options.

Table 32. Supported Driver Options (Sheet 1 of 3)

Option	Description
Option “PcfVersion” “integer”	This option indicates that the new Intel® EMGD format is being used for the Linux* Configuration files (`xorg.conf`). The new format is hierarchical (similar to the Microsoft Windows* INF file) and allows both global and per-configuration information to be stored in the X server’s configuration file (`xorg.conf`) rather than having per-configuration information stored separately in the EDIDx.bin file. This option is usually set to 0700 hex (1792 decimal) and is required for the new format.
Option “All/<ConfigID>/General/SWCursor” “boolean”	Enable the use of the software cursor. Default is off and the hardware cursor is used.
Option “All/<ConfigID>/General/ShadowFB” “boolean”	Enable or disable double buffering on the framebuffer. Default disables double buffering.
Option “All/<ConfigID>/General/XVideo” “boolean”	Disable or enable XVideo support. In a dual independent head configuration, either the first display or the second display support XVideo. Both displays cannot support XVideo simultaneously. Default enables XVideo support.
Option “All/<ConfigID>/General/XVideoBlend” “boolean”	Disable or enable XVideo support using the 3D blend manager. This provides XVideo support in configurations that cannot be supported with overlay. For example, this is supported on both displays in a dual independent head setup. It is also supported when the display is rotated or flipped. Color key is only supported if ShadowFB is enabled and the VideoKey is defined. By default XVideoBlend support is enabled.
Option “ALL/<ConfigID>/General/PuntTo3dBlit” “boolean”	Enable or disable use of the 3D engine for blitting. Useful for cases where 2D memory is insufficient. By default PuntTo3dBlit support is disabled.
Option “ConfigID” “integer”	This option identifies the configuration.
Option “All/<ConfigID>/Name” “string”	A quoted string used to identify the configuration name.
Option “All/<ConfigID>/Comment” “string”	A quoted string used to identify the configuration file. Comment is a required field for Linux* configurations.
Option “All/<ConfigID>/General/PortOrder” “string”	This option changes the default port allocation order. The default order can vary depending on chipset. List the port type numbers in the priority order starting from first to last. The port type numbers are as follows: 1 - Integrated TV Encoder (mobile chipsets only) 2 - sDVO B port 3 - sDVO C port 4 - Integrated LVDS port (mobile chipsets only) 5 - Analog CRT port To set the order as Integrated TV Encoder, ANALOG, LVDS, sDVO C, sDVO B set the PortOrder string to “15432”. Zeros can be used to specify don’t care. Setting this option incorrectly can result in port allocation failures.
Option “All/<ConfigID>/Port/<port number>/ General/Rotation” “integer”	Rotate the display. Valid values are 0, 90, 180, 270.
Option “All/<ConfigID>/Port/<port number>/ General/Flip” “boolean”	Invert the display horizontally.

Table 32. Supported Driver Options (Sheet 2 of 3)

Option	Description
Option “All/<ConfigID>/General/VideoKey” “integer”	This sets the color key for XVideo and XVideoBlend. This value is either a 24-bit value or a 16-bit value, depending on the pixel depth of the screen. The color key is always enabled for XVideo, even when it is not defined. The color key is always disabled for XVideoBlend unless both this option is defined and the ShadowFB option is enabled. The default color key for XVideo is 0x0000ff00. For XVideoBlend, the color key is disabled by default.
Option “All/<ConfigID>/General/CloneWidth” “integer”	This sets the display width for a clone port when CloneDisplay is active. Default is 640.
Option “All/<ConfigID>/General/CloneHeight” “integer”	This sets the display height for a clone port when CloneDisplay is active. Default is 480.
Option “All/<ConfigID>/General/CloneRefresh” “integer”	This sets the display vertical refresh rate for a clone port when CloneDisplay is active. Default is 60 Hz.
Option “All/<ConfigID>/Port/<port number>/ General/EDID” “boolean”	Enable or disable reading of EDID data from the output port device. Note that if the EDID option is specified in the config file (`xorg.conf`), all per-port EDID options in the configuration are overwritten by the EDID option specified in the config file.
Option “All/<ConfigID>/General/Accel” “boolean”	Enable 2D acceleration. Default is enabled.
Option “All/<ConfigID>/General/DRI2” “boolean”	Enable DRI2 support for OGL. Default is enabled.
Option “All/<configID>/General/ DihCloneEnable” “boolean”	Enable run-time VEXT or DIH to Clone display mode transition. Default is disabled.
Option “All/<configID>/General/FreezeFB” “boolean”	Enable freeze frame buffer feature. See Section 7.6
Option “All/<ConfigID>/General/ OverlayGammaCorrectR” “integer”	Gamma correction value for overlay (red) in 24i8f format.
Option “All/<ConfigID>/General/ OverlayGammaCorrectG” “integer”	Gamma correction value for overlay (green) in 24i8f format.
Option “All/<ConfigID>/General/ OverlayGammaCorrectB” “integer”	Gamma correction value for overlay (blue) in 24i8f format.
Option “All/<ConfigID>/General/ OverlayBrightnessCorrect” “integer”	Overlay brightness adjustments.
Option “All/<ConfigID>/General/ OverlayContrastCorrect” “integer”	Overlay contrast adjustments.
Option “All/<ConfigID>/General/ OverlaySaturationCorrect” “integer”	Overlay saturation adjustments.
Option “All/<ConfigID>/General/GlTextureStream” “integer”	Enable or disable GL texture streaming support. Default value is 0, disabled. Valid values are 0 or 1.
Option “All/<ConfigID>/Port/<port number>/ General/Name” “string”	A quoted string used to identify the port name, for example, “sdvo”.
Option “All/<ConfigID>/Port/<port number>/ General/”EdidAvail “string”	Specifies how standard and user-defined modes are used when EDID is available. Default is 0.
Option “All/<ConfigID>/Port/<port number>/ General/”EdidNotAvail “string”	Specifies how standard and user-defined modes are used when EDID is not available. Default is 0.
Option “All/<ConfigID>/Port/<port number>/ General/CenterOff” “boolean”	When this option is enabled it DISABLES centering. Also, depending on the combination of “edid” + “user-dtd” + connected hardware, Intel® EMGD will add missing compatibility modes (6x4, 8x6, 10x7& 12x10) via centering. Use this option to disable this feature.
Option “ALL/<ConfigID>/General/RefFreq” “integer”	Internal timing frequency for the sDVO Clip software workaround algorithm. Default is 199500 in KHz. Range 190000KHz -210000KHz.
Option “All/<ConfigID>/Port/<port number>/Dvo/ I2cDab” “string”	I2c device address.

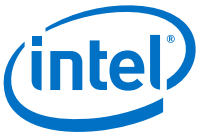


Table 32. Supported Driver Options (Sheet 3 of 3)

Option	Description
Option “All/<ConfigID>/Port/<port number>/Dvo/I2cSpeed” “string”	I2c bus speed.
Option “All/<ConfigID>/Port/<port number>/Dvo/DdcSpeed” “string”	DDC bus speed.
Option “All/<ConfigID>/Port/<port number>/Dvo/DdcDab” “string”	DDC device address.
Option “All/<ConfigID>/Port/<port number>/Dtd/PixelClock” “integer”	Pixel clock frequency in kHz.
Option “All/<ConfigID>/Port/<port number>/Dtd/HorzActive” “integer”	The active horizontal area in pixels.
Option “All/<ConfigID>/Port/<port number>/Dtd/HorzSync” “integer”	Starting pixel for horizontal sync pulse.
Option “All/<ConfigID>/Port/<port number>/Dtd/HorzSyncPulse” “integer”	Width of the horizontal sync pulse pixels.
Option “All/<ConfigID>/Port/<port number>/Dtd/HorzBlank” “integer”	Width of the horizontal blanking period in pixels.
Option “All/<ConfigID>/Port/<port number>/Dtd/VertActive” “integer”	The active vertical area in pixels.
Option “All/<ConfigID>/Port/<port number>/Dtd/VertSync” “integer”	Starting pixel for vertical sync pulse.
Option “All/<ConfigID>/Port/<port number>/Dtd/VertSyncPulse” “integer”	Width of the vertical sync pulse in pixels.
Option “All/<ConfigID>/Port/<port number>/Dtd/VertBlank” “integer”	Width of the vertical blanking period in pixels.
Option “All/<ConfigID>/Port/<port number>/Dtd/Flags” “integer”	Additional interlaced timing information.
Option “All/<ConfigID>/Port/<port number>/FpInfo/BkltMethod” “integer”	Specifies the backlight method.
Option “All/<ConfigID>/Port/<port number>/FpInfo/BkltT1” “integer”	Specifies backlight timing T1.
Option “All/<ConfigID>/Port/<port number>/FpInfo/BkltT2” “integer”	Specifies backlight timing T2.
Option “All/<ConfigID>/Port/<port number>/FpInfo/BkltT3” “integer”	Specifies backlight timing T3.
Option “All/<ConfigID>/Port/<port number>/FpInfo/BkltT4” “integer”	Specifies backlight timing T4.
Option “All/<ConfigID>/Port/<port number>/FpInfo/BkltT5” “integer”	Specifies backlight timing T5.

7.4.7 Sample Dual Independent Head (DIH) Configuration

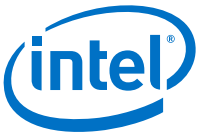
For DIH operation, several additional options must be set in the `Device` sections for each head. Both `Device` sections must specify the `BusID`, and the `BusID` must be the same for both devices. Each `Device` section must specify the `Screen` section that will associate the device with the `Screen` option.

```
BusID  - B:F:S (Bus, Function, Slot)
Screen  - number
```

The example below shows a sample DIH configuration. Only the `Device`, `Screen`, and `Server Layout` sections of the configuration file are shown.

Figure 35. Sample DIH Configuration

```
Section "Device"
      Identifier "IntelEGD-1"
      Driver     "emgd"
      BusID       "0:2:0"
      Screen      0
      VideoRam    32768
EndSection
Section "Device"
      Identifier  "IntelEGD-2"
      Driver      "emgd"
      BusID       "0:2:0"
      Screen      1
      VideoRam    32768
EndSection
Section "Screen"
      Identifier  "Screen 1"
      Device      "IntelEGD-1"
      Monitor     "Monitor1"
      DefaultDepth 24
      Subsection "Display"
         Depth       8
         Modes       "1280x1024" "1024x768" "800x600" "640x480"
         ViewPort    0 0
      EndSubsection
      Subsection "Display"
         Depth       16
         Modes       "1280x1024" "1024x768" "800x600" "640x480"
         ViewPort    0 0
      EndSubsection
      Subsection "Display"
         Depth       24
         Modes       "1280x1024" "1024x768" "800x600" "640x480"
         ViewPort    0 0
      EndSubsection EndSection
```



```
Section "Screen"
      Identifier  "Screen 2"
      Device      "IntelEGD-2"
      Monitor     "Monitor2"
      DefaultDepth 24
      Subsection "Display"
         Depth       8
         Modes       "1280x1024" "1024x768" "800x600" "640x480
         ViewPort    0 0
      EndSubsection
      Subsection "Display"
         Depth       16
         Modes       "1280x1024" "1024x768" "800x600" "640x480"
         ViewPort    0 0
      EndSubsection
      Subsection "Display"
         Depth       24
         Modes       "1280x1024" "1024x768" "800x600" "640x480"
         ViewPort    0 0
      EndSubsection
EndSection
Section "ServerLayout"
      Identifier  "Dual Head Layout"
      Screen "Screen 1"
      Screen "Screen 2" Right Of "Screen 1"
      InputDevice "Mouse1" "CorePointer"
      InputDevice "Keyboard1" "CoreKeyboard"
EndSection
```

7.4.8 Vertical Extended Configuration

For configuring Vertical Extended mode, several attributes are specified in the xorg.conf and user_config.c files.

For the `xorg.conf` file, the following attributes are specified:

Option “ALL/1/General/DisplayConfig” “2”

Option “ALL/1/General/SeamlessModeset” “1”

For the `user_config.c` file, set the DisplayConfig = 5 in the DRM configuration.

Note: VEXT mode only available for the MeeGo OS.

7.4.9 Enabling Runtime “DIH to CLONE” Function

The “DIH to CLONE” function allows you to switch seamlessly from DIH or VEXT display mode to CLONE mode during runtime while preserving FrameBuffer context of the old DIH/VEXT mode without blanking or flickering.

Note: This function is supported only on the MeeGo OS.

To enable this function, set following configuration option:

Option “All/<configID>/General/ DihCloneEnable” “1”

Option “All/<configID>/General/ DisplayConfig” “8”

During runtime, you can switch DIH or VEXT display mode to clone mode and vice versa via escape protocol. For details of the escape protocol, please refer to *Intel® Embedded Graphics Drivers, Intel® Embedded Media and Graphics Driver, EFI Video Driver, and Video BIOS API Reference Manual*.

7.4.10 Configuring Accelerated Video Encode and Decode for Intel® EMGD

See the following sections for configuration details for accelerated video decode:

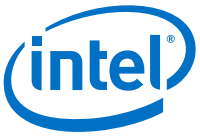
- "Hardware Video Acceleration Overview" on page 153
- "Installing the VA Library" on page 154
- "Installing Third-party GStreamer Plugins (MI-X)" on page 154
- "Installing MPlayer" on page 157
- "Video Encode with USB Camera" on page 158
- "Graphics Port Initialization" on page 159
- "OpenGL Support" on page 159
- "Sample Advanced EDID Configurations for Linux* OS" on page 160
- "Flash Plugin Support" on page 160
- "DMA to Overlay Support" on page 161
- "DMA to Framebuffer Support" on page 161
- "Seamless Switches from Clone Mode to VEXT Mode" on page 161

7.4.10.1 Hardware Video Acceleration Overview

Hardware Video Acceleration is the use of a specialized video engine to decode video streams (such as MPEG-2, MPEG-4, H.264 and VC-1) in order to free up the processor from having to do all of the decoding. Only some chipsets (such as the US15W and Intel® Atom™ Processor E6xx) support a video engine. The flow of video through the various components generally is as follows:

1. The video player, such as the Intel® EMGD-validated MPlayer, reads a video file and determines the type.
2. Based on type, the proper codec shared library object is loaded.
3. The codec loads the VA library shared library object.
4. The VA library loads the `emgd_drv_video.so` shared library object.
5. The `emgd_drv_video.so` communicates, over the X wire protocol, with the Intel® EMGD X driver to send encoded video to the hardware for decoding.

Please follow the installation steps in Section 7.4.10.2, "Installing the VA Library" on page 154 to install VA library.



7.4.10.2 Installing the VA Library

Note: Timesys Fedora Remix 14 and MeeGo 1.2 come with libva 1.0.12 pre-installed. For Timesys Fedora Remix 14 you must export the libva path using the first step below.

1. For Timesys Fedora Remix 14 only export the libva path:
   su
   echo 'export LIBVA_DRIVERS_PATH=/usr/lib/xorg/modules/drivers' > /etc/profile.d/libva-path.sh
   export LIBVA_DRIVERS_PATH=/usr/lib/xorg/modules/drivers
   For Meego 1.2, skip this step.
2. Install the libX and libva development libraries:
   zypper install libX* mesa-libGLU-devel
   wget http://repo.meego.com/MeeGo/releases/1.2.0/repos/oss/ia32/packages/i586/libva-1.0.12-1.4.i586.rpm
   wget http://repo.meego.com/MeeGo/releases/1.2.0/repos/oss/ia32/packages/i586/libva-devel-1.0.12-1.4.i586.rpm
   rpm --force -Uvh libva-1.0.12-1.4.i586.rpm
   rpm --force -Uvh libva-devel-1.0.12-1.4.i586.rpm
3. Set your allocated video memory by editing the boot configuration file as follows:
   — For MeeGo, edit `/boot/extlinux/extlinux.conf`:
      i. Find the line containing your boot kernel options. It will be similar to
         `append ro root=/dev/sda2 quiet vga=current`
      ii. To the end of this line, add
         `vmalloc=256MB`
      For example,
      `append ro root=/dev/sda2 quiet vga=current vmalloc=256MB`
   — For Timesys Fedora Remix 14, edit `/boot/grub/grub.conf`:
      i. Find the line containing your boot kernel options. It will be similar to
         `kernel /vmlinuz ro root=/dev/mapper/VolGroup-lvroot rhgb quiet`
      ii. To the end of this line, add
         `vmalloc=256MB`
      For example,
      `kernel /vmlinuz ro root=/dev/mapper/VolGroup-lvroot rhgb quiet vmalloc=256MB`
4. Reboot

7.4.10.3 Installing Third-party GStreamer Plugins (MI-X)

MI-X plugins provided by Intel along with the Intel® EMGD package consist of a GStreamer decoder element and GStreamer sink element, which is responsible for communicating with Intel® EMGD via the industry standard VAAPI specification. The MixVideoDecoder element parses the compressed elementary video bit-stream and sends decode-specific data to Intel® EMGD for handling.

Note: MI-X is provided by Intel as a proprietary solution for evaluation purposes. If a customer wants to use it in a production solution, they should consult with their lawyers on the requirement for a codec license for permissible use.

The GStreamer package contains a streaming media framework that enables applications to share a common set of plug-ins for things like video decoding and encoding, audio encoding and decoding, audio and video filters, audio visualization, Web streaming and anything else that streams in real-time or otherwise.

Getting the audio working requires various libraries and plug-ins. Due to various software dependency conflicts, the installation process requires following the instructions precisely. The procedure may also change over time as the software packages are updated.

Installing GStreamer with Audio for Timesys Fedora Remix 14

1. Make sure `libva` is installed correctly by typing the following command in a terminal:
   vainfo
   If you get an error, please see Section 7.4.10.2, “Installing the VA Library” on page 154 for instructions on installing `libva`.
2. Download these RPMs:
   http://download1.rpmfusion.org/free/fedora/rpmfusion-free-release-stable.noarch.rpm
   http://download1.rpmfusion.org/nonfree/fedora/rpmfusion-nonfree-release-stable.noarch.rpm
3. Install the RPMs using the following commands:
   rpm -ivh rpmfusion-free-release-stable.noarch.rpm
   rpm -ivh rpmfusion-nonfree-release-stable.noarch.rpm
4. Install the GStreamer plugins with the command:
   yum --releasever=15 install gstreamer-plugins-base gstreamer-plugins-good gstreamer-plugins-bad gstreamer-plugins-ugly gstreamer-ffmpeg --nogpgcheck
5. Install the MI-X package from the driver directory:
   cd /<your path>/IEMGD_HEAD_Linux/common/video_plugin
   rpm -Uvh *.rpm
6. SELinux should be disabled. If it is not, refer to Section 7.3.3 for instructions on doing so.
7. Run **gst-inspect** to check installed codecs. For example, using `gst-inspect` displays all installed plugins, while using `gst-inspect qtdemux` indicates whether the `qtdemux` plugin is installed and its status.

Installing GStreamer with Audio for MeeGo

1. Make sure `libva` is installed correctly by typing the following command in a terminal:
   vainfo
   If you get an error, please see Section 7.4.10.2, “Installing the VA Library” on page 154 for instructions on installing `libva`.
2. As root user, change directories:
   cd /
3. Download the files https://bugs.meego.com/attachment.cgi?id=4761 and https://bugs.meego.com/attachment.cgi?id=4762 to your / directory.
4. From the / directory, uncompress the files using:
   tar xvf <filename>
   Repeat this for each file.

5. Import the rpm keys. You will see many error messages, which can be safely ignored:
   cd /etc/pki/rpm-gpg
   rpm --import *
6. In the directory `/etc/zypp/repos.d/`edit the following files:
   - `rpmfusion-free.repo`
   - `rpmfusionfree-updates.repo`
   - `rpmfusion-nonfree.repo`
   - `rpmfusion-nonfree-updates.repo`
   - `fedora.repo`
   - `updates.repo`

   Make the following changes to each file:
   a. In the line beginning with `baseurl`, change the number `14` to `15`.
   b. In the line beginning with `gpgcheck`, change the number `1` to `0`.
7. Download the file https://fedoraproject.org/static/069C8460.txt, save it as `069C8460.txt` and then import it by rpm:
   r**pm --import 069C8460.txt**
8. Refresh zypper's cache:
   zypper clean --all
   zypper --gpg-auto-import-keys refresh
9. Install the following packages:
   gstreamer-plugins-bad gstreamer-plugins-ugly gstreamerffmpeg gstreamer-devel
   Zypper will ask you to make a choice twice. Select option 1 both times.
10. Reboot.
11. Install the MI-X package from the driver directory:
    cd /<your path>/IEMGD_HEAD_Linux/common/video_plugin
    rpm -Uvh *.rpm
12. Run **gst-inspect-0.10** to check installed codecs.
    For example, using **gst-inspect-0.10** displays all installed plugins, while using **gst-inspect qtdemux** indicates whether the `qtdemux` plugin is installed and its status.
13. Reinstall the `mpfr` library:
    zypper in --force mpfr
14. Disable the rpmfusion, planetccrma, and Timesys Fedora Remix 14 repositories by editing the following files:
    - `planetccrma.repo`
    - `planetcore.repo`
    - `planetupdates.repo`
    - `updates.repo`
    - `fedora.repo`
    - `rpmfusion-free.repo`
    - `rpmfusion-free-updates.repo`
    - `rpmfusion-nonfree.repo`
    - `rpmfusion-nonfree-updates.repo`

In each file, change the line
`enabled=1`
to
`enabled=0`

GStreamer Examples

- Playing a video H.264 file using gst-launch:
  - For Timesys Fedora Remix 14, use the following command:
    gst-launch filesrc location=./firefly.mp4 ! qtdemux name=demux ! queue ! MixVideoDecoderH264 ! vaimagesink demux. ! queue ! faad ! alsasink
  - For MeeGo 1.2, use the following command:
    gst-launch-0.10 filesrc location=./firefly.mp4 ! qtdemux name=demux ! queue ! MixVideoDecoderH264 ! vaimagesink demux. ! queue ! faad ! alsasink
- Playing a video MPEG-4 part 2 file using gst-launch:
  - For Timesys Fedora Remix 14, use the following command:
    gst-launch filesrc location=./firefly.mp4 ! qtdemux name=demux ! queue ! MixVideoDecoderMPEG4 ! vaimagesink demux. ! queue ! faad ! alsasink
  - For MeeGo 1.2, use the following command:
    gst-launch-0.10 filesrc location=./firefly.mp4 ! qtdemux name=demux ! queue ! MixVideoDecoderMPEG4 ! vaimagesink demux. ! queue ! faad ! alsasink

7.4.10.4 Installing MPlayer

Note: You must install `libva` before installing or running MPlayer. Please follow the installation steps in "Installing the VA Library" on page 154.

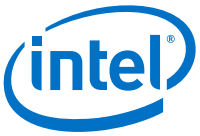
MPlayer has been validated for Linux. It supports video formats of MPEG-2, VC1, H.264 and MPEG-4. To install MPlayer, use the following setup procedures. Ensure that you have logged in as root before starting the setup.

1. Install `yasm` and `git`:
   yum –y install yasm git
2. To enable audio, install the `alsa` dependencies:
   yum –y install alsa*
3. Clone the MPlayer `git` repository:
   git clone git://gitorious.org/vaapi/mplayer.git

   If the command executes successfully, the MPlayer tree is downloaded in the target under the `mplayer` directory.
4. Configure MPlayer:
   cd mplayer
   git checkout –t origin/hwaccel-vaapi
   ../configure

   At the `No FFmpeg` checkout prompt, press **Enter**.

   cd ffmpeg
   git checkout –b ffmpeg-0.6.3 [you can use the latest version tag]



5. Return to the `mplayer` directory:
   cd ..
6. In the `mplayer` directory, configure MPlayer options:
   ./configure --prefix=/usr/local --enable-xv --enable-gl --enable-vaapi --disable-vdpau --disable-mencoder --disable-faad --enable-runtime-cpudetection
7. Compile and install `mplayer`:
   make
   make install
8. To run MPlayer with `vaapi`, add the options `-vo vaapi -va vaapi`.
   For example:
   /opt/mplayer/bin/mplayer -vo vaapi -va vaapi -ao alsa /root/videos/H264_Dolphins_720p.mp4

7.4.11 Video Encode with USB Camera

Video encode in Linux is supported up to standard definition at 30 fps.

1. Connect USB camera to the Atom E6xx board.
2. Try to run a simple camera test using the following command:
   — For Timesys Fedora Remix 14, use the following command:
   gst-launch v4l2src ! ffmpegcolorspace ! xvimagesink
   — For MeeGo 1.2, use the following command:
   gst-launch-0.10 v4l2src ! ffmpegcolorspace ! xvimagesink
   If the camera is not being detected, this might be because the UVC driver is not installed. You can download `uvcvideo-756ad91a832e[1].tar.gz` from http://linux-uvc.berlios.de/#download.
3. To encode video to an H.264 file:
   — For Timesys Fedora Remix 14, use the following command:
   gst-launch -e v4l2src ! ffmpegcolorspace ! 'video/x-raw-yuv,width=640,height=480,framerate=30/1' ! MixVideoEncoderH264 need-display=1 name=enc ! queue ! qtmux ! queue ! filesink location=/root/camera.h264 enc. ! vaimagesink
   — For MeeGo 1.2, use the following command:
   gst-launch-0.10 -e v4l2src ! ffmpegcolorspace ! 'video/x-raw-yuv,width=640,height=480,framerate=30/1' ! MixVideoEncoderH264 need-display=1 name=enc ! queue ! qtmux ! queue ! filesink location=/root/camera.h264 enc. ! vaimagesink
4. To encode video to a MPEG-4 part 2 file:
   — For Timesys Fedora Remix 14, use the following command:
   gst-launch -e v4l2src ! ffmpegcolorspace ! 'video/x-raw-yuv,width=640,height=480,framerate=30/1' ! MixVideoEncoderMPEG4 need-display=1 name=enc encode-format=video/mpeg profile=2 ! queue ! qtmux ! queue ! filesink location=/root/camera.mpeg4 enc. ! vaimagesink
   — For MeeGo 1.2, use the following command:
   gst-launch-0.10 -e v4l2src ! ffmpegcolorspace ! 'video/x-raw-yuv,width=640,height=480,framerate=30/1' ! MixVideoEncoderMPEG4 need-display=1 name=enc encode-format=video/mpeg profile=2 ! queue ! qtmux ! queue ! filesink location=/root/camera.mpeg4 enc. ! vaimagesink

5. To play back the recorded video:
   - For Timesys Fedora Remix 14, use the following command:
     gst-launch filesrc location=camera.mpeg4 ! qtdemux ! queue ! MixVideoDecoderMPEG4 ! vaimagesink
   - For MeeGo 1.2, use the following command:
     gst-launch-0.10 filesrc location=camera.mpeg4 ! qtdemux ! queue ! MixVideoDecoderMPEG4 ! vaimagesink
6. To get more details on encoder properties, execute:
   - For Timesys Fedora Remix 14, use the following command:
     gst-inspect MixVideoEncoderMPEG4
   - For MeeGo 1.2, use the following command:
     gst-inspect-0.10 MixVideoEncoderMPEG4

7.4.12 Graphics Port Initialization

When used with a graphic chipset that supports multiple graphics pipelines, the driver supports multiple screens and Xinerama. Enable this support by creating additional `Device` sections for each additional graphics device on the PCIe bus. The driver locates the first device on the bus and associates it with the device section that matches (or one that does not specify a busID). This becomes the primary display. If the chipset supports multiple display pipes, and the config file specifies two `Device` sections and two `Screen` sections, the driver attempts to operate in a DIH mode. After all the graphics devices and device sections have been matched up, the driver attempts to allocate any remaining output ports and attach them to the primary graphics device.

For example, two pipes and two ports allow for dual independent displays.

7.4.13 OpenGL Support

The Intel® EMGD supports OpenGL* for the following Intel chipsets:

- Intel® Atom™ Processor E6xx
- Intel® System Controller Hub US15W/US15WP/WPT chipset

The OpenGL implementation for Intel® EMGD consists of three components:

- Mesa DRI module (emgd_dri.so)
- Kernel DRM module (emgd.ko)
- Device Dependent X (DDX) driver (emgd_drv.so)

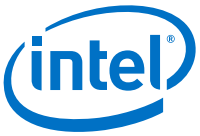
Installing the Intel® EMGD OpenGL driver provides a fully hardware accelerated implementation of the OpenGL library to applications. This implementation makes use of a Direct Rendering technology, which allows the client to directly write to DMA buffers that are used by the graphics hardware.

Due to the use of direct rendering technology, system designers should take special care to ensure that only trusted clients are allowed to use the OpenGL library. A malicious application could otherwise use direct rendering to destabilized the graphics hardware or, in theory, elevate their permissions on the system.

A system designer can control the access to the direct rendering functionality by limiting the access to the DRI device file located at:

```
/dev/dri/card0
```

The permissions on this device are set by the X server using the information provided in the “DRI” section of the configuration file.



See also "2D/3D API Support" on page 199.

7.4.14 Sample Advanced EDID Configurations for Linux* OS

The `edid_avail` and `edid_not_avail` parameters control the available timings for any display. Use the `edid_avail` parameter when reading EDID information from the display. If the driver is unable to read EDID information from the display, or if the `edid` parameter in the config file is set to "`0`" (disable), use the settings of the `edid_not_avail` parameter. Please see Section 3.0, "Platform Configuration Using CED" on page 21 and CED help.

An `edid` option can be placed in the `xorg.conf` to control the behavior of the overall driver. EDID settings also exist within CED that control the behavior on each port (`edid`, `edid_avail`, and `edid_not_avail`). The combination of these settings determines how the driver behaves. The table below shows various configurations and the expected behavior of the driver.

Table 33. Sample Advanced EDID Configurations for Linux* OS

Case	CED: Per port "edid" option	Expected driver behavior
1.	No `edid` flag specified	For every port, driver uses `edid_avail`.
2.	`edid=0` for some ports and `edid=1` for some ports	For edid=0 ports, driver uses `edid_not_avail` flags. For edid=1 ports driver uses `edid_avail` flags.
3.	Setting does not matter.	For all ports driver will not read `edid` and interprets `edid_not_avail` flags. Driver overrides any per-port `edid` flags, treats all displays as EDID-less displays, and uses `edid_not_avail` flags.
4.	`edid=0` for some ports and `edid=1` for some ports	Same as case 2

Notes: For all cases:

1. If there is not an `edid_not_avail` flag specified for a port, and an EDID-less display is detected, the driver defaults to using the standard built-in timings for that port.
2. If there is not an `edid_avail` flag specified for a port, and an EDID display is detected, the driver defaults to using the EDID data from the display, plus any user-specified DTDs.
3. If `edid=1` and the display device is EDID-less, the driver uses `edid_not_avail` flags.

7.4.15 Flash Plugin Support

Hardware-accelerated Flash is supported on the following system configurations:

- Timesys Fedora Remix 14 with Firefox 3.6.x
- MeeGo IVI 1.2 with Chromium 11

To download the plugin, and for the latest plugin-specific information, please refer to the Intel EDC website here: https://registrationcenter.intel.com/RegCenter/ComForm.aspx?ProductID=1618

7.4.16 DMA to Overlay Support

The DMA to Overlay feature enables you to design an external V4L2 camera source driver to DMA video content (in the form of YUV pixels) directly into video memory that gets displayed via the overlay plane. The DMA to Overlay feature is supported in the following two conditions:

- From kernel level without X Windows and GStreamer on primary display only. For details of the feature design and configuration, please refer to the *Early Direct Camera Presentation on Intel® Atom™ Processor E6xx Series Application Note.*
- In X Windows with GStreamer. For details of the feature design and configuration, please refer to the *Intel® EMGD Direct Camera Presentation Interface Application Note*.

7.4.17 DMA to Framebuffer Support

The DMA to Framebuffer feature enables you to design an FPGA V4L2 driver to DMA video content (in the form of RGB format) directly into video memory that gets displayed via the framebuffer.

For details of the feature design and configuration, please refer to the Intel® EMGD *Direct Camera Presentation Interface Application Note*.

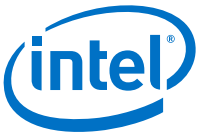
7.4.18 Seamless Switches from Clone Mode to VEXT Mode

Under normal circumstance, after system boot up and starting X, you may see programs or client applications loading information on the screen before the home screen is fully up and running. To make the program/client application loading screen invisible, Intel® EMGD provides an option for you to retain the boot up splash screen and switch to the home screen through an escape function call when the home screen is ready.

This feature is implemented for a specific usage model only, where the system is expected to boot up in Clone mode with a splash/logo screen and continues to be shown on the display until the X-Window/Home screen is fully up and running in VEXT mode. The transition from Clone to VEXT display mode will be handled seamlessly.

To enable this feature, use CED to generate the EFI and the Linux driver with appropriate attributes set.

1. Configure and generate EFI with Clone mode configuration.
2. Configure and generate the Linux driver installation package with VEXT mode and seamless mode flag set.
   a. For VEXT mode: configure the CED->Chipset Configuration Page -> Display Configuration Mode= Vertical Extended. If you like to edit the `xorg.conf` and `user_config.c` files directly, ensure that you specify the following attributes in the files:
      - `xorg.conf` file: Option “ALL/1/General/DisplayConfig” “2”
      - `user_config.c` file: DisplayConfig = 5 in the DRM configuration array
   b. Seamless mode from EFI GOP to DRM: configure the CED-> EFI GOP Configuration Page -> Enable Seamless Mode = Checked (1). This configuration applies to the `user_config.c` file. If you like to edit the file directly, ensure that you specify the following attribute in the file:
      `user_config.c` file: set the Seamless = 1 in the parameter configuration array



c. Seamless mode from DRM to X-Window: configure the CED-> EMGD Package Page->Linux Option Page -> Enable Seamless Mode Set for DRM->X transition = checked (1). This configuration applies to the `xorg.conf` file. If you like to edit the file directly, ensure that you specify the following attribute in the file: `xorg.conf` file: option "ALL/1/General/SeamlessModeset" "1"

3. Ensure the resolution configured for clone mode and VEXT primary display are of the same resolution.

Seamless switches from Clone mode to VEXT mode work in such a way that EFI boots up the system in Clone mode and the DRM driver will check the configuration mentioned above in steps 1-3. If the requirement is fulfilled, the display plane register will be locked and the framebuffer content for both screens in Clone mode will be retained. The kernel mode driver will keep deferring plane register updates (for VEXT mode) until the client application sends an ESCAPE API to end it.

To unlock the display plane register, the client application sends ESCAPE API INTEL_ESCAPE_SHOW_DESKTOP to the X11 driver. After the plane registers are unlocked, the registers are updated and enable X rendered screens to display in VEXT mode.

Note: This feature is applicable only for system configurations with a combination of EFI with Clone mode and the graphics driver with VEXT mode configurations.

7.5 Runtime Configuration GUI

You can change the configuration and runtime attributes of the driver using the `emgdgui` runtime configuration tool. The Intel® EMGD GUI (`emgdgui`) is an application that is used to view and control the Intel® EMGD. It retrieves status of the display and driver and is also used to configure the supported display attributes. You can change the configuration and runtime attributes of the driver using the `emgdgui` runtime configuration tool, which resides in the `<OS name>/Utilities` directory. The `emgdgui` also demonstrates multi-monitor support. Note that changes made with emgdgui are not permanent and are lost upon rebooting. For permanent changes, the `xorg.conf` file has to be edited directly. See Section 7.4.5, "The Linux* OS Configuration File" on page 140 for help on this.

7.5.1 emgdgui Setup

To run `emgdgui`, you need to ensure that the X server has been configured to use Intel® EMGD. See Section 7.4.1, "Configuration Overview" on page 138 for details on configuring and installing the Intel® EMGD.

You need GTK+ and libglade, which are part of the Linux* distribution and should already be installed.

Note: It is not necessary to manually install the `emgdgui`, the `install.sh` has included the installation. The `emgdgui` is installed to the system and you can execute it by typing **emgdgui** in xterm or find the **"**EMGD GUI Utility**"** in System Tools.

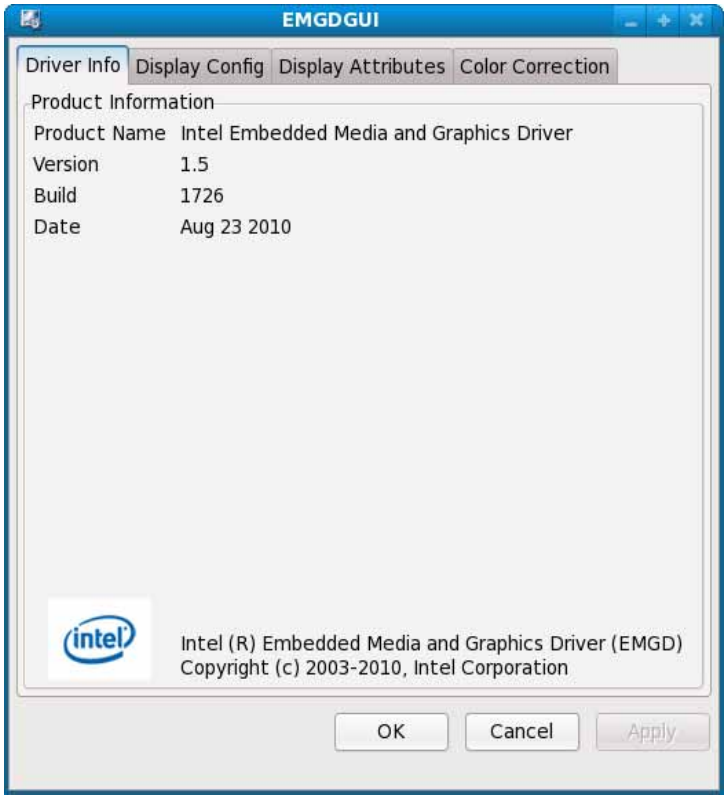
7.5.2 Using the emgdgui Runtime Configuration Utility

The `emgdgui` application provides four tabs: **Driver Info**, **Display Config**, **Display Attributes,** and **Color Correction**.

- **Driver Info**: Contains the driver information.
- **Display Config**: Contains current display information and allows configuration of display configurations, display resolutions for primary and secondary displays and enabling/disabling of a specified port.
- **Display Attributes**: Contains the supported Port Driver (PD) attributes and allows configuration of PD attributes.
- **Color Correction**: Contains current color-correction information for the framebuffer and overlay. Using this tab, you can change the framebuffer and overlay color settings.

The figure below shows the **Driver info** tab.

Figure 36. Example Linux Runtime Configuration GUI — Driver Info Tab

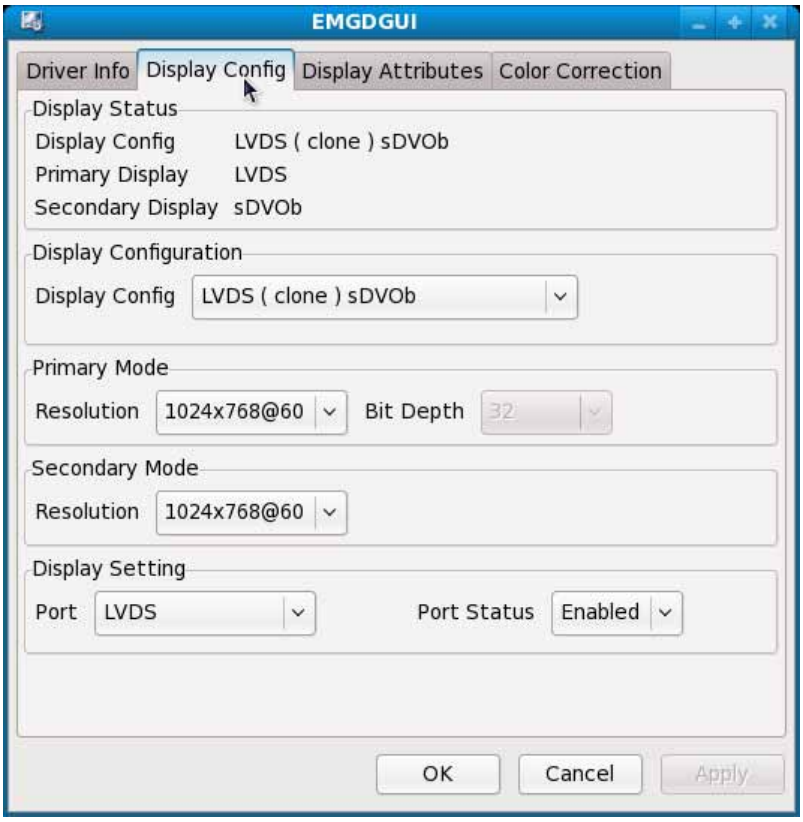


To view current display information and or to change the current configuration of display configurations, display resolutions of the primary and secondary displays, and enabling/disabling of a specified port, click the **Display Config** tab.

Note: If you make any changes to the configuration, click **Apply** for the changes to take effect.

The figure below shows a sample configuration.

Figure 37. Example Linux* Runtime Configuration GUI — Display Config Tab



The **Display Status** section of the above dialog shows the current configuration for the **Primary** and **Secondary** displays.

In the **Display Configuration** section of the dialog, select the required display configuration in the **Display Config** drop-down list. This allows the user to choose between Single, Clone, and Extended for all connected ports. A maximum of two ports per display configuration is currently allowed.

In the **Primary Mode** and **Secondary Mode** sections of the dialog, you can change display resolutions via the **Resolution** drop-down list.

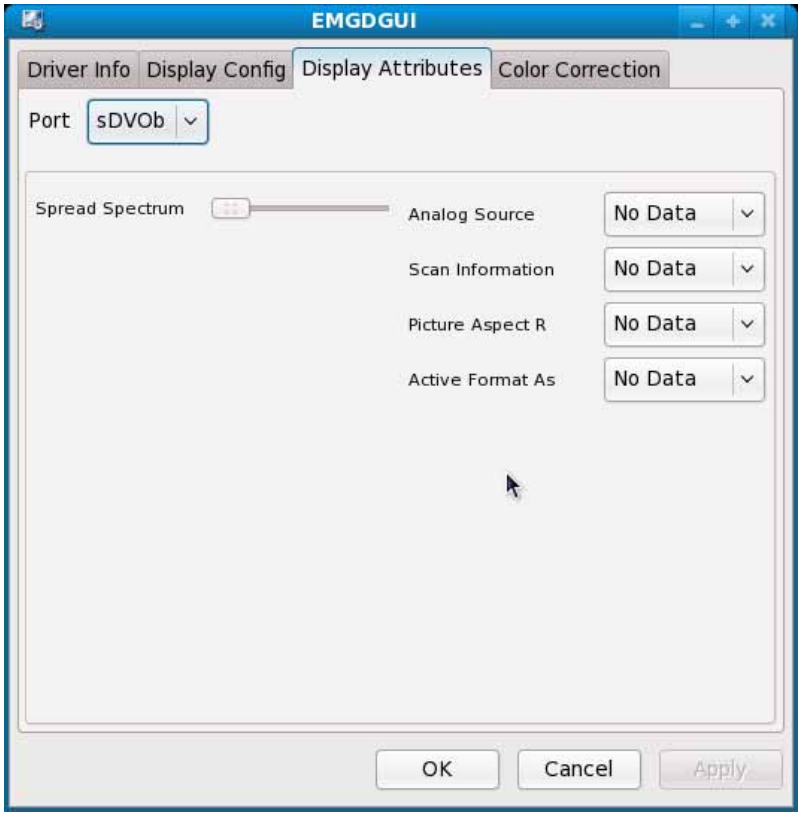
In the **Display Settings** section of the dialog you can view and change the settings for a port and then rotate and flip the display via the appropriate drop-down lists:

- **Port**: Allows you to select the required port.
- **Port Status**: Allows you to enable or disable the selected port.

Note: For Chrontel 7306, there is an additional Refresh button not shown in the screen shot above. If you change the output type between HDMI and VGA during runtime, click this button to reset the display output to the new display port.

To view or change the attributes for a port, click the **Display Attributes** tab. The figure below shows a sample configuration. Please see Appendix B for detailed information on port driver attributes.

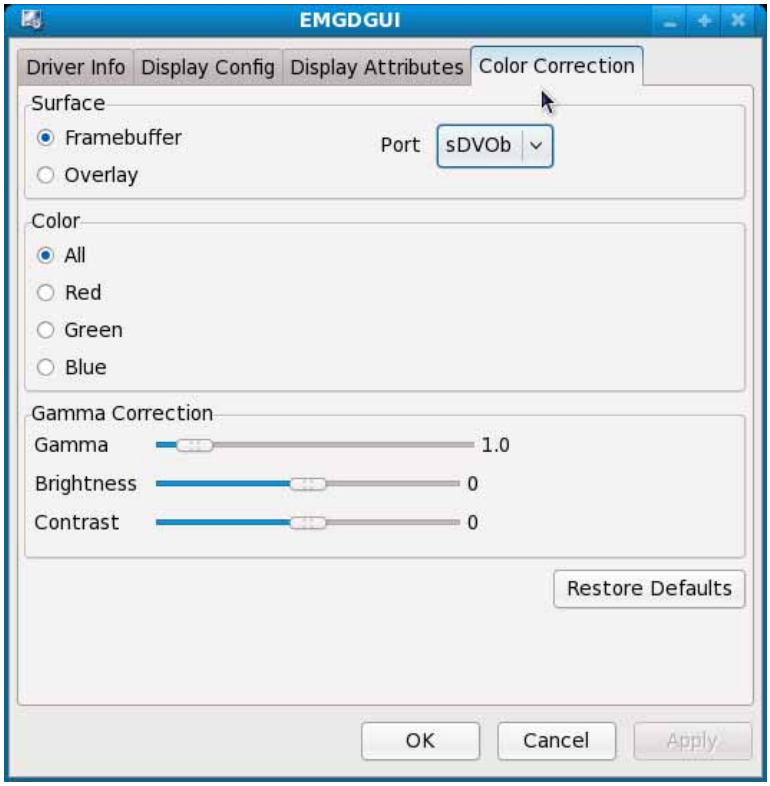
Figure 38. Example Linux* Runtime Configuration GUI — Display Attributes Tab



To view and change color corrections, click the **Color Correction** tab. The two figures below show sample Color Correction tab screens for **Framebuffer** and **Overlay**, color correction values for which are shown in Table 27 and Table 28.

Note: If you make any changes to the color-correction attributes, click **Apply** for the changes to take effect.

Figure 39. Example Linux* Runtime Configuration GUI — Color Correction Tab (Framebuffer)



The following steps present an example color-correction procedure:

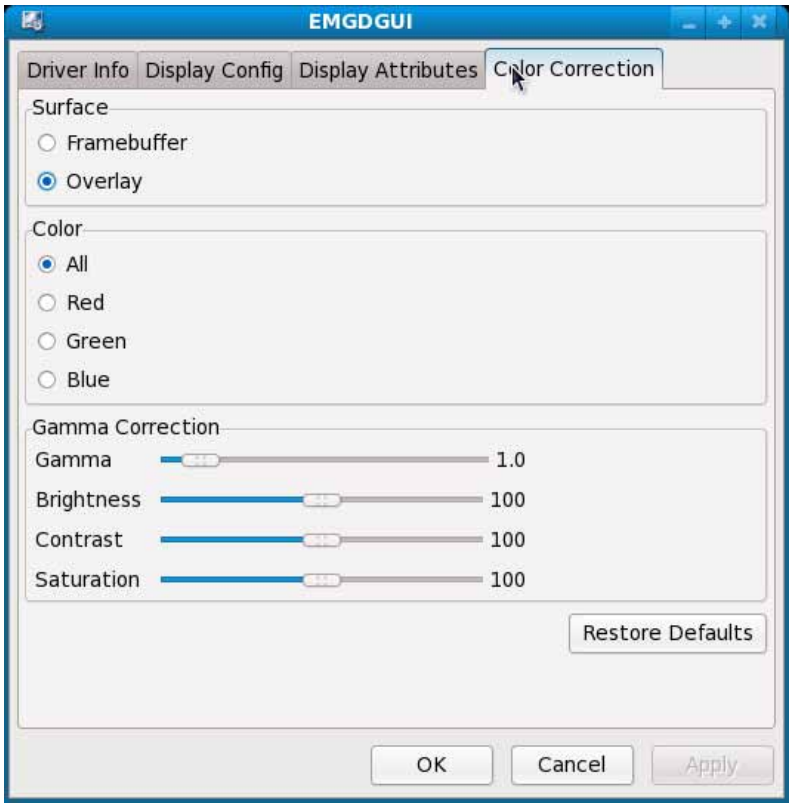
a. Select **Framebuffer** in the **Surface** section and select the appropriate port for the color correction to be applied to or select **Overlay** in the Surface section for color correction to be applied to the overlay.

b. Select the required color to be corrected in the **Color** section.

c. Select the required color attribute to be corrected in the **Gamma Correction** section.

d. Click **Restore Defaults** to restore the default values.

Figure 40. Example Linux* Runtime Configuration GUI — Color Correction Tab (Overlay)

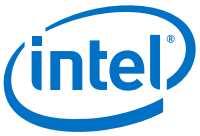


7.5.3 Environment Variables

7.5.3.1 VideoKey

This feature disables the automatic VideoKey painting of VAAPI and Xv API when FbBlendOvl mode is enabled by setting the EMGD_VIDEO_SKIP_COLORKEY_FILL environment variable to 1. The goal is that the application draws the rgb+alpha background (with subpicture) to the framebuffer with the video surface being displayed on the overlay plane. If EMGD_VIDEO_SKIP_COLORKEY_FILL is not set to 1, VAAPI and Xv API override this painting, making it impossible to paint a subpicture in the background.

- To enable this feature, execute
  'export EMGD_VIDEO_SKIP_COLORKEY_FILL=1'
- To disable this feature, execute
  'export EMGD_VIDEO_SKIP_COLORKEY_FILL=0'



7.6 Runtime Freeze and UnFreeze Display (Framebuffer)

Intel® EMGD provides Freeze and UnFreeze display framebuffer at runtime through escape function calls. This feature allows you to freeze both primary and secondary displays' framebuffers and resume respectively. The overlay or sprite planes are not affected by the freeze/unfreeze function calls and therefore a video playback using the overlay plane will keep running in freeze framebuffer mode. Please note that when the framebuffer is frozen, you should not perform any mode change to alter the display setting such as changing the display resolution.

Note: This function is supported only on the MeeGo OS. To enable this function, set the following configuration option:

Option "ALL/<configID>/General/FreezeFB" "1"

Please refer to the *Intel® Embedded Graphics Drivers, Intel® Embedded Media and Graphics Driver, EFI Video Driver, and Video BIOS API Reference Manual* for a detailed description of the Freeze/Unfreeze Framebuffer function calls.

7.7 Wayland and Weston Setup

Wayland is a display server protocol for applications that targets to simplify and improve the graphics stack in Linux and other Unix-based operating systems, with the long-term goal of eventually phasing out the legacy X server altogether. Weston is a proof of concept, or reference, compositing windows manager for Wayland. Both these technologies are in the alpha/beta stage, and should be considered extremely unstable.

7.7.1 Installation

This section covers installation of Wayland and/or Weston. Please note that due to the rapidly changing state of these two software packages, it is impossible to provide exact directions for installing them. Rather, this section will cover general guidelines and dynamic online resources for the user. The information provided here is only accurate at the time of publication, and may well be outdated by the time EMGD 1.14 is released. Please regard these steps as an example installation sequence rather than a final authoritative set of instructions.

To install Wayland and Weston on your system, there are two approaches.

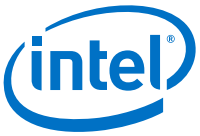
7.7.1.1 7.6.1.1 Installation by RPM

This approach uses the MeeGo 1.3 release (currently beta release) and installs the Wayland/Weston packages directly from the MeeGo 1.3 repository. The website at http://wiki.meego.com/Wayland_in_MeeGo provides reasonably up-to-date information on this sequence. At the time of publication, Weston is not yet available via this approach, but it should be made available eventually.

7.7.1.2 7.6.1.2 Installation by Source

This approach builds Wayland and Weston from source. Both MeeGo 1.2 and 1.3 may be used for this approach. Example steps are as follows:

1. Create a export.sh file, with the following contents
   export WLD=$HOME/wl
   export LD_LIBRARY_PATH=$WLD/lib
   export PKG_CONFIG_PATH=$WLD/lib/pkgconfig/:$WLD/share/pkgconfig/
   export ACLOCAL="aclocal -I $WLD/share/aclocal"
   export C_INCLUDE_PATH=$WLD/include
   export LIBRARY_PATH=$WLD/lib
   export PKG_CONFIG_ALLOW_SYSTEM_CFLAGS=1
2. Execute the file:
   . export.sh
3. Create the directory:
   mkdir –p $HOME/wl/share/aclocal
4. Install development tools:
   zypper in –t pattern meego-development-tools
5. For the following list of libraries, download them from the provided link and install them with the specified options.
   a. Go to the dir `$HOME/wl`
   b. Download the first package (e.g., libffi) from the download link provided. A basic knowledge of git is assumed.
   c. Decompress the package if necessary, and enter the package directory.
   d. Run the command:
      ./autogen.sh <Configure options>
      For example, for libffi, the command would be "./autogen.sh --prefix=$WLD"
      If `autogen.sh` does not exist, use `configure` in its place.
   e. Run the commands:
      make
      make install
      In general, if compilation of a package fails with an error indicating missing packages, install the `–devel` version of that package via zypper.



f. Repeat steps a through e for each subsequent package listed in the table below.

Package name	Method	Download link	Configure options
Libffi	HTTP	http://sourceware.org/libffi	--prefix=$WLD
llvm	HTTP	http://llvm.org/releases/download.html	--prefix=/usr
glib2	HTTP	http://ftp.gnome.org/pub/gnome/sources/glib	--prefix=$WLD
gobject-introspection	HTTP	http://ftp.gnome.org/pub/GNOME/sources/gobject-introspection/	--prefix=$WLD
gdk-pixbuf	HTTP	http://ftp.gnome.org/pub/GNOME/sources/gdk-pixbuf	--prefix=$WLD
Wayland	Git	git://anongit.freedesktop.org/wayland/wayland	--prefix=$WLD
Drm	Git	git://anongit.freedesktop.org/git/mesa/drm	--prefix=$WLD --enable-nouveau-experimental-api
Macros	Git	git://anongit.freedesktop.org/git/xorg/util/macros	--prefix=$WLD
Glproto	Git	git://anongit.freedesktop.org/xorg/proto/glproto	--prefix=$WLD
dri2proto	Git	git://anongit.freedesktop.org/xorg/proto/dri2proto	--prefix=$WLD
mesa	Git	git://anongit.freedesktop.org/mesa/mesa	--prefix=$WLD --enable-gles2 --disable-gallium-egl --with-egl-platforms=x11,wayland,drm --enable-gbm --enable-shared-glapi
xproto	Git	git://anongit.freedesktop.org/xorg/proto/xproto	--prefix=$WLD
kbproto	Git	git://anongit.freedesktop.org/xorg/proto/kbproto	--prefix=$WLD
libX11	Git	git://anongit.freedesktop.org/xorg/lib/libX11	--prefix=$WLD
libxkbcommon	Git	git://people.freedesktop.org/xorg/lib/libxkbcommon.git	--prefix=$WLD --with-xkb-config-root=/usr/share/X11/xkb
Pixman	Git	git://anongit.freedesktop.org/pixman	--prefix=$WLD
Cairo	Git	git://anongit.freedesktop.org/cairo	--prefix=$WLD --enable-gl --enable-xcb

6. Create a working directory in which to extract the EMGD modules from the rpm.

```
mkdir $HOME/wl/EMGD-extract
cd $HOME/wl/EMGD-extract
cp –vrf <EMGD driver location>/IEMGD_HEAD_Linux/MeeGo_Wayland/
    emgd-bin-xxxx.i586.rpm  $HOME/wl/EMGD-extract
rpm2cpio emgd-bin-xxxx.i586.rpm | cpio –idmv
cp –vrf usr/* /root/wl
```

7. Link the following modules. Please note that version numbers may change.

```
cd /root/wl/lib/
ln –sfv libwayland-egl.so.1.5.15.3226 libwayland-egl.so.1
ln –sfv libgbm.so.1.5.15.3226 libgbm.so.1
ln –sfv libemgdsrv_um.so.1.5.15.3226 libemgdsrv_um.so
ln –sfv libemgdsrv_init.so.1.5.15.3226 libemgdsrv_init.so
ln –sfv libemgdglslcompiler.so.1.5.15.3226 libemgdglslcompiler.so
ln –sfv libEMGDPVR2D_WAYLANDWSEGL.so.1.5.15.3226
    libEMGDPVR2D_WAYLANDWSEGL.so
ln –sfv libEMGDPVR2D_GBMWSEGL.so.1.5.15.3226
    libEMGDPVR2D_GBMWSEGL.so
ln –sfv libEMGDPVR2D_DRIWSEGL.so.1.5.15.3226
    libEMGDPVR2D_DRIWSEGL.so
ln –sfv libOpenVG.so.1.5.15.3226 libOpenVG.so
ln –sfv libOpenVGU.so.1.5.15.3226 libOpenVGU.so
ln –sfv libGLESv2.so.1.5.15.3226 libGLESv2.so.2
ln –sfv libGLES_CM.so.1.5.15.3226 libGLES_CM.so.1
ln –sfv libGLES_CM.so.1 libGLES_CM.so
ln –sfv libEMGDegl.so.1.5.15.3226 libEMGDegl.so
ln –sfv libEMGDScopeServices.so.1.5.15.3226 libEMGDScopeServices.so
ln –sfv libEMGDOGL.so.1.5.15.3226 libEMGDOGL.so
ln –sfv libEMG2d.so.1.5.15.3226 libEMGD2d.so
ln –sfv libEGL.so.1.5.15.3226 libEGL.so.1
```

8. Building Wayland:

   a. Download the EMGD Wayland source from http://download.meego.com/live/home:/matthew:/branches:/devel:/wayland/Trunk/src/

   b. Create a working directory in which to extract the Wayland source code:

```
mkdir /root/wl/EMGD_WAYLAND/
rpm –Uvh wayland-0.1-4.1.src.rpm
cd /root/rpmbuild/SOURCES/
cp –vrf wayland-0.1.tar.bz2 /root/wl/EMGD_WAYLAND/
rm –vrf /root/rpmbuild
```

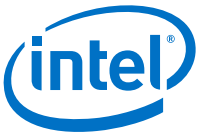
   c. Copy wayland-demos emgd package:

```
rpm –Uvh wayland-demos-0.1-3.1.src.rpm
cd /root/rpmbuild/SOURCES/
cp –vrf wayland-demos-emgd.tar.gz /root/wl/EMGD_WAYLAND/
rm –vrf /root/rpmbuild
```

   d. Install the extracted source files:

```
cd /root/wl/EMGD_WAYLAND/
tar –xvjf wayland-0.1.tar.bz2
cd wayland-0.1/
./autogen --prefix=$WLD
make && make install

tar –xvzf wayland-demos-emgd.tar.gz
cd wayland-demos-emgd/
./autogen --prefix=$WLD
make && make install
```

Wayland installation is done.



9. Building Weston:
   a. Rebuild the cairo package from step 5, replacing the configuration option
      `'--prefix=$WLD --enable-gl --enable-xcb'`
      with
      `'--prefix=$WLD --enable-glesv2 --enable-xcb'`
   b. Download the Weston source from
      git://anongit.freedesktop.org/wayland/Weston
   c. Compile and install Weston:
      cd weston
      ./autogen.sh --prefix=$WLD
      make
      make install

   Weston installation is done.

7.8 Runtime Video Overlay Plane or Sprite C Plane Selection via Escape Function Call

Intel® EMGD provides an option to manually select the video overlay plane or sprite-C plane at runtime through an escape function call. This feature allows you to choose between overlay and sprite-C planes for video playback depending on different use cases. Overlay can be configured to run on the secondary display and sprite-C on the primary display depending on needs. For example, video required for scaling and planar format can use overlay plane in any of the display pipelines (primary or secondary). In addition, the function call allows user to turn on/off FBBlend and overlay color correction (brightness, contrast, saturation, gamma) at runtime.

Note: This function is supported only on the MeeGo OS.

To enable this function, set following configuration option:

Option "ALL/<configID>/General/VideoColorCorrect" "1"

Refer to the *Intel® Embedded Graphics Drivers, Intel® Embedded Media and Graphics Driver, EFI Video Driver, and Video BIOS API Reference Manual* for a detailed description of the Video Overlay/Sprite-C planes selection function call.

7.9 Runtime Primary Display and Secondary Display Swap via Escape Function Call.

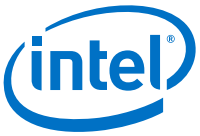
Intel® EMGD provides an option to swap the primary and secondary displays at runtime through an escape function call INTEL_ESCAPE_SWAPDC. This feature provides the capability to swap the framebuffer and the video plane. The escape API acts as a toggle switch; toggling twice un-swaps the display.

Note: This feature is enabled only when the display configuration is in VEXT mode. Do not perform any display mode changes, for example, changing resolution, rotation, etc., while in the display swap state.

In the situation when freeze display (framebuffer) is enabled, every swap API call would forcibly unfreeze the display. In the situation when display is in clone or reverse clone mode, a swap API call will switch the display mode back to VEXT mode followed by swapping the display. Similarly when the swap state is set in VEXT mode, a function call to change the display to clone or reverse clone mode would first unswap the display before putting it in clone or reverse clone mode.

Refer to the *Intel® Embedded Graphics Drivers, Intel® Embedded Media and Graphics Driver, EFI Video Driver, and Video BIOS API Reference Manual* for a description of the Display swap INTEL_ESCAPE_SWAPDC function call.

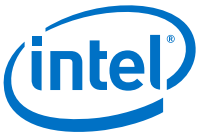
§ §



Appendix A Example INF File

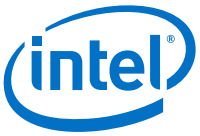
```
;*****************************************************************************
; Filename: iegd.inf
; $Revision: 1.438 $
; $Id: iegd.inf,v 1.438 2010/09/02 09:05:14 anthill Exp $
; $Source: /nfs/fm/proj/eia/cvsroot/ssigd/ial/dx/install/iegd.inf,v $
;
; Copyright (c) 2012 Intel Corporation.  All rights reserved.
;
;*****************************************************************************

[Version]
Signature="$WINDOWS NT$"
Class=Display
ClassGUID={4D36E968-E325-11CE-BFC1-08002BE10318}
Provider=%Intel%
;CatalogFile=iegd.cat
DriverVer = 09/02/2010,1.5.0.1728



;===========================================================================
[SourceDisksNames]
1=%DiskDesc%,,,""

[SourceDisksFiles]
iegdmini.sys = 1
iegdckey.vp  = 1
iegdmsys.vp  = 1
iegdcagt.cpa = 1
iegdcagt.vp  = 1
iegddis.dll  = 1
iegd3dga.dll = 1
iegdglga.dll = 1
libGLES_CM.dll = 1
libGLESv2.dll  = 1
lvds.sys     = 1
sdvo.sys     = 1
sdvo.vp      = 1
lvds.vp      = 1

;===========================================================================
[DestinationDirs]
DefaultDestDir    = 11; System directory
iegd.Display_plb  = 11
iegd.OpenGL_plb   = 11
iegd.Display_tnc  = 11
iegd.OpenGL_tnc   = 11
iegd.Miniport     = 12; Drivers directory
iegd.Copp         = 12
iegd.PortDrvs_plb = 12
iegd.PortDrvs_tnc = 12

;===========================================================================
[Manufacturer]
%Intel%=Intel.Mfg
```

```
;===============================================================================
[Intel.Mfg]
%Intel% %i900G0%  = iegd_plb, PCI\VEN_8086&DEV_8108
%Intel% %iTNC0%   = iegd_tnc, PCI\VEN_8086&DEV_4108
%Intel% %iTNC1%   = iegd_tnc_ext, PCI\VEN_8086&DEV_8182

;===============================================================================
[iegd_plb.GeneralConfigData]
MaximumNumberOfDevices = 2
MaximumDeviceMemoryConfiguration = 256

[iegd_tnc.GeneralConfigData]
MaximumNumberOfDevices = 2
MaximumDeviceMemoryConfiguration = 256

;===============================================================================
[iegd_plb]
CopyFiles = iegd.Miniport, iegd.Display_plb, iegd.OpenGL_plb, iegd.PortDrvs_plb,
iegd.Copp

[iegd_tnc]
CopyFiles = iegd.Miniport, iegd.Display_tnc, iegd.OpenGL_tnc, iegd.PortDrvs_tnc,
iegd.Copp

[iegd_tnc_ext]
CopyFiles = iegd.Null

;===============================================================================
[iegd.Miniport]
iegdmini.sys

[iegd.Copp]
iegdckey.vp
iegdmsys.vp
sdvo.vp
lvds.vp
iegdcagt.cpa
iegdcagt.vp

[iegd.Display_plb]
iegddis.dll
iegd3dga.dll

[iegd.Display_tnc]
iegddis.dll
iegd3dga.dll

[iegd.OpenGL_plb]
iegdglga.dll
libGLES_CM.dll
libGLESv2.dll

[iegd.OpenGL_tnc]
iegdglga.dll
libGLES_CM.dll
libGLESv2.dll

[iegd.PortDrvs_plb]
sdvo.sys
lvds.sys

[iegd.PortDrvs_tnc]
sdvo.sys
lvds.sys

[iegd.Null]
```



```
;=============================================================================
[iegd_plb.Services]
AddService = iegdmini, 0x00000002, iegd_Service_Inst, iegd_EventLog_Inst
AddService = lvds,   ,lvds_Service_Inst, iegd_EventLog_Inst
AddService = sdvo,   ,sdvo_Service_Inst, iegd_EventLog_Inst

[iegd_tnc.Services]
AddService = iegdmini, 0x00000002, iegd_Service_Inst, iegd_EventLog_Inst
AddService = lvds,   ,lvds_Service_Inst, iegd_EventLog_Inst
AddService = sdvo,   ,sdvo_Service_Inst, iegd_EventLog_Inst

[iegd_tnc_ext.Services]
AddService = , 0x00000002; no service

;=============================================================================
[iegd_Service_Inst]
ServiceType    = 1
StartType      = %SERVICE_DEMAND_START%
ErrorControl   = 0
LoadOrderGroup = Video
ServiceBinary  = %12%\iegdmini.sys

[lvds_Service_Inst]
DisplayName    = "lvds"
ServiceType    = %SERVICE_KERNEL_DRIVER%
StartType      = %SERVICE_DEMAND_START%
ErrorControl   = %SERVICE_ERROR_IGNORE%
ServiceBinary  = %12%\lvds.sys

[sdvo_Service_Inst]
DisplayName    = "sdvo"
ServiceType    = %SERVICE_KERNEL_DRIVER%
StartType      = %SERVICE_DEMAND_START%
ErrorControl   = %SERVICE_ERROR_IGNORE%
ServiceBinary  = %12%\sdvo.sys

;=============================================================================
[iegd_EventLog_Inst]
AddReg = iegd_EventLog_AddReg

[iegd_EventLog_AddReg]
HKR,,EventMessageFile,0x00020000,"%SystemRoot%\System32\IoLogMsg.dll;%SystemRoot%\
System32\drivers\iegdmini.sys"
HKR,,TypesSupported,0x00010001,7

;=============================================================================
[iegd_plb.SoftwareSettings]
AddReg = iegd_SoftwareDeviceSettings_plb
AddReg = iegd_ICDSoftwareSettings

[iegd_tnc.SoftwareSettings]
AddReg = iegd_SoftwareDeviceSettings_tnc
AddReg = iegd_ICDSoftwareSettings

;-----------------------------------------------------------------------------
[iegd_SoftwareDeviceSettings_plb]
HKR,, InstalledDisplayDrivers, %REG_MULTI_SZ%, iegddis
HKR,, MultiFunctionSupported, %REG_MULTI_SZ%, 1
HKR,, VgaCompatible, %REG_DWORD%, 0
HKR,, PcfVersion,    %REG_DWORD%, 0x0700

HKR,, ConfigId, %REG_DWORD%, 1

HKR, ALL\1 , name, %REG_SZ%, "US15"
HKR, ALL\1\General , DisplayConfig, %REG_DWORD%, 1
HKR, ALL\1\General , DisplayDetect, %REG_DWORD%, 1
```

```
HKR, ALL\1\General , PortOrder, %REG_SZ%, "24000"
HKR, ALL\1\General , DxvaOptions, %REG_DWORD%, 1
HKR, ALL\1\Port\4\General , name, %REG_SZ%, "LVDS10x7"
HKR, ALL\1\Port\4\General , Rotation, %REG_DWORD%, 0
HKR, ALL\1\Port\4\General , Edid, %REG_DWORD%, 1
HKR, ALL\1\Port\4\General , EdidAvail, %REG_DWORD%, 3
HKR, ALL\1\Port\4\General , EdidNotAvail, %REG_DWORD%, 4
HKR, ALL\1\Port\4\General , CenterOff, %REG_DWORD%, 1
HKR, ALL\1\Port\4\FpInfo , bkltmethod, %REG_DWORD%, 1
HKR, ALL\1\Port\4\FpInfo , BkltT1, %REG_DWORD%, 60
HKR, ALL\1\Port\4\FpInfo , BkltT2, %REG_DWORD%, 200
HKR, ALL\1\Port\4\FpInfo , BkltT3, %REG_DWORD%, 200
HKR, ALL\1\Port\4\FpInfo , BkltT4, %REG_DWORD%, 50
HKR, ALL\1\Port\4\FpInfo , BkltT5, %REG_DWORD%, 400
HKR, ALL\1\Port\4\Dtd\1 , PixelClock, %REG_DWORD%, 65000
HKR, ALL\1\Port\4\Dtd\1 , HorzActive, %REG_DWORD%, 1024
HKR, ALL\1\Port\4\Dtd\1 , HorzSync, %REG_DWORD%, 24
HKR, ALL\1\Port\4\Dtd\1 , HorzSyncPulse, %REG_DWORD%, 136
HKR, ALL\1\Port\4\Dtd\1 , HorzBlank, %REG_DWORD%, 320
HKR, ALL\1\Port\4\Dtd\1 , VertActive, %REG_DWORD%, 768
HKR, ALL\1\Port\4\Dtd\1 , VertSync, %REG_DWORD%, 3
HKR, ALL\1\Port\4\Dtd\1 , VertSyncPulse, %REG_DWORD%, 6
HKR, ALL\1\Port\4\Dtd\1 , VertBlank, %REG_DWORD%, 38
HKR, ALL\1\Port\4\Dtd\1 , Flags, %REG_DWORD%, 0x20000
HKR, ALL\1\Port\4\Attr , 27, %REG_DWORD%, 0
HKR, ALL\1\Port\4\Attr , 26, %REG_DWORD%, 18
HKR, ALL\1\Port\4\Attr , 60, %REG_DWORD%, 1
HKR, ALL\1\Port\2\General , name, %REG_SZ%, "SDVOB"
HKR, ALL\1\Port\2\General , Rotation, %REG_DWORD%, 0
HKR, ALL\1\Port\2\General , Edid, %REG_DWORD%, 1
HKR, ALL\1\Port\2\General , EdidAvail, %REG_DWORD%, 3
HKR, ALL\1\Port\2\General , EdidNotAvail, %REG_DWORD%, 4
HKR, ALL\1\Port\2\General , CenterOff, %REG_DWORD%, 1

HKR,, No_D3D, %REG_DWORD%, 0

HKR,, PortDrivers, %REG_SZ%, "sdvo lvds"
HKR, ALL\1\General, DxvaOptions, %REG_DWORD%, 0x00000001

HKCU,"Software\Microsoft\Avalon.Graphics",,,""
HKCU,"Software\Microsoft\Avalon.Graphics",DisableHWAcceleration,%REG_DWORD%, 1

;-------------------------------------------------------------------------------
[iegd_SoftwareDeviceSettings_tnc]
HKR,, InstalledDisplayDrivers, %REG_MULTI_SZ%, iegddis
HKR,, MultiFunctionSupported, %REG_MULTI_SZ%, 1
HKR,, VgaCompatible, %REG_DWORD%, 0
HKR,, PcfVersion,    %REG_DWORD%, 0x0700

HKR,, ConfigId, %REG_DWORD%, 1

HKR, ALL\1 , name, %REG_SZ%, "Atom_E6xx_13X7"
HKR, ALL\1\General , DisplayConfig, %REG_DWORD%, 1
HKR, ALL\1\General , DisplayDetect, %REG_DWORD%, 1
HKR, ALL\1\General , PortOrder, %REG_SZ%, "42000"
HKR, ALL\1\General , DxvaOptions, %REG_DWORD%, 1
HKR, ALL\1\Port\4\General , name, %REG_SZ%, "LVDS13x7"
HKR, ALL\1\Port\4\General , Rotation, %REG_DWORD%, 0
HKR, ALL\1\Port\4\General , CenterOff, %REG_DWORD%, 1
HKR, ALL\1\Port\4\General , Edid, %REG_DWORD%, 1
HKR, ALL\1\Port\4\General , EdidAvail, %REG_DWORD%, 3
HKR, ALL\1\Port\4\General , EdidNotAvail, %REG_DWORD%, 4
HKR, ALL\1\Port\4\FpInfo , bkltmethod, %REG_DWORD%, 1
HKR, ALL\1\Port\4\FpInfo , BkltT1, %REG_DWORD%, 60
HKR, ALL\1\Port\4\FpInfo , BkltT2, %REG_DWORD%, 200
HKR, ALL\1\Port\4\FpInfo , BkltT3, %REG_DWORD%, 200
HKR, ALL\1\Port\4\FpInfo , BkltT4, %REG_DWORD%, 50
```



```
HKR, ALL\1\Port\4\FpInfo , BkltT5, %REG_DWORD%, 400
HKR, ALL\1\Port\4\Dtd\1 , PixelClock, %REG_DWORD%, 72300
HKR, ALL\1\Port\4\Dtd\1 , HorzActive, %REG_DWORD%, 1366
HKR, ALL\1\Port\4\Dtd\1 , HorzSync, %REG_DWORD%, 48
HKR, ALL\1\Port\4\Dtd\1 , HorzSyncPulse, %REG_DWORD%, 32
HKR, ALL\1\Port\4\Dtd\1 , HorzBlank, %REG_DWORD%, 160
HKR, ALL\1\Port\4\Dtd\1 , VertActive, %REG_DWORD%, 768
HKR, ALL\1\Port\4\Dtd\1 , VertSync, %REG_DWORD%, 3
HKR, ALL\1\Port\4\Dtd\1 , VertSyncPulse, %REG_DWORD%, 5
HKR, ALL\1\Port\4\Dtd\1 , VertBlank, %REG_DWORD%, 22
HKR, ALL\1\Port\4\Dtd\1 , Flags, %REG_DWORD%, 0x20000
HKR, ALL\1\Port\4\Attr , 27, %REG_DWORD%, 0
HKR, ALL\1\Port\4\Attr , 26, %REG_DWORD%, 18
HKR, ALL\1\Port\4\Attr , 60, %REG_DWORD%, 1
HKR, ALL\1\Port\4\Attr , 70, %REG_DWORD%, 100
HKR, ALL\1\Port\4\Attr , 71, %REG_DWORD%, 20300
HKR, ALL\1\Port\2\General , name, %REG_SZ%, "SDVOB"
HKR, ALL\1\Port\2\General , Rotation, %REG_DWORD%, 0
HKR, ALL\1\Port\2\General , Edid, %REG_DWORD%, 1
HKR, ALL\1\Port\2\General , EdidAvail, %REG_DWORD%, 3
HKR, ALL\1\Port\2\General , EdidNotAvail, %REG_DWORD%, 4
HKR, ALL\1\Port\2\General , CenterOff, %REG_DWORD%, 1

HKR, ALL\2 , name, %REG_SZ%, "Atom_E6xx_10X7"
HKR, ALL\2\General , DisplayConfig, %REG_DWORD%, 1
HKR, ALL\2\General , DisplayDetect, %REG_DWORD%, 1
HKR, ALL\2\General , PortOrder, %REG_SZ%, "42000"
HKR, ALL\2\General , DxvaOptions, %REG_DWORD%, 1
HKR, ALL\2\Port\4\General , name, %REG_SZ%, "LVDS10x7"
HKR, ALL\2\Port\4\General , Rotation, %REG_DWORD%, 0
HKR, ALL\2\Port\4\General , CenterOff, %REG_DWORD%, 1
HKR, ALL\2\Port\4\General , Edid, %REG_DWORD%, 1
HKR, ALL\2\Port\4\General , EdidAvail, %REG_DWORD%, 3
HKR, ALL\2\Port\4\General , EdidNotAvail, %REG_DWORD%, 4
HKR, ALL\2\Port\4\FpInfo , bkltmethod, %REG_DWORD%, 1
HKR, ALL\2\Port\4\FpInfo , BkltT1, %REG_DWORD%, 60
HKR, ALL\2\Port\4\FpInfo , BkltT2, %REG_DWORD%, 200
HKR, ALL\2\Port\4\FpInfo , BkltT3, %REG_DWORD%, 200
HKR, ALL\2\Port\4\FpInfo , BkltT4, %REG_DWORD%, 50
HKR, ALL\2\Port\4\FpInfo , BkltT5, %REG_DWORD%, 400
HKR, ALL\2\Port\4\Dtd\1 , PixelClock, %REG_DWORD%, 65000
HKR, ALL\2\Port\4\Dtd\1 , HorzActive, %REG_DWORD%, 1024
HKR, ALL\2\Port\4\Dtd\1 , HorzSync, %REG_DWORD%, 24
HKR, ALL\2\Port\4\Dtd\1 , HorzSyncPulse, %REG_DWORD%, 136
HKR, ALL\2\Port\4\Dtd\1 , HorzBlank, %REG_DWORD%, 320
HKR, ALL\2\Port\4\Dtd\1 , VertActive, %REG_DWORD%, 768
HKR, ALL\2\Port\4\Dtd\1 , VertSync, %REG_DWORD%, 3
HKR, ALL\2\Port\4\Dtd\1 , VertSyncPulse, %REG_DWORD%, 6
HKR, ALL\2\Port\4\Dtd\1 , VertBlank, %REG_DWORD%, 38
HKR, ALL\2\Port\4\Dtd\1 , Flags, %REG_DWORD%, 0x20000
HKR, ALL\2\Port\4\Attr , 27, %REG_DWORD%, 0
HKR, ALL\2\Port\4\Attr , 26, %REG_DWORD%, 18
HKR, ALL\2\Port\4\Attr , 60, %REG_DWORD%, 1
HKR, ALL\2\Port\4\Attr , 70, %REG_DWORD%, 100
HKR, ALL\2\Port\4\Attr , 71, %REG_DWORD%, 20300
HKR, ALL\2\Port\2\General , name, %REG_SZ%, "SDVOB"
HKR, ALL\2\Port\2\General , Rotation, %REG_DWORD%, 0
HKR, ALL\2\Port\2\General , Edid, %REG_DWORD%, 1
HKR, ALL\2\Port\2\General , EdidAvail, %REG_DWORD%, 3
HKR, ALL\2\Port\2\General , EdidNotAvail, %REG_DWORD%, 4
HKR, ALL\2\Port\2\General , CenterOff, %REG_DWORD%, 1

HKR,, No_D3D, %REG_DWORD%, 0

HKR,, PortDrivers, %REG_SZ%, "sdvo lvds"
HKR, ALL\1\General, DxvaOptions, %REG_DWORD%, 0x00000001
```

```
;------------------------------------------------------------------------------
[iegd_ICDSoftwareSettings]
HKLM, "SOFTWARE\Microsoft\Windows NT\CurrentVersion\OpenGLDrivers\iegddis", DLL,
%REG_SZ%, iegdglga
HKLM, "SOFTWARE\Microsoft\Windows NT\CurrentVersion\OpenGLDrivers\iegddis",
DriverVersion, %REG_DWORD%, 0x00000001
HKLM, "SOFTWARE\Microsoft\Windows NT\CurrentVersion\OpenGLDrivers\iegddis", Flags,
%REG_DWORD%, 0x00000001
HKLM, "SOFTWARE\Microsoft\Windows NT\CurrentVersion\OpenGLDrivers\iegddis",
Version, %REG_DWORD%, 0x00000002

;==============================================================================
[Strings]

;------------------------------------------------------------------------
; Localizable Strings
;------------------------------------------------------------------------
Intel="Intel Corporation"
DiskDesc="Embedded Installation"

i900G0="US15 Intel® Embedded Media and Graphics Driver Function 0"
iTNC0="Atom™ E6xx Intel® Embedded Media and Graphics Driver Function 0"
iTNC1="Atom™ E6xx Intel® Embedded Media and Graphics Driver Extension"

;------------------------------------------------------------------------
; Non Localizable Strings
;------------------------------------------------------------------------
SERVICE_BOOT_START     = 0x0
SERVICE_SYSTEM_START   = 0x1
SERVICE_AUTO_START     = 0x2
SERVICE_DEMAND_START   = 0x3
SERVICE_DISABLED       = 0x4

SERVICE_KERNEL_DRIVER  = 0x1

SERVICE_ERROR_IGNORE   = 0x0; Continue on driver load fail
SERVICE_ERROR_NORMAL   = 0x1; Display warn, but continue
SERVICE_ERROR_SEVERE   = 0x2; Attempt LastKnownGood
SERVICE_ERROR_CRITICAL = 0x3; Attempt LastKnownGood, BugCheck

REG_EXPAND_SZ = 0x00020000
REG_MULTI_SZ  = 0x00010000
REG_DWORD     = 0x00010001
REG_SZ        = 0x00000000
```

§ §

Appendix B Port Driver Attributes

B.1 Standard Port Driver Attributes

Port drivers are modules within the Intel® Embedded Media and Graphics Driver that control SCH-specific modules such as SCH LVDS, SCH TV or add-on modules to SCH. The table below lists the attributes available to port drivers. Some of these standard attributes can be customized for specific port drivers and are detailed in the following sections of this appendix.

In the following tables, device-specific (non-standard) attributes are highlighted in gray.

- "Internal LVDS Port Driver Attributes (Mobile chipsets only)" on page 182
- "HDMI Port Driver Attributes" on page 183
- "Chrontel CH7307 Port Driver Attributes" on page 184
- "Chrontel CH7308 Port Driver Attributes" on page 184
- "Chrontel CH7315/CH7319/CH7320 Port Driver Attributes" on page 185
- "Chrontel CH7317b Port Driver Attributes" on page 185
- "Chrontel CH7022 Port Driver Attributes" on page 185
- "Chrontel CH7036 Port Driver Attributes" on page 187
- "Silicon Image SiI 1362/SiI 1364 Port Driver DVI Attributes" on page 187
- "LAPIS Semiconductor* ML7213" on page 188
- "STM* IOH ConneXt" on page 188

Note: Not all standard attributes are supported by all port drivers. Please see the following sections for details on the specific attributes supported by each port driver. Flat panel settings are specified via the FPINFO options of the configuration; please see Table 21, "Parameter Configuration Format" on page 58.

Table 34. Standard Port Driver Attributes (Sheet 1 of 3)

Attribute Name	Attribute ID Number	Description
BRIGHTNESS	0	Brightness adjustment.
CONTRAST	1	Contrast adjustment.
HUE	2	Hue adjustment.
FLICKER	3	Setting to reduce flicker.
HPOSITION	4	Controls the horizontal position of the display.
VPOSITION	5	Controls the vertical position of the display.
HSCALE	6	Horizontal scaling ratio.
VSCALE	7	Vertical scaling ratio.
TVFORMAT	8	TV formats are device-specific.

Table 34. Standard Port Driver Attributes (Sheet 2 of 3)

Attribute Name	Attribute ID Number	Description
DISPLAY TYPE	9	Allows selection of different displays for multi-display devices. This attribute is device-specific.
LUMA FILTER	10	TV Luma Filter adjustment.
CHROMA FILTER	11	Chroma Filter adjustment.
TEXT FILTER	12	Text Filter adjustment.
TV OUTPUT TYPE	14	TV output types. This attribute is device-specific.
SATURATION	15	Saturation adjustment.
PANEL FIT	18	Panel fitting. Yes or no.
SCALING RATIO	19	Output Scaling. Device-specific.
FP BACKLIGHT ENABLE	20	Enable flat panel backlight.
PANEL DEPTH	26	Can be either 18 or 24. 18 specifies 6-bit output per color, 24 specifies 8-bit output per color.
DUAL CHANNEL PANEL	27	Is it a dual channel panel or not? Takes 0 or 1.
GANG MODE	28	For achieving a Gang mode output using two digital ports.
GANG MODE EVEN ODD	29	Gang display even or odd. This attribute is to be set along with Gang mode (28). This mode (Gang Mode Even Odd) puts even pixels on one digital port and odd pixels on the other, and needs to be selected based on the display panel used.
SHARPNESS	31	Sharpness.
HWCONFIG	32	Hardware Configuration for sDVO encoders that support multiple configurations.
HORZFILTER	33	Horizontal Filter.
VERTFILTER	34	Vertical Filter.
FRAME BUFFER GAMMA	35	Framebuffer gamma correction.
FRAME BUFFER BRIGHTNESS	36	Framebuffer brightness.
FRAME BUFFER CONTRAST	37	Framebuffer contrast.
2D FLICKER	39	Two-dimension flicker.
ADAPTIVE FLICKER	40	Adaptive flicker.
HORIZONTAL OVERSCAN	41	Horizontal overscan.
VERTICAL OVERSCAN	42	Vertical overscan.
SPREAD SPECTRUM CLOCKING	43	Spectrum Clocking
DOT_CRAWL	44	Dot crawl affects the edges of color and manifests itself as moving dots of color along these edges.
DITHER	45	Dither setting
PANEL PROTECT HSYNC	46	Horizontal sync panel protection
PANEL PROTECT VSYNC	47	Vertical sync panel protection
PANEL PROTECT PIXCLK	48	Pixel clock protection
LVDS PANEL TYPE	49	This is used to select SPWG vs. OpenLDI panel types. 0 = SPWG; 1 = OpenLDI.
VGA 2X IMAGE	57	Controls VGA image in Gang mode.
TEXT ENHANCEMENT	58	Controls text tuning.

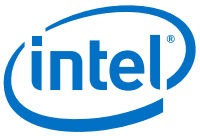


Table 34. Standard Port Driver Attributes (Sheet 3 of 3)

Attribute Name	Attribute ID Number	Description
MAINTAIN ASPECT RATIO	59	This controls scaled image to match source image aspect ratio or full screen image.
FIXED TIMING	60	This indicates whether the attached display is a fixed timing display.
INTENSITY	70	This attribute provides a method to control the backlight intensity. It is not a method to turn on backlight but provides a way to adjust its value in percentages from 0% to 100%

B.2 Port Driver Attributes

This section provides the supported attributes for each of the port drivers.

B.2.1 Internal LVDS Port Driver Attributes (Mobile chipsets only)

Table 35. Internal LVDS Port Driver Attributes (Sheet 1 of 2)

Attribute Name	Attribute ID	Description	Possible Ranges
PANELDEPTH	26	Specify Panel Depth based on connected panel.	Default is 18, however, on some SCH chipsets 24-bit also is supported. For example,US15W supports both 18 and 24-bit outputs.
DUALCHANNEL	27	Single or Dual Channel Panel	0 = Single 1 = Dual Default is 0.
SPREAD SPECTRUM CLOCKING	43	Spectrum Clocking	3-9 for US15W 0-15 for other chipsets Default = 7 Step = 1 ***Notes:*** This setting changes the EMI characteristics, which can be measured with tuning equipment. The change will not necessarily be visible in the display. For US15W/WP/WPT, when Spread Spectrum Clocking (SSC) setting is enabled, the 100Mhz input clock is selected. When SSC disabled, 96MHz input clock is selected.
DITHER	45	On and off Dithering	Dither=0 for 24-bit panels Dither=1 for 18-bit panels Default: • dither = 1 for 18-bit panels • dither = 0 for 24-bit panels.

Table 35. Internal LVDS Port Driver Attributes (Sheet 2 of 2)

Attribute Name	Attribute ID	Description	Possible Ranges
LVDS PANEL TYPE	49	LVDS panel connector.	0 = SPWG formatted LVDS output (default) 1 = OpenLDI unbalanced color mapping output Default = 0
FIXED TIMING	60	This indicates whether attached display is a fixed timing display.	0 = on 1 = off
INTENSITY	70	This attribute provides a method to control the backlight intensity. It is not a method to turn on backlight but provides a way to adjust its value in percentages from 0% to 100%	Valid range is 0-100%. Default is 100.
INVERTER FREQUENCY	71	A method of controlling the backlight. It determines the number of time base events in total for a complete cycle of modulated backlight control.	Valid range is 0-65535 Hz. Typical value is 300 – 1000. Default depends on the board. For Atom E6xx, default is 20300. For most boards, default is 0. ***Note:*** For Atom E6xx platforms provided by Intel to EA customers, the inverter frequency and intensity needs to be set.
BACKLIGHT LEGACY MODE	72	A method for controlling whether to use legacy mode for PWM duty cycle. Legacy mode is where the PWM duty cycle will be calculated using a combination of Backlight duty cycle and Legacy backlight Control (LBPC). In non-legacy mode, it will be calculated using Backlight duty cycle only.	Valid values are 0 for non-legacy mode or 1 for legacy mode. Default is 0.

B.2.2 HDMI Port Driver Attributes

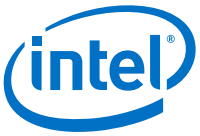
B.2.2.1 Audio

The Intel® EMGD package does not include an HDMI audio driver, so you must obtain and install the driver yourself. The HDMI audio driver needs to support Intel HD Audio to be compatible with Intel® EMGD. You must also obtain Microsoft patch KB888111 to enable HDMI audio. Intel® EMGD supports only the Windows* HDMI audio driver.

B.2.2.2 sDVO-HDMI (CH7315)

Intel® EMGD supports only one type of sDVO-HDMI encoder, which is CH7315. sDVO-B cannot coexist with HDMI-B; sDVO-C cannot coexist with HDMI-C.

sDVO takes precedence over the HDMI port driver. If no sDVO encoder is available HDMI is automatically loaded by default (only in the GM45 Express chipset).



B.2.2.3 Internal HDMI

Internal HDMI is available only for the GM45 Express chipset. Only one HDMI port has audio at any one time. The first port in the port order has audio while the second port would have only display without audio.

Only one HDMI port has HDCP at any one time. The first port to receive a request for HDCP has HDCP enabled only in that port.

B.2.2.4 HDCP

HDCP is supported through the Certified Output Protection Protocol* (COPP) interface in Windows.

B.2.3 Chrontel CH7307 Port Driver Attributes

The table below shows the attributes for the Chrontel CH7307* port driver.

Note: For flat panel backlight timing settings, please see Table 21, “Parameter Configuration Format” on page 58.

Table 36. Chrontel CH7307 Port Driver Attributes

Attribute Name	Attribute ID	Description	Possible Ranges
SPREAD SPECTRUM CLOCKING	43	Spectrum clocking	0-15 Default = 0 Step = 1
FIXED TIMING	60	This indicates whether attached display is a fixed timing display.	0 = off 1 = on

B.2.4 Chrontel CH7308 Port Driver Attributes

The table below shows the attributes for the Chrontel CH7308* port driver.

Note: For FPINFO panel width, height, and backlight timing settings, please see Table 21, “Parameter Configuration Format” on page 58.

Table 37. Chrontel CH7308 Port Driver Attributes (Sheet 1 of 2)

Attribute Name	Attribute ID	Description	Possible Ranges
LVDS COLOR DEPTH	26	Panel depth	18 = 18 bits 24 = 24 bits Default = 18
DUAL_CHANNEL	27	Dual-channel pane	Default - 0
SPREAD SPECTRUM CLOCKING	43	Spectrum Clocking	0-15 Default = 7 Step = 1
DITHER	45	Dither setting	Default = 0
HSYNC PANEL PROTECTION	46	Horizontal sync panel protection	Default = 0
VSYNC PANEL PROTECTION	47	Vertical sync panel protection	Default = 0
PIXEL CLOCK PROTECTION	48	Pixel clock protection	Default = 0

Table 37. Chrontel CH7308 Port Driver Attributes (Sheet 2 of 2)

Attribute Name	Attribute ID	Description	Possible Ranges
LVDS PANEL TYPE	49	LVDS panel connector.	0 = SPWG formatted LVDS output (default) 1 = OpenLDI unbalanced color mapping output Default = 0
TEXT ENHANCEMENT	58	Controls text tuning.	0-4.
FIXED TIMING	60	This indicates whether attached display is a fixed timing display.	0 = off 1 = on

B.2.5 Chrontel CH7315/CH7319/CH7320 Port Driver Attributes

Note: For flat panel backlight timing settings, please see Table 21, "Parameter Configuration Format" on page 58.

B.2.6 Chrontel CH7317b Port Driver Attributes

The table below shows the attributes for the Chrontel CH7317b port driver.

Table 38. Chrontel CH7317b Port Driver Attributes

Attribute Name	Attribute ID	Description	Possible Ranges
VGA BYPASS	9	Enables VGA bypass. To enable VGA Bypass, this configuration setting line must exist in the configuration file with the value of 2. Attribute 9 is used to enable selection of several possible display types based on what was supported on an sDVO device as defined in sDVO specifications. Default value of 2 represent VGA display.	1) Enable VGA Bypass

B.2.7 Chrontel CH7022 Port Driver Attributes

The table below shows the attributes for the Chrontel CH7022 port driver.

Table 39. Chrontel CH7022 Port Driver Attributes (Sheet 1 of 2)

Attribute Name	Attribute ID	Description	Possible Ranges
DISPLAY TYPE	9	Allows selection of different displays for multi-display devices. This attribute is device-specific. ***Note:*** TV Out is not available with VBIOS.	1) VGA Bypass (2) 2) Composite (4) 3) S-Video (8) 4) YPrPb (16)
BRIGHTNESS	0	Brightness adjustment.	0-255
SATURATION	15	Saturation adjustment.	0-127

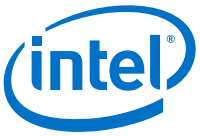


Table 39. Chrontel CH7022 Port Driver Attributes (Sheet 2 of 2)

Attribute Name	Attribute ID	Description	Possible Ranges
HUE	2	Hue adjustment.	0-127
CONTRAST	1	Contrast adjustment.	0-127
HORIZONTAL OVERSCAN	41	Horizontal overscan.	0-47
VERTICAL OVERSCAN	42	Vertical overscan.	0-47
VERTICAL POSITION/ VPOSITION	5	Controls the vertical position of the display.	0-1023
SHARPNESS	31	Sharpness.	0-7
TV CHROMA FILTER	11	ChromaFilter adjustment.	0-3
TV LUMA FILTER	10	TV Luma Filter adjustment.	0-2
ADAPTIVE FLICKER FILTER	40	Adaptive flicker.	0-7
DOT CRAWL	44	Dot crawl affects the edges of color and manifests itself as moving dots of color along these edges.	1) Have Dot Crawl Run Freely (0) 2) Freeze Dot Crawl (1)
TV OUTPUT FORMAT	8	TV formats are device-specific.	Refer to the Attributes Page for the complete list of choices.
ANALOG SOURCE	52	VGA	1) No Data (0) 2) Analog Source (1) 3) Pre-recorded Packaged (2) 4) Not Analog Pre-recorded (3)
SCAN INFORMATION	53	TV attributes are device specific.	1) No Data (0) 2) Overscanned (1) 3) Under scanned (2)
PICTURE ASPECT RATIO	54	The relative horizontal and vertical sizes.	1) No Data (0) 2) 4:3 (1) 3) 16:9 (2)
ACTIVE FORMAT RATIO	55	Output ratio.	1) No Data (0) 2) Active Format (1) 3) Square Pixels(8) 4) 4:3 Center (9) 5) 16:9 Center (10) 6) 14:9 Center (11) 7) 16:9 Letterbox (Top)(2) 8) 14:9 Letterbox (Top)(3) 9) 16:9 Letterbox (Center) 10) 4:3 (with shoot and protect 14:9 center) 11) 16:9 (with shoot and protect 14:9 center) (10610) 12) 16:9 (with shoot and protect 4:3 center)

B.2.8 Chrontel CH7036 Port Driver Attributes

The table below shows the attributes for the Chrontel CH7036 port driver.

Table 40. Chrontel CH7036 Port Driver Attributes

Attribute Name	Attribute ID	Description	Possible Ranges
H POSITION	4	Horizontal position adjustment for VGA output	0-4096 2048 (Default)
V POSITION	5	Vertical position adjustment for VGA output	0-4096 2048 (Default)
H SCALE	6	Horizontal display image size adjustment. Does not apply if video output is bypassed.	1 - 20 15 (HDMI Default) ***Note:*** If CRT is detected, CRT Default (18) is used.
V SCALE	7	Vertical display image size adjustment. Do not apply if video output is bypass	1 - 20 15 (HDMI Default) ***Note:*** If CRT is detected, CRT Default (18) is used.
Display Channel	9	Display output channel selection.	Possible Range: 1- Auto-Detect (Default) 2 -LVDS+HDMI 3- LVDS+DVI 4- LVDS+VGA 5- HDMI 6- DVI 7- VGA
DITHER SEL	45	Enable CH7036 LVDS panel dithering function if QUALITY ENHANCEMENT is disabled.	0: for 18-bit input -> 18-bit panel (default) 1: for 18-bit input -> 24-bit panel 2: for 24-bit input -> 18-bit panel 3: for 24-bit input -> 24-bit panel
AUDIO TYPE	89	Input audio format select	0 - SPDIF (Default) 1 - I2S
QUALITY ENHANCEMENT	93	CH7036 LVDS dithering bypass function enable. If enabled, DITHER SEL is ignored.	0 - Disable 1 - Enable (Default)

B.2.9 Silicon Image SiI 1362/SiI 1364 Port Driver DVI Attributes

Note: For flat panel backlight timing settings, please see Table 21, “Parameter Configuration Format” on page 58.

B.2.10 LAPIS Semiconductor* ML7213

Attribute Name	Attribute ID Number	Description	Possible Ranges
DITHER	45	Dither setting	Default = 0
PIXEL CLOCK PROTECTION	48	Pixel clock protection	Default = 0
RB SWAP	73	Red Blue color swap for LAPIS RGB panel	Enable = 1 Disable = 0 Default = 0

B.2.11 STM* IOH ConneXt

Attribute Name	Attribute ID Number	Description	Possible Ranges
DISPLAY TYPE	9	Allows selection of different displays for multi-display devices. This attribute is device-specific. ***Note:*** TV Out is not available with VBIOS.	1) VGA Bypass (2) 2) Composite (4) 3) S-Video (8) 4) YPrPb (16)
LVDS PANEL TYPE	49	LVDS panel connector.	0 = SPWG formatted LVDS output (default) 1 = OpenLDI unbalanced color mapping output Default = 0

B.3 Chipset and Port Driver-specific Installation Information

B.4 Default Search Order

Note: See more information pertaining to port order in the description for Section , “Port Devices (Available Ports, Port Order)” on page 28.

Table 41. Default Search Order

Chipset	Default Search Order
Intel® Atom™ Processor E6xx	LVDS, sDVOB
Intel® US15W/US15WP/WPT	LVDS, sDVOB

B.5 Default GPIO Pin Pair Assignments

Table 42. Default GPIO Pin Pair Assignments

Chipset	Default GPIO Pin Pair for EDID	
	sDVOB	LVDS
Intel® Atom™ Processor E6xx	4	2
Intel® US15W/WP/WPT	4	2

B.6 Default I2C Device Address Byte Assignment

Table 43. Default I²C Device Address Byte Assignment

Port Driver	Default Device Address Bytes (DAB)
CH7315, CH7317b, CH7319, CH7320, CH7022	0x70 (for first sDVO device) 0x72 (for second sDVO device)
CH7307	0x70 (for first sDVO device) 0x72 (for second sDVO device)
CH7308	0x70 (for first sDVO device) 0x72 (for second sDVO device)
SiI 1362	0x70 (for first sDVO device) 0x72 (for second sDVO device)
SiI 1364	0x70 (for first sDVO device) 0x72 (for second sDVO device)

§ §

Appendix C Intel® 5F Extended Interface Functions

The BIOS provides a set of proprietary function calls to control operation of the extended features. These function calls all use AH = 5Fh in their designed interface for easy identification as a proprietary function.

These functions are designed to maintain maximum compatibility with the Desktop and Mobile Video BIOS. As such many of the definitions behave identically. When the behavior of the Embedded Video BIOS is not identical to the Desktop and Mobile Video BIOS it is noted.

In addition to these 5F functions, the Video BIOS also supports all 4F functions defined by the *VESA BIOS Extension (VBE) Core Functions Standard, Version 3.0* with the exception of the 0A function (Return VBE Protected Mode Interface). All other functions, from 00 through 09 and 0B are supported by the Video BIOS. The *VESA BIOS Extension (VBE) Core Functions Standard, Version 3.0* document is available from http://www.vesa.org/vesa-standards/free-standards/

The table below provides a summary of the Intel® EMGD supported Intel 5F functions.

Table 44. Summary of Intel 5F Extended Interface Functions (Sheet 1 of 2)

Function	Function Name	Description	Page
BIOS Extended Interface Functions			
5F01h	Get Video BIOS Information	Gets VBIOS Build Information.	191
5F05h	Refresh Rate	Sets a new vertical refresh rate for a given mode and returns the current vertical refresh rate	191
5F10h	Get Display Memory Information	Returns information about the linear memory.	193
5F1Ch	BIOS Pipe Access	Sets the BIOS pipe access and returns the BIOS pipe access status.	193
5F29h	Get Mode Information	Returns information on the requested mode.	194
5F61h	Local Flat Panel Support Function	Supports local flat panel features.	194
5F68h	System BIOS Callback	Allows SoftBIOS to do any system callbacks through INT 15h	195
Hooks for the System BIOS			
5F31h	POST Completion Notification Hook	Signals the completion of video POST (Power On Self Test)	195
5F33h	Hook After Mode Set	Allows System BIOS to intercept Video BIOS at the end of a mode set.	196
5F35h	Boot Display Device Hook	Allows System BIOS to override video display default setting.	196
5F36h	Boot TV Format Hook	Allows System BIOS to boot TV in selected TV format state.	197

Table 44. Summary of Intel 5F Extended Interface Functions (Sheet 2 of 2)

Function	Function Name	Description	Page
5F38h	Hook Before Set Mode	Allows System BIOS to intercept Video BIOS before setting the mode.	197
5F40h	Config ID Hook	Allows System BIOS to supply a configuration ID that is passed to the driver.	198

C.1 BIOS Extended Interface Functions

The BIOS provides a set of proprietary function calls to control operation of the extended features. These function calls all use AH = 5Fh in their designed interface for easy identification as a proprietary function

These functions are designed to maintain maximum compatibility with the Desktop and Mobile Video BIOS. As such many of the definitions behave identically. When the behavior of the Embedded Video BIOS is not identical to the Desktop and Mobile Video BIOS it is noted.

C.1.1 5F01h – Get Video BIOS Information

This function returns the Video BIOS Build information.

Note: This function is an extension of the Desktop and Mobile Video BIOS. If register ECX does not contain ASCII characters “IEGD” then the VBIOS is not described by this specification.

Calling Register:

AX = 5F01h, Get Video Information function

Return Registers:

AX = Return Status (function not supported if AL != 5Fh):
= 005Fh, Function supported and successful
= 015Fh, Function supported but failed
EBX = 4 bytes Video BIOS Build Number ASCII string, e.g., ‘1000’
ECX = 4 bytes Embedded Identifier, ASCII string ‘IEGD’

C.1.2 5F05h – Refresh Rate

This function sets a new vertical refresh rate for a given mode and returns the current vertical refresh rate and available refresh rate for a given non-VGA mode.

C.1.2.1 5F05h, 00h – Set Refresh Rate

This sub-function sets a new default refresh rate for the selected pipe. If the mode is currently active, the CRT controller and other registers will be automatically programmed setting the requested refresh rate.

Note: This function is not entirely compatible with the Desktop and Mobile versions. It is not possible to set the refresh rate for a given mode in advance. This function sets the “desired” refresh rate which will be applied to all subsequent mode sets when possible. If the mode provided in BL is the current mode, then a mode change will be automatically performed.

Calling Register:

AX = 5F05h, Refresh Rate function
BH = 00h, Set Refresh Rate sub-function
BL = Mode Number
ECX = Refresh rate (indicated by setting one bit):
Bits 31 - 9 = Reserved
Bit 8 = 120 Hz
Bit 7 = 100 Hz
Bit 6 = 85 Hz
Bit 5 = 75 Hz
Bit 4 = 72 Hz
Bit 3 = 70 Hz
Bit 2 = 60 Hz
Bit 1 = 56 Hz
Bit 0 = 43 Hz (Interlaced - Not supported)

Return Registers:

AX = Return Status (function not supported if AL != 5Fh):
= 005Fh, Function supported and successful
= 015Fh, Function supported but failed

C.1.2.2 5F05h, 01h – Get Refresh Rate

This sub-function returns current vertical refresh rate for the selected pipe and available refresh rates information for a given Non-VGA mode.

Note: This sub-function returns a status of supported but failed (AX = 015Fh) if executed with a standard VGA mode.

Calling Registers:

AX = 5F05h, Refresh Rate function
BH = 01h, Get Refresh Rate sub-function
BL = Mode number

Return Registers:

AX = Return Status (function not supported if AL != 5Fh):
= 005Fh, Function supported and successful
= 015Fh, Function supported but failed
EBX = Available refresh rates (indicated by one or more bits set):
Bits 31 - 9 = Reserved
Bit 8 = 120 Hz
Bit 7 = 100 Hz
Bit 6 = 85 Hz
Bit 5 = 75 Hz
Bit 4 = 72 Hz
Bit 3 = 70 Hz
Bit 2 = 60 Hz
Bit 1 = 56 Hz
Bit 0 = 43 Hz (Interlaced - Not supported)
ECX = Current refresh rate (see EBX for bit definitions)

C.1.3 5F10h – Get Display Memory Information

This function returns information regarding the linear memory starting address, size and memory mapped base address.

Calling Register:

AX = 5F10h, Get Linear Display Memory Information function

Return Registers:

AX = Return Status (function not supported if AL != 5Fh):
= 005Fh, Function supported and successful
= 015Fh, Function supported but failed
ESI = Display memory base address
ECX = Total physical display memory size (in bytes)
EDX = Available display memory size (in bytes)
EDI = Memory Mapped I/O Base Address
EBX = Stride (memory scan line width in bytes)

C.1.4 5F1Ch – BIOS Pipe Access

This function will set the BIOS pipe access or return the BIOS pipe access status.

C.1.4.1 5F1Ch, 00h – Set BIOS Pipe Access

This sub-function will set the currently selected pipe. All 5f functions operate on the currently selected pipe.

When not in clone modes this value cannot be set.

Calling Registers:

AX = 5F1Ch, BIOS Pipe Access function
BH = 00h, Set BIOS Pipe Access sub-function
CH = BIOS Pipe access:
= 00h, Pipe A
= 01h, Pipe B

Return Registers:

AX = Return Status (function not supported if AL != 5Fh):
= 005Fh, Function supported and successful
= 015Fh, Function supported but failed

C.1.4.2 5F1Ch, 01h – Get BIOS Pipe Access

This sub-function will return the currently selected pipe.

Calling Registers:

AX = 5F1Ch, BIOS Pipe Access function
BH = 01h, Get BIOS Pipe Access sub-function

Return Registers:

AX = Return Status (function not supported if AL != 5Fh):
= 005Fh, Function supported and successful
= 015Fh, Function supported but failed
CH = BIOS Pipe access:
= 00h, Pipe A
= 01h, Pipe B

C.1.5 5F29h – Get Mode Information

This function returns the requested mode's resolution, color depth, and maximum required bandwidth using its current refresh rate. This function is applied to extended-graphics modes only. If the mode number is not an extended graphics mode, the function will return failure.

Calling Registers:

AX = 5F29h, Get Mode Information function
BH = Mode To Use:
= 80h, Current Mode
= 00h - 7Fh, Given Mode Number

Return Registers:

AX = Return Status (function not supported if AL != 5Fh):
= 005Fh, Function supported and successful
= 015Fh, Function supported but failed
EBX bits 31 - 16 = Mode horizontal (X) resolution in pixels
EBX bits 15 - 0 = Mode vertical (Y) resolution in pixels
ECX bits 31 - 16 = Maximum bandwidth in megabytes per second
ECX bits 15 - 0 = Color depth in bits per pixel

C.1.6 5F61h – Local Flat Panel Support Function

This function supports local flat panel only features.

Note: Only Subfunctions 5h and 8h of the 5f61h interface are supported for the Embedded VBIOS.

C.1.6.1 5F61h, 05h – Get Configuration ID

This function is used to return the Configuration ID.

Note: This function is known as "Get Local Flat Panel Number" in the Desktop and Mobile Video BIOS. This function performs a similar purpose however, the configuration IDs have no pre-defined meaning. The Configuration ID is reported to the Intel® EMGD.

Calling Registers:

AX = 5F61h, Local Flat Panel Support function
BH = 05h, Get Config ID Subfunction

Return Registers:

AX = Return Status (function not supported if AL != 5Fh):
= 005Fh, Function supported and successful
= 015Fh, Function supported but failed
BL = Config ID

C.1.6.2 5F61h, 08h - Set the LVDS Backlight Level

This function is used to handle the LVDS backlight level.

Note: To enable to LVDS backlight control functionality, ensure you configure the following attributes in the CED Attribute Settings page:

- Intensity (attribute #70) to your desired value.
- Inverter Frequency (attribute #71) based on your requirement.
- Backlight Method (attribute #72) to `Legacy + PWM mode(1)`

Calling Registers:

AX = 5F61h, Local Flat Panel Support function
BH = 08h, Set the LVDS backlight level Subfunction
BL = 0 – 255 (backlight level)

Return Registers:

AX = Return Status (function not supported if AL != 5Fh):
= 005Fh, Function supported and successful
= 015Fh, Function supported but failed

C.1.7 5F68h – System BIOS Callback

This is a generic function that allows SoftBIOS to do any system callbacks through INT 15h. The Input/Output of this function is dependent on the definition of the desired INT 15h hook except for the EAX register.

Calling Registers:

AX = 5F68h, System BIOS Callback Function
EAX bits 31:16 = System BIOS INT 15h Hook Function

Return Registers:

AX = Return Status (function not supported if AL != 5Fh):
= 005Fh, Function supported and successful
= 015Fh, Function supported but failed

C.2 Hooks for the System BIOS

The video BIOS performs several system BIOS interrupt function calls (interrupt 15h hooks). Each function provides the system BIOS with the opportunity to gain control at specific times to perform any custom processing that may be required. After each interrupt hook, the system BIOS must return control to the video BIOS. INT 10h calls could be made within the INT 15h hook calls provided that it is not recursive and thus cause a deadlock.

C.2.1 5F31h – POST Completion Notification Hook

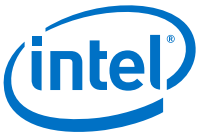
This hook signals the completion of video POST (Power On Self Test). The hook executes after the sign-on message is displayed and PCI BIOS resizing.

Calling Registers:

AX = 5F31h, POST Completion Notification Hook

Return Registers:

AX = Return Status (function not supported if AL != 5Fh):
= 015Fh, Function supported but failed
= 005Fh, Function supported and successful



C.2.2 5F33h – Hook After Mode Set

This hook allows the system BIOS to intercept the video BIOS at the end of a mode set.

Calling Registers:

AX = 5F33h, Hook After Mode Set
BH = Number of character columns
BL = Current mode number
CH = Active display page

Return Registers:

AX = Return Status (function not supported if AL != 5Fh):
= 015Fh, Function supported but failed
= 005Fh, Function supported and successful

C.2.3 5F35h – Boot Display Device Hook

This hook allows the system BIOS to override the video display default setting. The graphics BIOS will set the returned video display during POST (power up initialization).

Note: This function is not entirely compatible with the Desktop and Mobile Video BIOS. The bits in CL have a configurable mapping to the Port Numbers as defined in Section 4.0, "Video Firmware" on page 76. The assigned meanings used in the Desktop specification can be duplicated with a correct configuration. The values below are the default values if no "Common To Port" mapping is provided.

Calling Registers:

AX = 5F35h, Boot Display Device Hook

Return Registers:

AX = Return Status (function not supported if AL != 5Fh);
= 005Fh, Function supported and successful
= 015Fh, Function supported but failed
CL = Display Device Combination to boot (1 = Enable display,
0 = Disable display):

= 00h, VBIOS Default
Bit 7 - 6 = Reserved
Bit 5 = Port 5 (or common_to_port[5])
Bit 4 = Port 4 (or common_to_port[4])
Bit 3 = Port 3 (or common_to_port[3])
Bit 2 = Port 2 (or common_to_port[2])
Bit 1 = Port 1 (or common_to_port[1])
Bit 0 = Port 0 (or common_to_port[0])

C.2.4 5F36h – Boot TV Format Hook

This hook allows the system BIOS to boot TV in selected TV format state.

Calling Registers:

AX = 5F36h, Boot TV Format Hook

Return Registers:

AX = Return Status (function not supported if AL != 5Fh):
= 015Fh, Function supported but failed
= 005Fh, Function supported and successful
BL = TV Format requested:
= 00h, No Preference
= 01h, NTSC_M
= 11h, NTSC_M_J
= 21h, NTSC_433
= 31h, NTSC_N
= 02h, PAL_B
= 12h, PAL_G
= 22h, PAL_D
= 32h, PAL_H
= 42h, PAL_I
= 52h, PAL_M
= 62h, PAL_N
= 72h, PAL_60
= 03h, SECAM_L
= 13h, SECAM_L1
= 23h, SECAM_B
= 33h, SECAM_D
= 43h, SECAM_G
= 53h, SECAM_H
= 63h, SECAM_K
= 73h, SECAM_K1

C.2.5 5F38h – Hook Before Set Mode

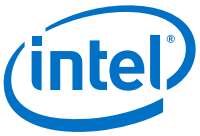
This hook allows the system BIOS to intercept the video BIOS before setting the mode.

Calling Registers:

AX = 5F38h, Hook Before Set Mode
CL = New video mode to be set

Return Registers:

AX = Return Status (function not supported if AL != 5Fh):
= 015Fh, Function supported but failed
= 005Fh, Function supported and successful



C.2.6 5F40h – Config ID Hook

This function is known as “Boot Panel Type Hook” in the Desktop and Mobile Video BIOS. It allows the system BIOS to supply a configuration ID that will eventually be passed to the driver. This configuration ID is unused by the Video BIOS; however, it alters the behavior of the driver as described in Section 4.0, “Video Firmware” on page 76.

Calling Registers:

AX = 5F40h, Config ID Hook

Return Registers:

AX = Return Status (function not supported if AL != 5Fh):
= 005Fh, Function supported and successful
= 015Fh, Function supported but failed
CL = Configuration ID

§ §

Appendix D 2D/3D API Support

This sectionprovides information on supported and non-supported OpenGL and OpenGL ES APIs. See Section 7.4.13, “OpenGL Support” on page 159 for additional information.

D.1 2D Support

Intel® EMGD provides 2D capabilities on Linux through XRender and OpenVG 1.1 and on Windows through DirectX*/GDI.

D.2 3D Support

Intel® EMGD provides 3D capabilities through several industry-standard APIs, such as OpenGL, OpenGL ES, Direct3D, and D3DMobile. These APIs are described in the following sections.

D.2.1 OpenGL APIs

The following OpenGL versions are supported:

- OGLES 1.1 and 2.0
- OGL 2.0 and 2.1 (Linux only)
- OpenVG 1.1 (Windows Embedded Compact 7 and Linux)
- EGL

For general OpenGL information, visit http://www.opengl.org/about/overview/.

Table 45. Supported Intel® OpenGL APIs for Windows (Sheet 1 of 3)

GL_APPLE_packed_pixels
GL_ARB_depth_texture
GL_ARB_draw_buffers
GL_ARB_fragment_program
GL_ARB_fragment_shader
GL_ARB_multisample
GL_ARB_multitexture
GL_ARB_pixel_buffer_object
GL_ARB_point_parameters
GL_ARB_point_sprite
GL_ARB_shader_objects
GL_ARB_shading_language_100
GL_ARB_shading_language_120
*Not supported on Intel US15W series chipsets.

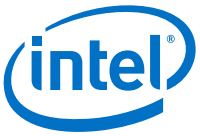


Table 45. Supported Intel® OpenGL APIs for Windows (Sheet 2 of 3)

GL_ARB_shadow
GL_ARB_texture_border_clamp
GL_ARB_texture_compression
GL_ARB_texture_cube_map
GL_ARB_texture_env_add
GL_ARB_texture_env_combine
GL_ARB_texture_env_crossbar
GL_ARB_texture_env_dot3
GL_ARB_texture_mirrored_repeat
GL_ARB_texture_non_power_of_two
GL_ARB_texture_rectangle
GL_ARB_transpose_matrix
GL_ARB_vertex_buffer_object
GL_ARB_vertex_program
GL_ARB_window_pos
GL_ATI_blend_equation_separate
GL_ATI_separate_stencil
GL_EXT_abgr
GL_EXT_bgra
GL_EXT_blend_color
GL_EXT_blend_equation_separate
GL_EXT_blend_func_separate
GL_EXT_blend_logic_op
GL_EXT_blend_minmax
GL_EXT_blend_subtract
GL_EXT_clip_volume_hint*
GL_EXT_compiled_vertex_array
GL_EXT_copy_texture
GL_EXT_cull_vertex
GL_EXT_draw_range_elements
GL_EXT_fog_coord
GL_EXT_framebuffer_blit
GL_EXT_framebuffer_object
GL_EXT_multi_draw_arrays
GL_EXT_packed_pixels
GL_EXT_paletted_texture
GL_EXT_pixel_buffer_object
GL_EXT_point_parameters
GL_EXT_polygon_offset
GL_EXT_rescale_normal
GL_EXT_secondary_color
*Not supported on Intel US15W series chipsets.

Table 45. Supported Intel® OpenGL APIs for Windows (Sheet 3 of 3)

GL_EXT_separate_specular_color
GL_EXT_shadow_funcs
GL_EXT_stencil_two_side*
GL_EXT_stencil_wrap
GL_EXT_subtexture
GL_EXT_texture
GL_EXT_texture_compression_s3tc
GL_EXT_texture_edge_clamp
GL_EXT_texture_env_add
GL_EXT_texture_env_combine
GL_EXT_texture_env_dot3
GL_EXT_texture_filter_anisotropic
GL_EXT_texture_lod_bias
GL_EXT_texture_mirror_clamp
GL_EXT_texture_object
GL_EXT_texture_rectangle
GL_EXT_texture3D
GL_EXT_vertex_array
GL_IBM_rasterpos_clip
GL_IBM_texture_mirrored_repeat
GL_MESA_window_pos
GL_MESA_ycbcr_texture
GL_NV_blend_square
GL_NV_light_max_exponent
GL_NV_point_sprite
GL_NV_texgen_reflection
GL_NV_texture_rectangle
GL_OES_read_format
GL_SGI_color_matrix
GL_SGIS_generate_mipmap
GL_SGIS_texture_border_clamp
GL_SGIS_texture_edge_clamp
GL_SGIS_texture_lod
GL_SUN_multi_draw_arrays
Mesa 3-D graphics library 7.1
GLX_ARB_get_proc_address
*Not supported on Intel US15W series chipsets.

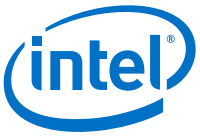


Table 46. Supported Intel® OpenGL APIs for Linux (Sheet 1 of 3)

GL_ARB_depth_texture
GL_ARB_draw_buffers
GL_ARB_fragment_program
GL_ARB_fragment_shader
GL_ARB_half_float_pixel
GL_ARB_matrix_palette
GL_ARB_multisample
GL_ARB_multitexture
GL_ARB_occlusion_query
GL_ARB_pixel_buffer_object
GL_ARB_point_parameters
GL_ARB_point_sprite
GL_ARB_shader_objects
GL_ARB_shading_language_100
GL_ARB_shadow
GL_ARB_shadow_ambient
GL_ARB_texture_border_clamp
GL_ARB_texture_compression
GL_ARB_texture_cube_map
GL_ARB_texture_env_add
GL_ARB_texture_env_combine
GL_ARB_texture_env_crossbar
GL_ARB_texture_env_dot3
GL_ARB_texture_float
GL_ARB_texture_mirrored_repeat
GL_ARB_texture_non_power_of_two
GL_ARB_texture_rectangle
GL_ARB_transpose_matrix
GL_ARB_vertex_blend
GL_ARB_vertex_buffer_object
GL_ARB_vertex_program
GL_ARB_vertex_shader
GL_ARB_window_pos
GL_EXT_abgr
GL_EXT_bgra
GL_EXT_blend_color
GL_EXT_blend_equation_separate
GL_EXT_blend_func_separate
GL_EXT_blend_minmax
*Note that `glinfo` will list the supported modes on the current installation.

Table 46. Supported Intel® OpenGL APIs for Linux (Sheet 2 of 3)

GL_EXT_blend_subtract
GL_EXT_compiled_vertex_array
GL_EXT_draw_range_elements
GL_EXT_fog_coord
GL_EXT_framebuffer_object
GL_EXT_multi_draw_arrays
GL_EXT_packed_pixels
GL_EXT_rescale_normal
GL_EXT_secondary_color
GL_EXT_separate_specular_color
GL_EXT_shadow_funcs
GL_EXT_stencil_two_side
GL_EXT_stencil_wrap
GL_EXT_texture_compression_s3tc
GL_EXT_texture_cube_map
GL_EXT_texture_edge_clamp
GL_EXT_texture_env_add
GL_EXT_texture_env_combine
GL_EXT_texture_env_dot3
GL_EXT_texture_filter_anisotropic
GL_EXT_texture_lod_bias
GL_EXT_texture_object
GL_EXT_texture3D
GL_EXT_vertex_array
GL_IMG_texture_compression_pvrtc
GL_NV_blend_square
GL_NV_texgen_reflection
GL_NV_texture_rectangle
GL_S3_s3tc
GL_SGIS_generate_mipmap
GLX_ARB_get_proc_address*
GLX_ARB_multisample*
GLX_EXT_import_context*
GLX_EXT_texture_from_pixmap
GLX_EXT_visual_info*
GLX_EXT_visual_rating*
GLX_MESA_allocate_memory*
GLX_MESA_copy_sub_buffer*
GLX_MESA_swap_control*
GLX_MESA_swap_frame_usage*
*Note that `glinfo` will list the supported modes on the current installation.

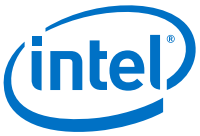


Table 46. Supported Intel® OpenGL APIs for Linux (Sheet 3 of 3)

GLX_OML_swap_method*
GLX_OML_sync_control*
GLX_SGI_make_current_read*
GLX_SGI_swap_control*
GLX_SGI_video_sync*
GLX_SGIS_multisample*
GLX_SGIX_fbconfig*
GLX_SGIX_pbuffer*
GLX_SGIX_visual_select_group*
*Note that `glinfo` will list the supported modes on the current installation.

Table 47. Non-Supported Intel® OpenGL APIs (Sheet 1 of 2)

GL_3DFX_texture_compression_FXT1
GL_ARB_color_buffer_float
GL_ARB_fragment_program_shadow
GL_ARB_occlusion_query
GL_ARB_vertext_shader
GL_EXT_cull_vertex (on Atom E6xx)
GLX_ARB_multisample
GLX_EXT_import_context
GLX_EXT_texture_from_pixmap
GLX_EXT_visual_info
GLX_EXT_visual_rating
GLX_MESA_allocate_memory
GLX_MESA_copy_sub_buffer
GLX_MESA_swap_control
GLX_MESA_swap_frame_usage
GLX_OML_swap_method
GLX_OML_sync_control
GLX_SGI_make_current_read
GLX_SGI_swap_control
GLX_SGI_swap_control
GLX_SGI_video_sync
GLX_SGIS_multisample
GLX_SGIX_fbconfig
GLX_SGIX_hyperpipe
GLX_SGIX_pbuffer
GLX_SGIX_swap_barrier
GLX_SGIX_visual_select_group
WGL_ARB_buffer_region
WGL_ARB_extensions_string

Table 47. Non-Supported Intel® OpenGL APIs (Sheet 2 of 2)

WGL_ARB_make_current_read
WGL_ARB_pbuffer
WGL_ARB_pixel_format
WGL_EXT_swap_control

D.2.2 OpenGL ES 1.1

The following chipsets support OpenGL ES 1.1:

- Intel® System Controller Hub US15W/US15WP/WPT chipset
- Intel® Atom™ Processor E6xx

Except where noted by individual chipsets, the following OpenGL ES 1.1 extensions are supported:

- GL_ARB_multisample
- GL_EXT_texture_filter_anisotropic
- GL_EXT_framebuffer_blit
- GL_IMG_texture_compression_pvrtc
- GL_OES_blend_equation_separate
- GL_OES_blend_func_separate
- GL_OES_blend_subtract
- GL_OES_byte_coordinates
- GL_OES_depth24
- GL_OES_depth32
- GL_OES_draw_texture
- GL_OES_element_index_uint
- GL_OES_fbo_render_mipmap
- GL_OES_fixed_point
- GL_OES_framebuffer_object
- GL_OES_mapbuffer
- GL_OES_matrix_get
- GL_OES_point_size_array
- GL_OES_point_sprite
- GL_OES_query_matrix
- GL_OES_read_format
- GL_OES_rgb8_rgba8
- GL_OES_single_precision
- GL_OES_texture_cube_map
- GL_OES_texture_env_crossbar
- GL_OES_texture_mirrored_repeat

D.2.3 OpenGL ES 2.0

The following chipsets support OpenGL ES 2.0:

- Intel® System Controller Hub US15W/US15WP/WPT chipset
- Intel® Atom™ Processor E6xx

Except where noted by individual chipsets, the following OpenGL ES 2.0 extensions are supported:

- GL_OES_depth_texture
- GL_OES_standard_derivatives
- GL_OES_texture_3D
- GL_OES_texture_npot
- GL_EXT_texture_type_2_10_10_10_REV
- GL_OES_compressed_paletted_texture
- GL_OES_packed_depth_stencil

Table 48. Non-Supported Intel® OpenGL ES APIs on US15W/WP/WPT

Non-Supported API Name(s)
GL_OES_stencil_wrap
GL_OES_compressed_ETC1_RGB8_texture
GL_OES_matrix_palette
GL_OES_EGL_image
GL_AMD_compressed_3DC_texture
GL_AMD_compressed_ATC_texture
GL_OES_texture_float
GL_OES_texture_half_float
GL_OES_texture_float_linear
GL_OES_texture_half_float_linear
GL_OES_vertex_half_float
GL_OES_vertex_type_10_10_10_2
GL_OES_fragment_precision_high

D.2.4 OpenVG 1.1

The following chipsets support OpenVG 1.1 on Microsoft Windows Embedded Compact 7 and Linux:

- Intel® System Controller Hub US15W/US15WP/WPT chipset
- Intel® Atom™ Processor E6xx

D.2.5 EGL

The following EGL functions are supported:

- EGL_NOK_image_shared
- EGL_KHR_image_base

§ §

Appendix E Using the AMI* Video BIOS Utility

E.1 Introduction

The AMI* Video BIOS Utility (AVBU*) is a command line utility that can extract, replace, insert, and delete PCI video option ROM modules from a ROM image file. This utility is used in conjunction with the Intel® Embedded Media and Graphics Driver (Intel® EMGD) CED utility. CED can be used to generate a customized video BIOS (VBIOS) ROM file that meets your specific display settings. AVBU can then be used to merge the VBIOS ROM image with the hardware platform system BIOS ROM image.

This utility is provided by American Megatrends Incorporated* (http://www.ami.com), and is designed to work only on the specific hardware platform listed below. For other platforms, please contact your hardware platform supplier. To obtain a copy of this utility, please visit http://www.ami.com/crownbayavbu/ and register for your copy free of charge.

E.2 Getting Started with the AMI Utility

1. To obtain the AMI utility, fill out the registration form at
   http://www.ami.com/crownbayavbu/
   After submitting the form you will receive a confirmation e-mail and a link to a license agreement or a phone call to verify your information.
2. Using the link provided in the e-mail, download the utility.
3. Expand the utility with the password provided from the download page.
   You will have an `AVBU.exe` utility and an AVBU Users Guide from the expansion.
4. Use the AVBU utility to update or merge your EMGD VBIOS with your AMI system BIOS. Use the 'replace' (/r) command. This is an example replace command for Crown Bay:
   AVBU OABTN019.ROM /r vga.bin 8086 4108 /o updated.rom

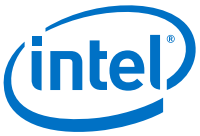
E.3 Supported Hardware Platforms

Currently the utility exclusively supports the Intel® Atom™ Processor E660 with Intel® Platform Controller Hub EG20T Development Kit.

E.4 Scope

The AVBU command line utility uses an internal table to control whether an option ROM can be added or removed from the image, depending on its Firmware ID, Firmware GUID, and the vendor and device IDs of the module.

This version of AVBU is designed to work specifically with the Intel® Atom™ Processor E660 with Intel® Platform Controller Hub EG20T Development Kit; it will not function on other platforms.



E.5 Features

- Filtered access: an internal table controls access to option ROM modules based on the identity of the BIOS and the module vendor and device identifiers. If there is no table match, the program issues the error message: “Access denied”
- Extract or copy a video option ROM
- Insert or Replace existing video option ROM
- Delete video option ROM

§ §

Appendix F Installing Video Camera Drivers for LAPIS[1] Semiconductor* ML7213 Board

Note: These steps should be done BEFORE installing Intel® EMGD or anything else.

1. These instructions assume that you have a fresh install of the MeeGo 1.2 IVI release set up on your system. If that is not the case, you can download the MeeGo 1.2 IVI installer from
   https://meego.com/downloads/releases/1.2/meego-v1.2-in-vehicle-infotainment-ivi
   Installation instructions are available from the site.
   If the GUI install mode does not work, try using text install mode instead.
2. If you are getting a blank screen upon boot, press **CTRL-ALT-F1** to access a terminal. You will need to modify your `xorg.conf` file to suit your display settings.
   If you are using a 13-inch LVDS panel, use the following workaround:
   Log in as root. Default password is **meego**.
   mv /etc/X11/xorg.conf /etc/X11/xorg.conf.bac
   mv /usr/share/doc/emgd-bin/emgd-cb.conf /etc/X11/xorg.conf
   reboot
3. Make sure your system time is set correctly in the BIOS.
4. Open a terminal (home menu, tools, others, terminal) and su to root. Default password is **meego**.
5. Download the kernel source file `kernel-adaptation-intel-automotive-2.6.37.6-10.5.src.rpm` from
   http://repo.meego.com/MeeGo/releases/1.2.0/repos/oss/source/
6. Install it with the command:
   rpm -Uvh kernel-adaptation-intel-automotive-2.6.37.6-10.5.src.rpm
7. Go to the folder `/root/rpmbuild/SPECS`.
8. Install the following tools:
   zypper in rpmdevtools make linux-firmware elfutils-libelf-devel binutils-devel newt-devel ncurses-devel libtool
9. Build the spec file:
   rpmbuild -bp kernel-adaptation-intel-automotive.spec
10. Go to the folder
    `/root/rpmbuild/BUILD/kernel-adaptation-intel-automotive-2.6.37.6/linux-2.6.37`
11. Install gcc:
    zypper in gcc

Note: Do not install gcc before step 7 (rpmbuild), or rpmbuild will fail with an error message regarding NETFILTER.

1. Formerly OKI Semiconductor.

12. Configure the kernel:
    make menuconfig
13. In the menu selection program, select the camera module you want to support:
    a. Go to the menu Device Drivers -> Multimedia Support -> Video Capture Adapters -> **Select IOH VIDEO IN Device**
    b. Select the correct camera device for your system. If you are unsure, select **OV7620**.
    c. Exit and save the configuration.
14. Build the kernel:
    make
    make modules
    make modules_install
    make install
15. Reboot
16. Install the EMGD drivers, libva, and GStreamer (see Section 7.3, "Installation" on page 134 and Section 7.4.10.4, "Installing MPlayer" on page 157)

Note: You may use the LAPIS camera drivers (`Lin_Camera_EMGD_1.10`, obtainable from your Intel representative, or the QUAD system) or use the generic `v4l2src` drivers. The following instructions presume use of the LAPIS proprietary drivers.

17. Uncompress the file:
    tar xvf Lin_Camera_EMGD_1.16
18. Change directories to `IEMGD_LIN_CPlugin\camera\eci`
19. Build and install the `eci` driver
    chmod +x autogen.sh
    ./autogen.sh
    ./configure --prefix=/usr CFLAGS=-O2 Note: O2 = letter "O"2, not zero
    make
    make install
20. Change directories to `IEMGD_LIN_CPlugin\camera\gst_egd_camera_src`.
21. Build and install the gstreamer egdcamsrc module.
    chmod +x autogen.sh
    ./autogen.sh
    ./configure --prefix=/usr CFLAGS=-O2 Note: O2 = letter "O"2, not zero
    make
    make install
22. Enable `Video In` in the BIOS:
    a. Reboot.
    b. Press F2 to enter the CMOS Setup page.
    c. Go to Advanced, IVI IOH Options and set Digital Video Input Configuration to **Enabled**.
23. Encode with the following command from terminal:
    gst-launch-0.10 egdcamsrc num-buffers=1800 ! video/x-raw-yuv,width=640,height=480,format=\(fourcc\)UYVY ! ffmpegcolorspace ! MixVideoEncoderH264 bit-rate=2000000 rate-control=CBR need-display=1 name=enc ! queue ! qtmux ! filesink location=~/test.H264 enc. ! vaimagesink sync=false

§ §