

Intelligent  
Systems

# Intel<sup>®</sup> Embedded Media and Graphics Driver v1.15 for Intel<sup>®</sup> Atom<sup>™</sup> Processor N2000 and D2000 Series

Preliminary User Guide

---

*August 2012*

---

**Notice:** This document contains preliminary information on new products in production. The specifications are subject to change without notice. Verify with your local Intel sales office that you have the latest datasheet before finalizing a design.

---

Document Number: 493848-004US



INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

A "Mission Critical Application" is any application in which failure of the Intel Product could result, directly or indirectly, in personal injury or death. SHOULD YOU PURCHASE OR USE INTEL'S PRODUCTS FOR ANY SUCH MISSION CRITICAL APPLICATION, YOU SHALL INDEMNIFY AND HOLD INTEL AND ITS SUBSIDIARIES, SUBCONTRACTORS AND AFFILIATES, AND THE DIRECTORS, OFFICERS, AND EMPLOYEES OF EACH, HARMLESS AGAINST ALL CLAIMS COSTS, DAMAGES, AND EXPENSES AND REASONABLE ATTORNEYS' FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PRODUCT LIABILITY, PERSONAL INJURY, OR DEATH ARISING IN ANY WAY OUT OF SUCH MISSION CRITICAL APPLICATION, WHETHER OR NOT INTEL OR ITS SUBCONTRACTOR WAS NEGLIGENT IN THE DESIGN, MANUFACTURE, OR WARNING OF THE INTEL PRODUCT OR ANY OF ITS PARTS.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or go to: <http://www.intel.com/design/literature.htm>

Any software source code reprinted in this document is furnished for informational purposes only and may only be used or copied and no license, express or implied, by estoppel or otherwise, to any of the reprinted source code is granted by this document.

[This document contains information on products in the design phase of development.](#)

Intel processor numbers are not a measure of performance. Processor numbers differentiate features within each processor family, not across different processor families. Go to: <http://www.intel.com/products/processor%5Fnumber/>

Intel, the Intel logo, and Intel Atom are trademarks of Intel Corporation in the U.S. and/or other countries.

\*Other names and brands may be claimed as the property of others.

Copyright © 2012, Intel Corporation. All rights reserved.



# Contents

---

<b>1.0</b>	<b>Introduction.....</b>	<b>8</b>
1.1	Purpose .....	8
1.2	Intended Audience .....	8
1.3	Related Documents .....	9
1.4	Conventions .....	9
1.5	Acronyms and Terminology.....	9
<b>2.0</b>	<b>Architectural Overview .....</b>	<b>13</b>
2.1	Introduction .....	13
2.1.1	Display Options.....	15
2.2	Features .....	16
2.2.1	Chipsets Supported .....	16
2.2.2	OS and API Support.....	16
2.2.3	DisplayID Support .....	16
2.2.4	EDID-Less Configuration .....	16
2.2.5	Rotation.....	17
<b>3.0</b>	<b>Platform Configuration Using CED.....</b>	<b>18</b>
3.1	Before You Begin.....	18
3.2	Creating a Configuration in CED – Summary Steps.....	19
3.3	Starting CED.....	19
3.4	Creating a New Customized DTD .....	20
3.4.1	DTD Example Specifications.....	22
3.5	Creating a New Configuration.....	23
3.5.1	Setting Color Correction .....	26
3.5.2	Changing Windows Embedded Compact 7 OS Options .....	28
3.5.3	Configuring Ports .....	29
3.5.4	Configuring EFI GOP .....	35
3.5.5	Configuring the Video BIOS and EFI .....	38
3.6	Creating a New Package.....	41
3.6.1	Entering Windows OS Options .....	42
3.6.2	Entering General Windows Embedded Compact 7 Options .....	44
3.6.3	Generating a VBIOS Package .....	44
3.6.4	Entering EFI Options.....	45
3.7	Generating an Installation .....	47
3.8	Configuring the System BIOS for Use with the Intel® EMGD.....	47
3.9	System BIOS Settings.....	47
3.9.1	GMCH PCIe Device Enabling.....	47
3.9.2	Graphics Mode Select (GMS).....	48
3.9.3	AGP (Accelerated Graphics Port) Aperture Size .....	48
3.10	VBIOS and Driver Configuration .....	48
3.11	Configuration Options .....	51
3.12	Display Detection and Initialization .....	58
3.12.1	Display Detect Operation.....	58
3.12.2	Detectable Displays .....	59
3.13	Advanced EDID Configuration .....	60
3.13.1	Sample Advanced EDID Configurations.....	61
3.13.2	User-Specified DTDs .....	61
3.14	Enhanced Clone Mode Support .....	61
3.14.1	Clone Mode CED Configuration .....	62



- 3.15 Scaling and Centering Configurations ..... 63
  - 3.15.1 Internal LVDS Scaling with EDID Panels ..... 63
  - 3.15.2 Centering Primary Display ..... 63
  - 3.15.3 Alignment in Clone Mode ..... 64
- 4.0 Video BIOS Firmware ..... 65**
  - 4.1 Overview ..... 65
  - 4.2 System Requirements ..... 65
  - 4.3 Configuring and Building the VBIOS with CED ..... 65
    - 4.3.1 Selecting the Build Folder ..... 66
    - 4.3.2 Configuring the Video BIOS ..... 67
    - 4.3.3 Building the VBIOS ..... 69
  - 4.4 VBIOS, Driver Compatibility, and Data Dependencies ..... 72
    - 4.4.1 VESA and VGA Video Modes ..... 72
- 5.0 Configuring and Installing Microsoft Windows Drivers ..... 76**
  - 5.1 Editing the Microsoft Windows INF File ..... 76
  - 5.2 Configuration Information ..... 76
    - 5.2.1 Universal INF Configuration ..... 76
    - 5.2.2 Dual Panel Configuration ..... 77
    - 5.2.3 Chipset Dual Display Example ..... 77
    - 5.2.4 Creating Registry Settings for Graphics Driver INF File ..... 77
    - 5.2.5 Dynamic Port Driver Configuration ..... 80
    - 5.2.6 Changing Default Display Mode ..... 81
    - 5.2.7 Creating an .sld file for Microsoft Windows XP Embedded Systems ..... 82
  - 5.3 Installing Intel® Embedded Media and Graphics Driver on Microsoft Windows ..... 82
    - 5.3.1 Silent Installation ..... 83
  - 5.4 Uninstalling the Current Version of the Driver ..... 83
  - 5.5 Runtime Operation ..... 84
  - 5.6 Viewing and Changing the Driver Configuration from Microsoft Windows ..... 84
- 6.0 Configuring and Building Intel® EMGD for Microsoft Windows\* Embedded Compact 7 ..... 89**
  - 6.1 Microsoft Windows\* Embedded Compact 7 Installation ..... 89
    - 6.1.1 Prerequisites ..... 89
    - 6.1.2 Platform Builder Requirements ..... 89
    - 6.1.3 Integrating Intel® EMGD DirectX DirectShow Codecs ..... 93
  - 6.2 Microsoft Windows\* Embedded Compact 7 Configuration ..... 94
    - 6.2.1 Basic Driver Configuration ..... 94
    - 6.2.2 Configuration Sets ..... 99
    - 6.2.3 General Configuration ..... 99
    - 6.2.4 Per Port Platform Customization ..... 103
    - 6.2.5 Miscellaneous Configuration Options ..... 106
    - 6.2.6 Sample emgd.reg File ..... 106
- A Example INF File ..... 109**
- B Port Driver Attributes ..... 114**
  - B.1 Standard Port Driver Attributes ..... 114
  - B.2 Port Driver Attributes ..... 116
    - B.2.1 Internal LVDS Port Driver Attributes (Mobile chipsets only) ..... 116
    - B.2.2 HDMI Port Driver Attributes ..... 117
  - B.3 Chipset and Port Driver-specific Installation Information ..... 118
    - B.3.1 Default Search Order ..... 118
    - B.3.2 Default GPIO Pin Pair Assignments ..... 118



<b>C</b>	<b>Intel® 5F Extended Interface Functions</b> .....	<b>119</b>
C.1	BIOS Extended Interface Functions .....	120
C.1.1	5F01h – Get Video BIOS Information .....	120
C.1.2	5F05h – Refresh Rate .....	120
C.1.3	5F10h – Get Display Memory Information .....	122
C.1.4	5F1Ch – BIOS Pipe Access .....	122
C.1.5	5F29h – Get Mode Information .....	123
C.1.6	5F61h – Local Flat Panel Support Function .....	123
C.1.7	5F68h – System BIOS Callback .....	124
C.2	Hooks for the System BIOS .....	124
C.2.1	5F31h – POST Completion Notification Hook .....	124
C.2.2	5F33h – Hook After Mode Set .....	124
C.2.3	5F35h – Boot Display Device Hook .....	125
C.2.4	5F36h – Boot TV Format Hook .....	126
C.2.5	5F38h – Hook Before Set Mode .....	126
C.2.6	5F40h – Config ID Hook .....	127
<b>D</b>	<b>2D/3D API Support</b> .....	<b>128</b>
D.1	2D Support .....	128
D.2	3D Support .....	128

## Figures

1	Intel® Embedded Media and Graphics Driver .....	13
2	Graphics Driver Architecture .....	14
3	Firmware Architecture .....	14
4	Intel® EMGD Configuration Editor Main Window .....	19
5	EMGD DTD Page .....	20
6	Chipset Configuration Page .....	24
7	Overlay Color Correction Page .....	26
8	Framebuffer Color Correction Page .....	27
9	Windows Embedded Compact 7 Configuration Page .....	28
10	Port Configuration Page .....	30
11	Attribute Settings Page for Internal LVDS .....	32
12	DDC Settings Page .....	33
13	Panel Settings Page .....	34
14	EFI GOP Configuration Page .....	35
15	Splash Video with 8 MB of Stolen Memory Example .....	37
16	Video BIOS Configuration Page .....	39
17	Intel® EMGD Package Editor Page .....	41
18	Windows Options Page .....	43
19	General Driver Windows Embedded Compact 7 Settings Page .....	44
20	VBIOS Generation Page .....	45
21	EFI Generation Page .....	45
22	LVDS Configuration Page .....	49
23	Intel® EMGD Configuration Editor Page .....	50
24	Video BIOS Directory Structure .....	67
25	Example Runtime Configuration GUI — Driver Info Tab .....	85
26	Example Runtime Configuration GUI — Display Config Tab .....	85
27	Example Runtime Configuration GUI — Display Attributes Tab .....	86
28	Example Runtime Configuration GUI — Color Correction Tab .....	87
29	Typical Memory Map Using Static Memory Model .....	96



## Tables

1	Acronyms and Terminology	9
2	Types of Displays Supported	15
3	Display Configuration Definitions	15
4	Supported Display Configurations	15
5	Chipsets Supported by Intel® EMGD v1.15	16
6	Intel® EMGD DTD Setting Options	21
7	Timing Specification Example Values	23
8	Chipset Configuration Page Settings	25
9	Overlay Color Correction Values (applies to ALL color)	26
10	Framebuffer Color Correction Values (applies to R, G, B color)	27
11	Windows Embedded Compact 7 Settings	28
12	Port Configuration Settings	31
13	DDC Settings	33
14	Panel Settings Options	34
15	EFI GOP Options	36
16	Video BIOS Settings Options	40
17	Intel® EMGD Package Editor Setting Options	42
18	Windows OS Setting Options	43
19	Driver Windows Embedded Compact 7 Settings Options	44
20	GMCH Device 2, Function 1 BIOS Setting	47
21	GMS Settings	48
22	Parameter Configuration Format	51
23	Detectable Displays	59
24	Sample Advanced EDID Configurations	61
25	Supported VGA Video Display Modes	72
26	VESA Modes Supported by Video BIOS	74
27	Dual Display Parameter Setting	77
28	Framebuffer Color Correction Values (applies to R, G, B color)	87
29	Overlay Color Correction Values (applies to ALL color)	87
30	[HKLM\DRIVERS\Display\Intel] Registry Keys	95
31	[HKLM\Drivers\Display\Intel\<platform>\<config id>\]Registry Keys	100
32	PortOrder Information	102
33	Standard Port Driver Attributes	114
34	Internal LVDS Port Driver Attributes	116
35	Default Search Order	118
36	Default GPIO Pin Pair Assignments	118
37	Summary of Intel 5F Extended Interface Functions	119



## Revision History

---

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
August 2012	004	Intel® Embedded Media and Graphics Driver v1.15 for Intel® Atom™ Processor N2000 and D2000 Series Preliminary User Guide.
July 2012	003	Intel® Embedded Media and Graphics Driver v1.12 for Intel® Atom™ Processor N2000 and D2000 Series User Guide.
June 2012	002	Intel® Embedded Media and Graphics Driver v1.15 for Intel® Atom™ Processor N2000 and D2000 Series Preliminary User Guide.
April 2012	001	Intel® Embedded Media and Graphics Driver v1.12 for Intel® Atom™ Processor N2000 and D2000 Series Preliminary User Guide.

§ §



## 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 N2000 and D2000 Series (Windows XP/Windows Embedded Compact 7)

*Note:* If you need support for a chipset that is not listed above but is in the same family as those listed, please first check for an appropriate version for download on the Intel® Embedded Design Center ([www.intel.com/go/EMGD](http://www.intel.com/go/EMGD) or [www.intel.com/go/IEGD](http://www.intel.com/go/IEGD)) or contact your Intel representative.

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

- Analog CRT
- LVDS flat panels
- HDMI
- DisplayPort
- Embedded DisplayPort (eDP)

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 Microsoft Windows\* operating systems, including embedded versions of these operating systems.

Intel® Embedded Media and Graphics Driver provides the following features:

- Support of a Universal INF file.
- Display discovery and initialization.

*Note:* **Certain features are not applicable in this release. Please refer to product release notes for more detail.**

### 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://www.intel.com/go/EMGD> (<http://edc.intel.com/Software/Downloads/EMGD/>).

- *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.

<b>Boldface</b>	Represents text that you type and text that appears on a screen.
<i>Italics</i>	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* operating systems, and VBIOS across numerous platforms and display combinations.



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

Term	Description
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 DirectX*. 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.
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.
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.
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).



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

Term	Description
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.
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.
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	The portion of the VBIOS or graphics driver that handles the display interface hardware (port).
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 Radiorecepteurs 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.
Single Display Configuration	A type of display configuration that supports one and only one display device.
SSC	Spread Spectrum Clock.



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

Term	Description
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.
VESA	Video Electronics Standards Organization.
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.





## 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

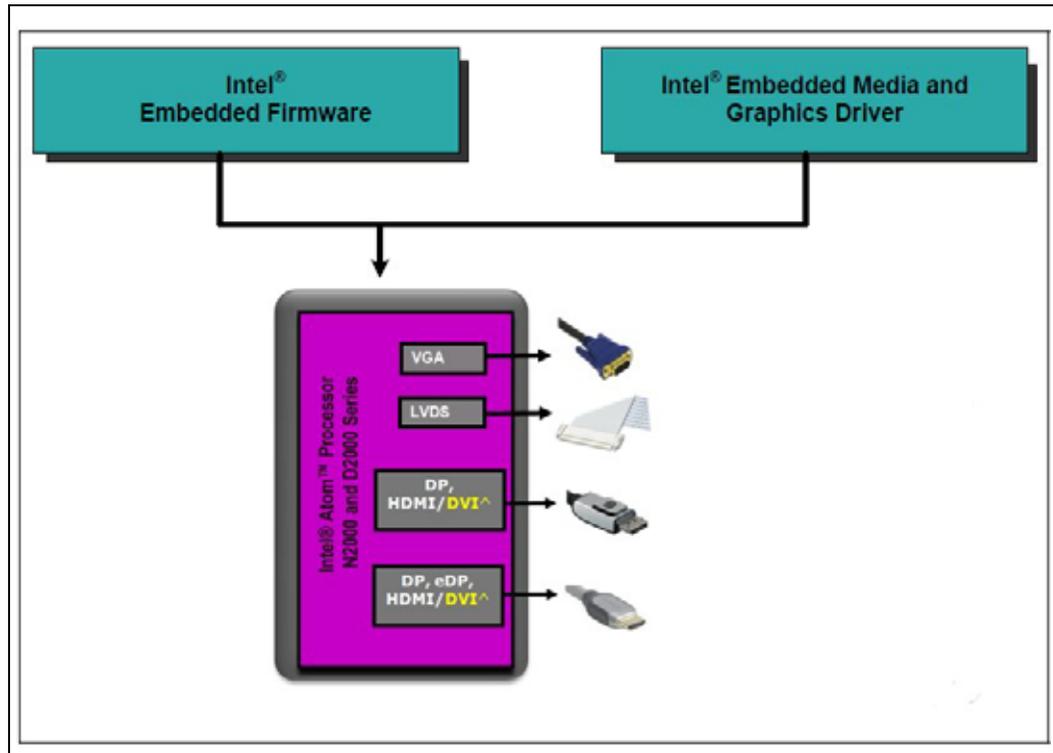


Figure 2. Graphics Driver Architecture

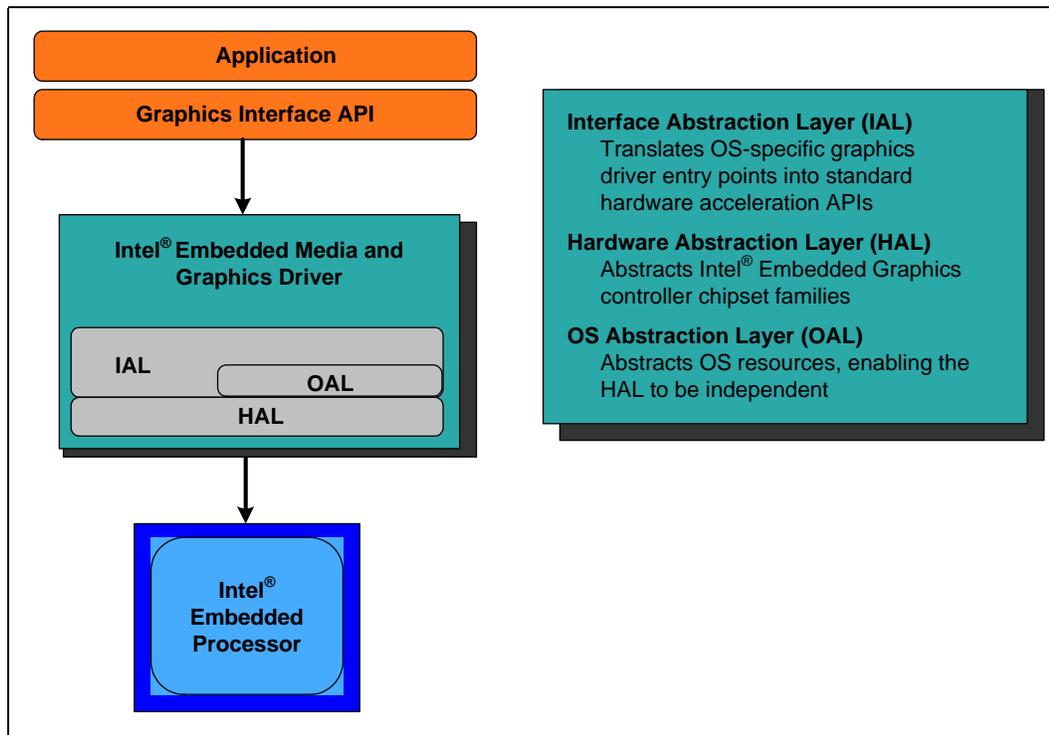
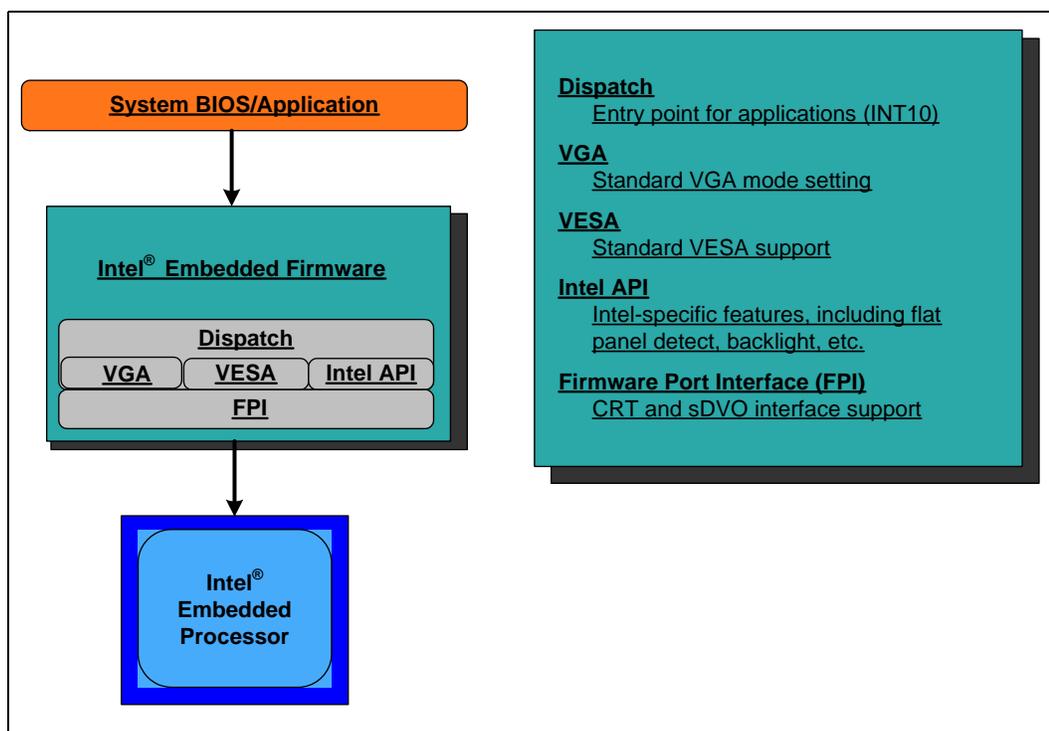


Figure 3. Firmware Architecture





## 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. Also known as “VGA” typically using a 15 pin D-Sub connector.
Flat Panel	TMDS and LVDS compliant flat panels are supported.
HDMI	High-Definition Multimedia Interface (video only, no audio)
DP	DisplayPort
eDP	embedded DisplayPort

### 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, different content, independent resolutions

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
Intel® Atom™ Processor N2000 and D2000 Series	Single, Clone, Extended	Single, Clone, Extended

Intel® EMGD supports Clone mode through custom APIs. Microsoft Windows operating systems natively support Extended mode.



## 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 Intel® EMGD v1.15

Chipset	Intel® EMGD VBIOS Support	Intel® EMGD Support
Intel® Atom™ Processor N2000 and D2000 Series	No	Yes

All supported chipsets provide for SINGLE LVDS output. In addition, digital monitors, CRTs and TVs are supported through the integrated display ports such as Display Port (DP), embedded Display Port (eDP), HDMI, etc. 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.

- Microsoft Windows\* XP with Service Pack 3, Windows\* XP Embedded with Embedded Standard 2009
  - DirectX\* 9.0C (DirectDraw\* and Direct3D\*)
- Microsoft Windows\* Embedded Compact 7

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

- 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](http://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](http://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 20](#).

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 operating systems:

- Intel® Atom™ Processor N2000 and D2000 Series

*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 N2000 and D2000 Series

Refer to [Section 2.2.2, “OS and API Support” on page 16](#) 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 20](#)
- [“Creating a New Configuration” on page 23](#)
- [“Creating a New Package” on page 41](#)
- [“Generating an Installation” on page 47](#)

*Note:* There are two versions of CED, one for Windows XP 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 23](#) 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.

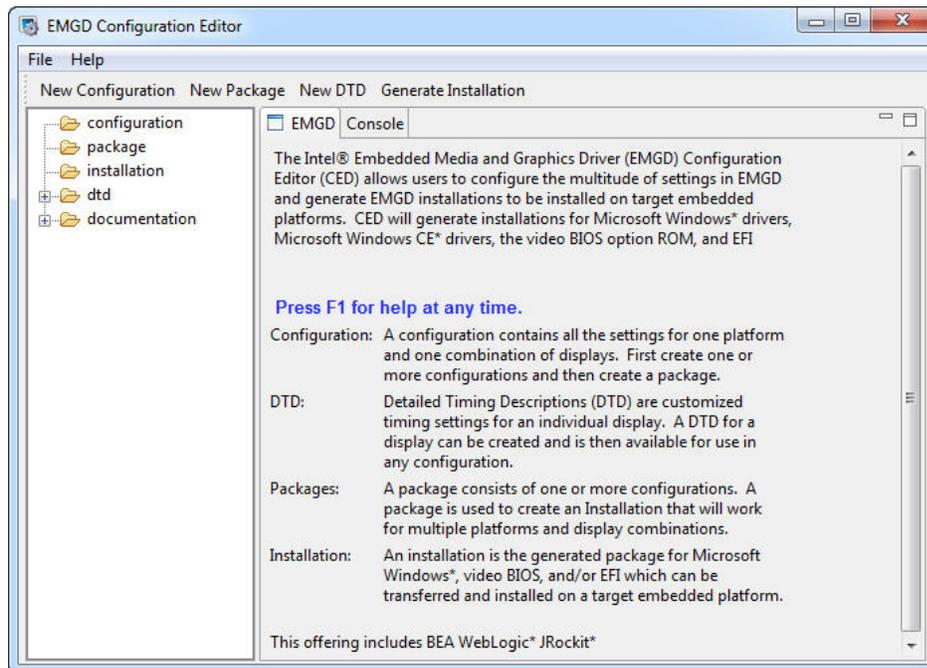
1. (Optional) If you have custom panels and timings you may want to create your own DTD; otherwise you can use the standard DTDs provided by CED. If needed, select **New DTD**.
  - Choose the DTD Type that most closely aligns with your display parameters, enter parameters, and then click **Finish**. Or, to create a DTD, see [“Creating a New Customized DTD” on page 20](#).
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 23](#).
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 41](#).
4. Select the created package and then select **Generate Installation**.
 

The generated files are placed in the installation folder. The zip files for Windows, and Windows Embedded Compact 7 operating systems contain the generated configuration files. For details, see [“Generating an Installation” on page 47](#).

## 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 4. 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.

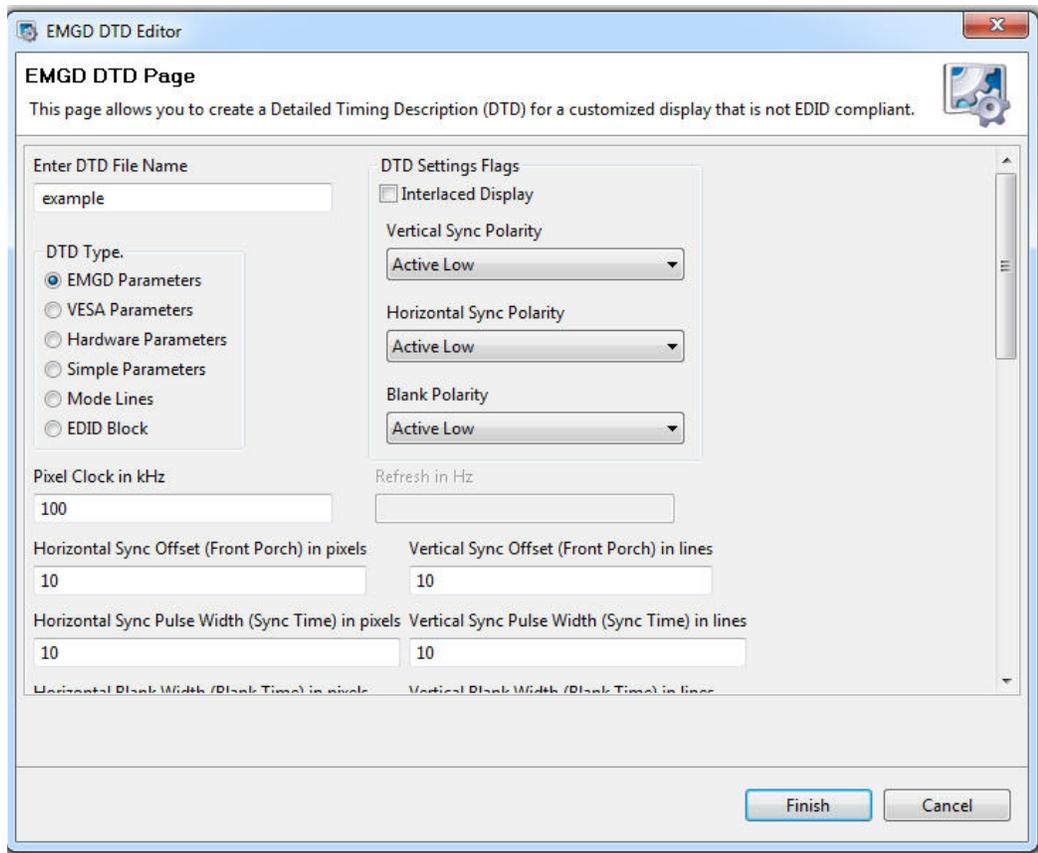
### 3.4 Creating a New Customized DTD

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 5. 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 above.
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	<p>Select the DTD Type that most closely aligns with your display parameters. Options are:</p> <ul style="list-style-type: none"> <li>• <b>Intel® EMGD Parameters:</b> The Intel® EMGD Parameters are the same as the current PCF/CED DTD parameters.</li> <li>• <b>VESA Parameters:</b> The VESA Parameters allow the user to create a DTD from a VESA monitor timing standard.</li> <li>• <b>Hardware Parameters:</b> The Hardware Parameters are the parameters that are used by Intel® EMGD.</li> <li>• <b>Simple Parameters:</b> The Simple Parameters (CVT Standard) is a process for computing standard timing specifications. The method for developing Reduced Blanking timings is not included.</li> <li>• <b>Mode Lines:</b> 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.</li> <li>• <b>EDID Block:</b> 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</li> </ul>
Pixel Clock	Pixel clock value in KHz. Range 0-0x7ffffff.
DTD Settings Flags	<p>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.</p> <p><b>Interlaced Display:</b></p> <ul style="list-style-type: none"> <li>• Check for Interlaced</li> <li>• Cleared for Non-interlaced</li> </ul> <p><b>Vertical Sync Polarity:</b></p> <ul style="list-style-type: none"> <li>• Active Low (Default)</li> <li>• Active High</li> </ul> <p><b>Horizontal Sync Polarity:</b></p> <ul style="list-style-type: none"> <li>• Active Low (Default)</li> <li>• Active High</li> </ul> <p><b>Blank Polarity:</b></p> <ul style="list-style-type: none"> <li>• Active Low (Default)</li> <li>• Active High</li> </ul> <p><b>Note:</b> These flags are Intel® EMGD-specific and do not correspond to VESA 3.0 flags.</p>



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

DTD Parameter	Description
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].
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

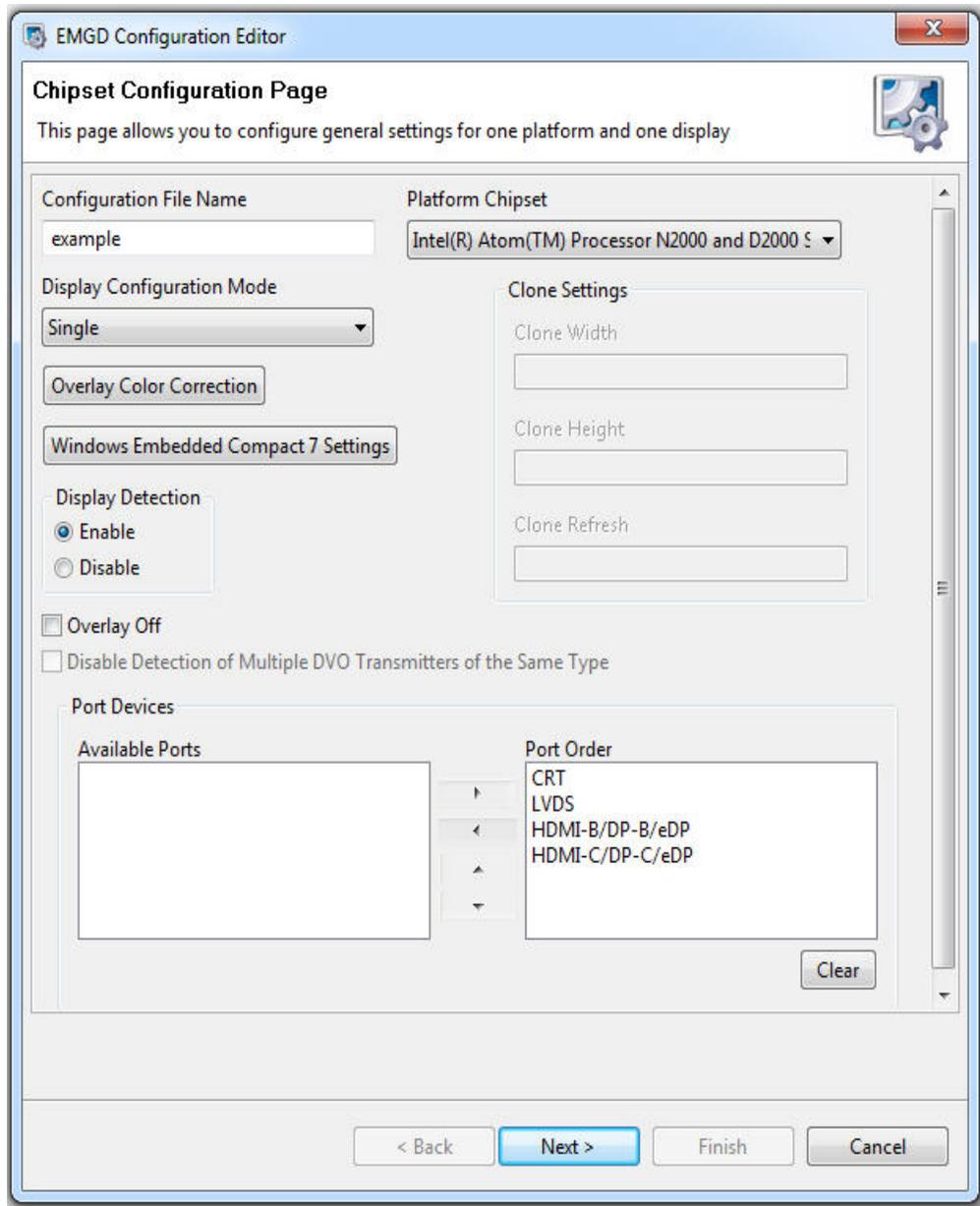
Item		Symbol	Standard value			Unit
			Min.	Typ.	Max.	
Clock	Frequency	1/ts	29.91	33.231	36.55	MHz
	Period	ts	27.36	30.06	33.43	ns
	Hi-time	tsh	7	–	–	ns
	Low-time	tsl	7	–	–	ns
	DUTY ratio	th/tl	35	50	65	ns
Data	Setup time	tds	7	–	–	ns
	Hold time	tdh	4	–	–	ns
H sync.	Period	t <sub>lpl</sub> , t <sub>lpd</sub>	24.51	31.75	32.05	us
			880	1056	1088	clk
	Pulse width	t <sub>lw</sub>	3	128	200	clk
H display	Term	t <sub>hd</sub>	800	800	800	clk
Enable	Setup time	t <sub>drs</sub>	7	–	–	ns
	Hold time	t <sub>drh</sub>	4	–	–	ns
V sync.	Period	t <sub>fpf</sub> , t <sub>fpd</sub>	520	525	680	Line
	Pulse width	t <sub>fw</sub>	1	2	3	Line
V display	Term	t <sub>vd</sub>	480	480	480	Line
	Start	t <sub>fd</sub>	10	33	40	Line
Phase difference	H sync. ~ enable	t <sub>drds</sub>	50	216	260	clk
	H sync. ~ clock	t <sub>ls</sub>	7	–	–	ns
	H sync. ~V sync.	t <sub>n</sub>	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 6. 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

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: <ul style="list-style-type: none"> <li>• Single — Single display configuration.</li> <li>• Clone — Two displays where both displays have the same content but can have different resolutions and timings.</li> <li>• DIH — Dual Independent Head. This is a configuration where both displays can have different resolutions, different refresh rates, and different content.</li> </ul> <p><b>Note:</b> 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.</p>
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 <a href="#">“Overlay Color Correction” on page 26</a> .
Windows Embedded Compact 7 Settings	If you are creating a package for a Microsoft Windows* CE platform, click the <b>Windows Embedded Compact 7 Settings</b> button for additional settings that may be required for your configuration. Please see <a href="#">“Changing Windows Embedded Compact 7 OS Options” on page 28</a> for descriptions of these settings.
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 <a href="#">“Display Detection and Initialization” on page 58</a> .
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 <a href="#">Table 35, “Default Search Order” on page 118</a> 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 <b>Next</b> . For each port listed in the Port Order box, you can click <b>Next</b> to configure each port. See <a href="#">“Configuring Ports” on page 29</a> for information on configuring ports.
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 <a href="#">“Enhanced Clone Mode Support” on page 61</a> .
Overlay Off	This field allows you disable Overlay support, which is enabled by default.

### 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

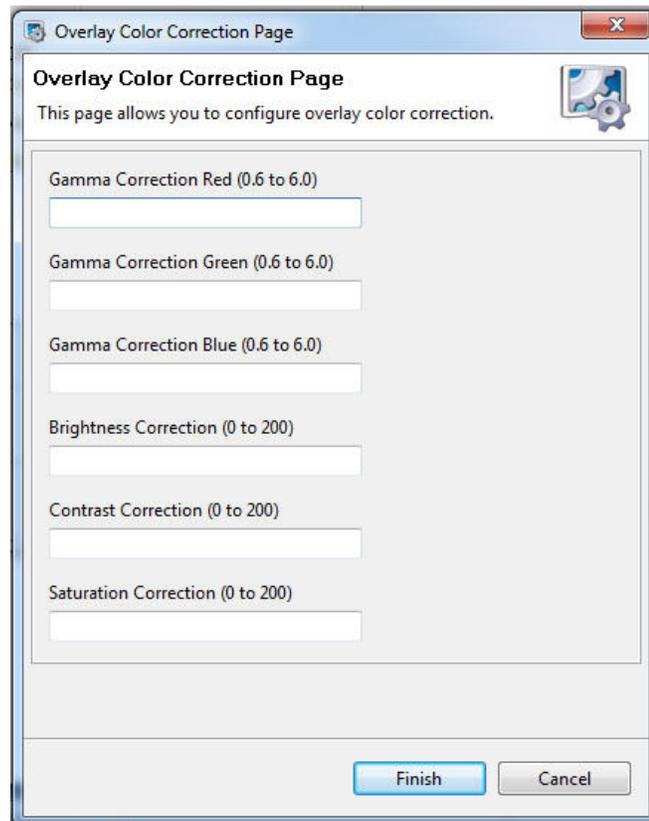
Overlay Color Correction allows the Overlay plane to have color-correction settings that are different from the main framebuffer color-correction settings. This feature lets you color-correct for red, green, and blue, plus it enables you to adjust brightness, contrast, and saturation.

**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 7. 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). The Framebuffer Color Correction Page appears, as shown in [Figure 8](#).

**Figure 8. Framebuffer Color Correction Page**

**Color Correction Attributes Page**

This page allows you to configure color correction attributes.

Gamma Correction Red (0.6 to 6.0)

Gamma Correction Green (0.6 to 6.0)

Gamma Correction Blue (0.6 to 6.0)

Brightness Correction Red (-127 to 127)

Brightness Correction Green (-127 to 127)

Brightness Correction Blue (-127 to 127)

Contrast Correction Red (-127 to 127)

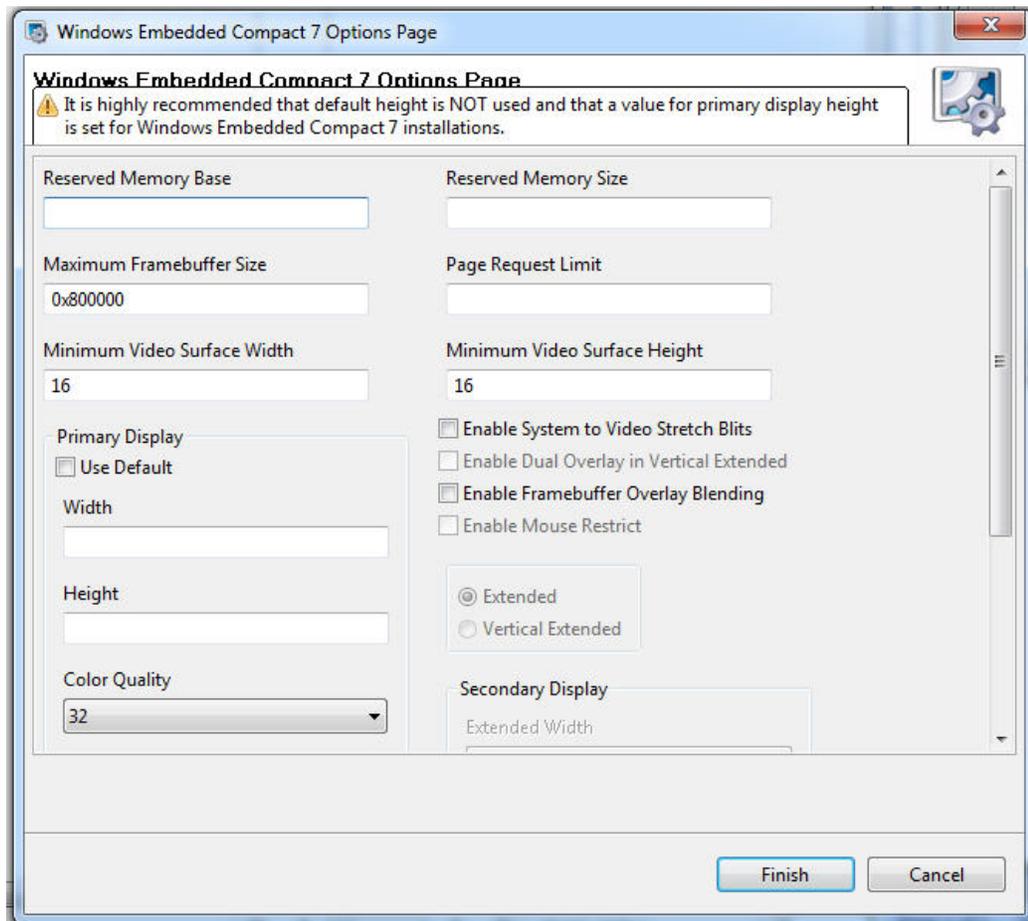
Finish Cancel

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

### 3.5.2 Changing Windows Embedded Compact 7 OS Options

The Windows Embedded Compact 7 Options Page allows you to enter Windows CE OS-specific options into the configuration. When you click the **Windows Embedded Compact 7 Settings** button from the EMGD Package Page (see [Figure 6, “Chipset Configuration Page”](#) on page 24), the following page appears.

**Figure 9. Windows Embedded Compact 7 Configuration Page**



The table below describes each field on this page.

**Table 11. Windows Embedded Compact 7 Settings**

Windows Embedded Compact 7 Option	Description
Reserved Memory Base Reserved Memory Size	<p>These two fields let you specify the amount and the starting point of statically reserved video memory. Video memory can be statically reserved or dynamically allocated on demand. If both Reserved Memory Base and Reserved Memory Size are non-zero, video memory allocation uses the static model. Base plus Size must extend to TOM (Top Of Memory) and not conflict with other reserved memory arenas in the config.bib file.</p> <p>The default for both Reserved Memory Base and Reserved Memory Size is zero, indicating a dynamic allocation model.</p> <p>Default behavior disables static memory model.</p>



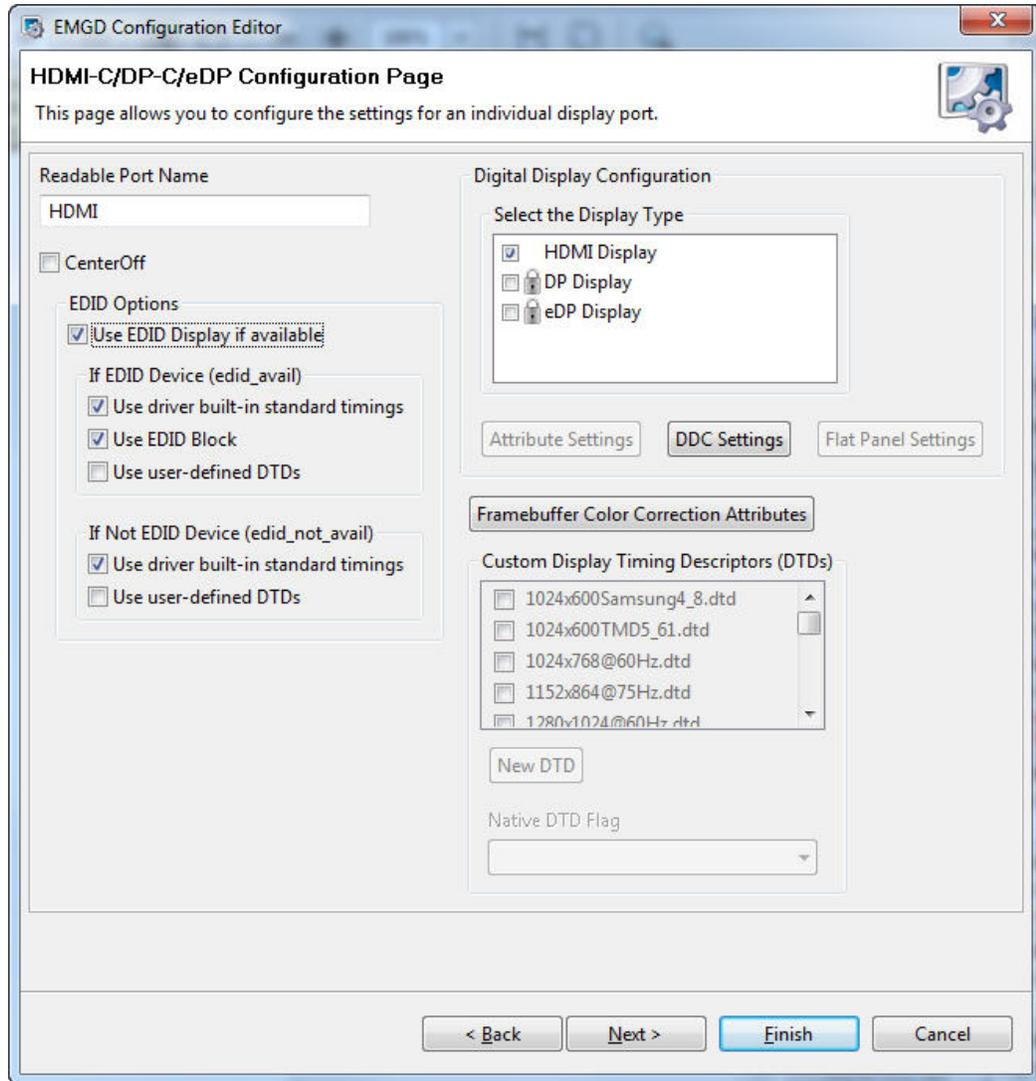
Table 11. Windows Embedded Compact 7 Settings

Windows Embedded Compact 7 Option	Description
Maximum Frame Buffer Size	The maximum size of the expected frame buffer. By providing this hint, the display driver can more efficiently organize GART memory, leading to a smaller video memory consumption. This value must be greater than or equal to the expected size of the frame buffer. Units represent the number of bytes and are specified in hexadecimal. Specifying zero causes the default frame buffer reservation sizing. The default is 0x300000
Page Request Limit	The Page Request Limit controls the maximum allocations of offscreen video surfaces, buffers, etc. This value represents the number of pages (4K) allocated and is independent of dynamic or static memory configuration. The maximum is 128MB (0x8000)
Minimum Video Surface Width Minimum Video Surface Height	In pixels, the minimum width and height of surfaces 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. Default value for both width and height is 16.
Enable System to Video Stretch Blits	When checked, this 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. The default is disabled.
Disable D3D Support	Specify whether to disable or enable D3D graphics.
Enable Dual Overlay in Vertical Extended	This option is available only if DIH (vertical extended) mode has been selected as the display configuration on the Chipset Configuration page. See <a href="#">Table 8, "Chipset Configuration Page Settings" on page 25</a> for details.
Enable Frame Buffer Overlay Blending	When checked, this option enables overlay blending with the framebuffer on both display outputs (if in VEXT mode) and when display mode resolution is 32-bit XRGB.
Enable Mouse Restrict	This is enabled only if the Dual Independent Head display config mode is selected. This control is disabled if Single or Clone display configuration mode is selected. When checked, it is displayed in the emgd.reg file as  "MouseRestrict"=dword: 1  The purpose of this option is to prevent the mouse from entering areas of the display buffer that are not visible on screen.
Display Use Default Width Height Color Quality Refresh	The Display 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 configuration uses the default display mode for the operating system it is installed on.  Intel recommends that you set the values here instead of leaving the resolution to be auto-detected.

### 3.5.3 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 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, DDC Settings, and Flat Panel Settings and create a new DTD that can be used with any configuration.

Table 12 describes each field on this page.



Table 12. 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	<p>This section allows you to set EDID options for the display. The Intel® EMGD supports three different types of EDID display modes:</p> <ul style="list-style-type: none"> <li>Built-in display modes: These modes are hard-coded in the Intel® EMGD. These modes can be filtered based on the EDID block.</li> <li>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.</li> <li>User-specified DTDs.</li> </ul> <p>If you want to use the display's EDID information if it is available, click the <b>Use EDID Display if Available</b> check box.</p> <p>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:</p> <ul style="list-style-type: none"> <li>Use driver built in standard timings — If this box is checked, the standard timings built into the Intel® EMGD are used.</li> <li>Use EDID block — If this box is checked, the EDID block is used.</li> <li>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 <b>New DTD</b> and create a custom DTD. For information on creating custom DTD, refer to Table 18, "Windows OS Setting Options" on page 43.</li> </ul> <p>If you select both <b>Use driver built-in standard timings</b> and <b>Use EDID block</b>, the Intel® EMGD uses its built-in display timings and the timings provided by the display.</p> <p>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:</p> <ul style="list-style-type: none"> <li>Use driver built-in standard timings — If this box is checked, the standard timings are used.</li> <li>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 <b>New DTD</b> and create a custom DTD. For information on creating custom DTD, refer to Table 18, "Windows OS Setting Options" on page 43.</li> </ul> <p>See "Sample Advanced EDID Configurations" on page 61 for example configurations.</p>
Digital Display Configuration	<p>This section lets you to specify the type of digital display connected to a port and encoder Attributes, I2C settings, and Flat Panel settings for the port.</p> <p>The <b>Select the Display Type</b> list contains the list of all supported digital devices. Select the device that will be connected to this port.</p> <p>To change the device's attributes, click the <b>Attribute Settings</b> button. Refer to "Changing Port Attribute Settings" for information on device attributes.</p> <p>To change the device's I2C settings, click the <b>I2C Settings</b> button. See "Changing DDC Settings" on page 33 for information on I2C settings.</p> <p>To change the device's flat panel settings, click the <b>Flat Panel Settings</b> button. See "Changing Flat Panel Settings" on page 33 for information for changing flat panel settings.</p>

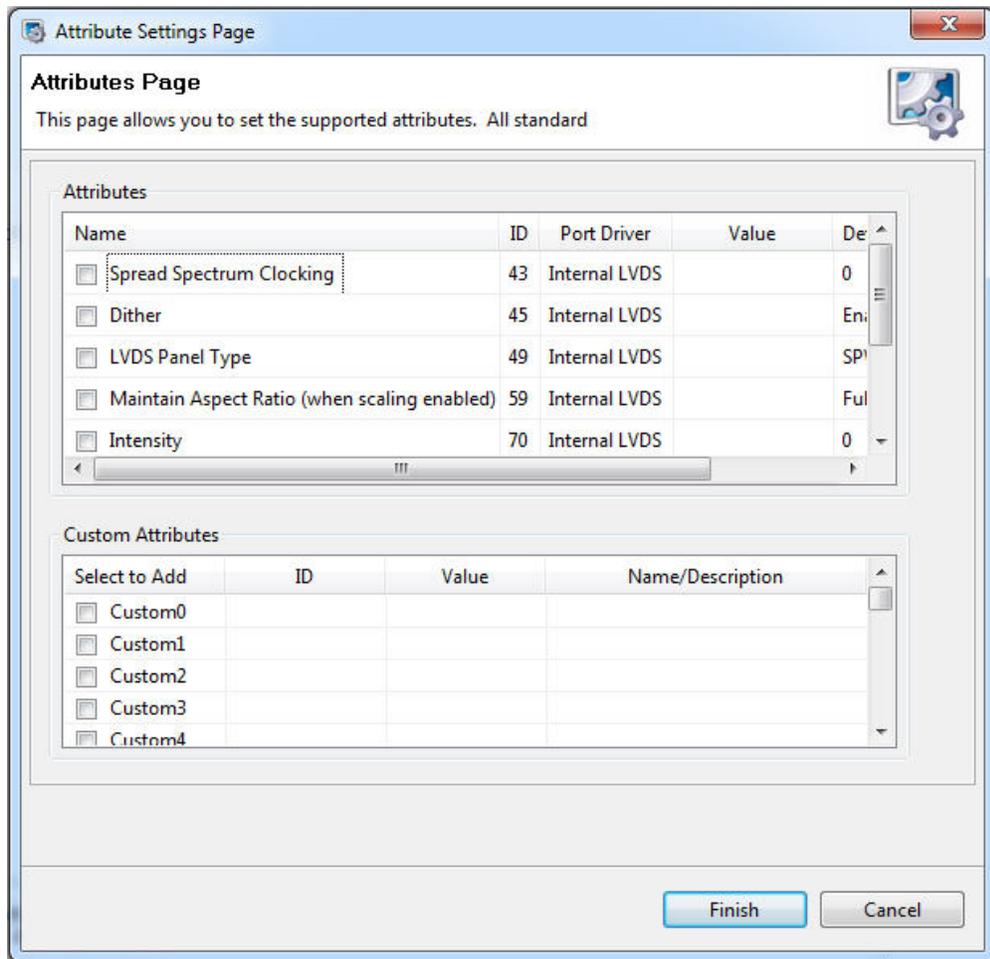
Table 12. Port Configuration Settings (Sheet 2 of 2)

Port Configuration Field	Description
Framebuffer Color Correction Attributes	Framebuffer Color Correction Attributes allow you to adjust the main Frame Buffer color attributes. See <a href="#">“Framebuffer Color Correction Attributes”</a> on page 27.
New DTD	To define a new Dynamic Timings Definition, click this option. See <a href="#">Section 3.4, “Creating a New Customized DTD”</a> on page 20.
Native DTD Flag	The Native DTD list lets you choose whether to use a display’s built-in timings.

### 3.5.3.1 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. Attribute Settings Page for Internal LVDS





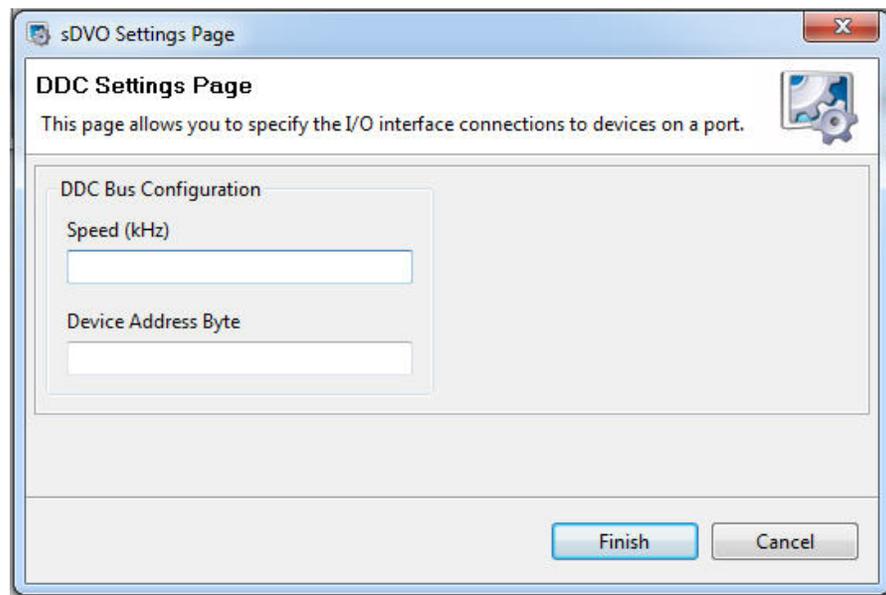
When the Attributes Settings Page first appears, it shows the **Use Default** box checked for all attributes.

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”](#).

### 3.5.3.2 Changing DDC Settings

The DDC Settings Page allows you to specify the I/O interface connections to devices on the HDMI, DisplayPort, or embedded DisplayPort ports. When you click **DDC Settings** from the Port Configuration Page, the following screen appears.

**Figure 12. DDC Settings Page**



The following table describes each field on this page.

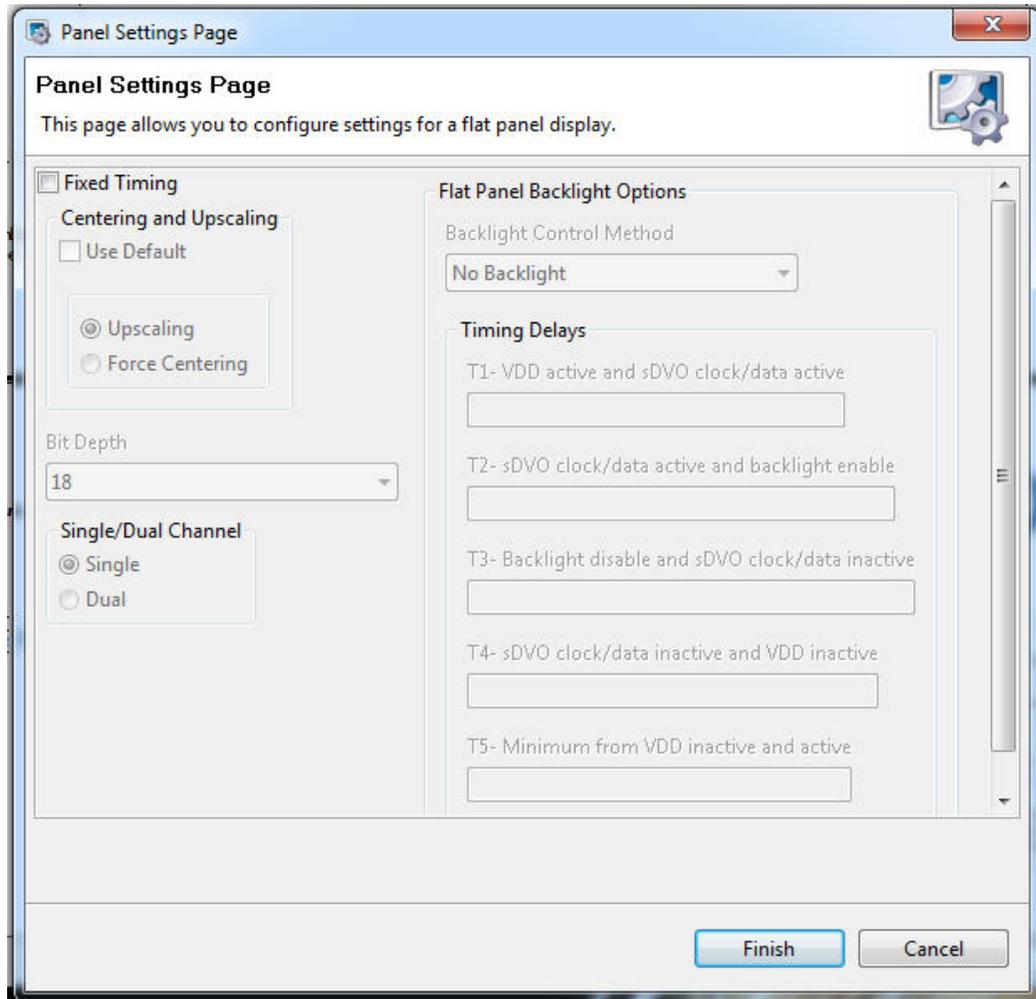
**Table 13. DDC Settings**

DDC Bus Configuration	Description
Speed (KHz)	Speed of DDC bus for the device. The range for this field is 10-400 KHz.
Device Address Byte	Enter a device address byte for the device that this port is connected to. The DDC Device Address Byte is the device address for reading EDID data from the display through the DDC bus.

### 3.5.3.3 Changing Flat Panel Settings

The Panel Settings Page allows you to specify settings for a flat panel display connected to the LVDS 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 14. 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 <b>Use Default</b> 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	<p>This section provides options for controlling the backlight of the flat panel display and specifying timing delays.</p> <ul style="list-style-type: none"> <li>The Backlight Control Methods list lets you choose either <b>No Backlight</b> or <b>Port Driver</b> 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.</li> </ul>

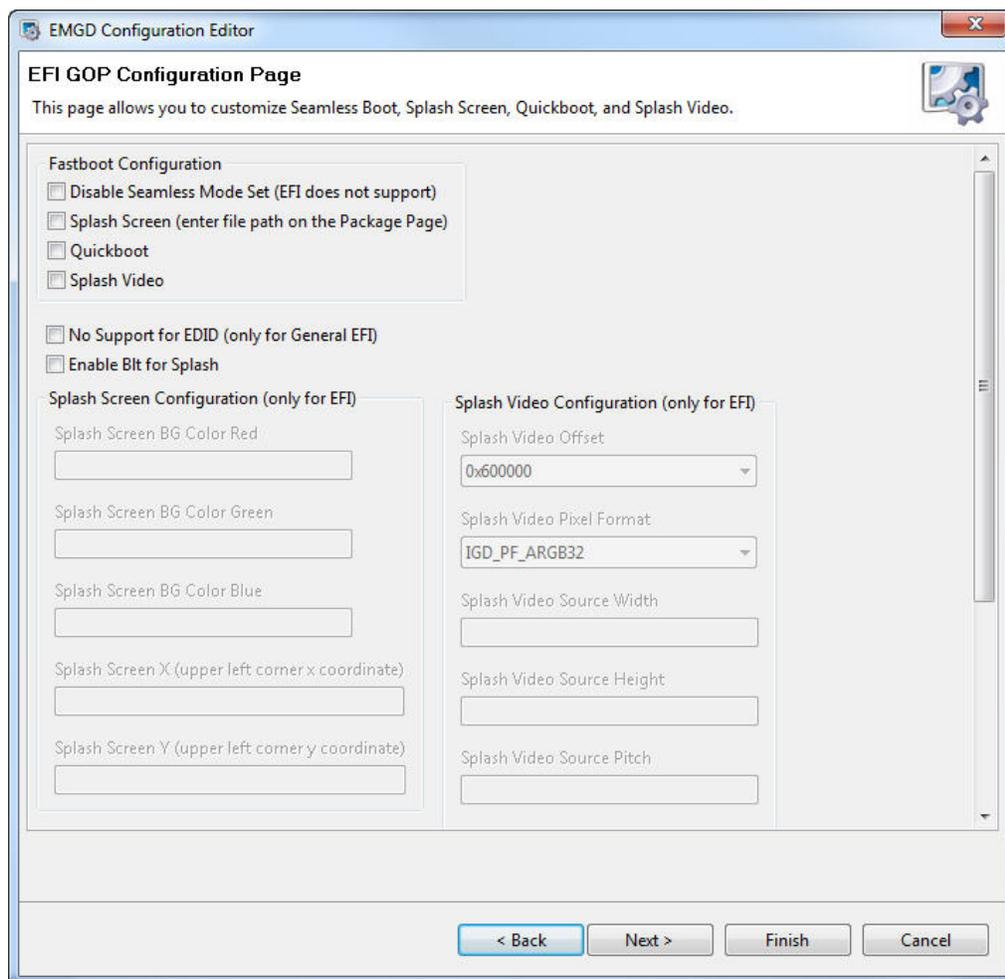


Table 14. Panel Settings Options (Sheet 2 of 2)

Flat Panel Settings	Description
Timing Delays	<p>This section lets you specify timing delays for the backlight signals as follows:</p> <ul style="list-style-type: none"> <li>• T1-VDD active: 1-512, increment by 1.</li> <li>• T2-DVO active and backlight enable: 2-256, increment by 2.</li> <li>• T3-Backlight disable and DVO clock/data inactive: 2-256, increment by 2.</li> <li>• T4-DVO clock/data active and inactive: 1-512, increment by 1.</li> <li>• T5-Minimum from VDD inactive and active: 1-1600, increment by 50.</li> </ul> <p><b>Note:</b> 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.</p>

### 3.5.4 Configuring EFI GOP

Figure 14. EFI GOP Configuration Page





**Note:** Enter the file path for the splash video on the Package Page. See Figure 17, “Intel® EMGD Package Editor Page” on page 41.

**Table 15. EFI GOP Options (Sheet 1 of 2)**

Fastboot Settings	Description
Disable Seamless Mode Set	The Seamless mode set feature ensures that on a properly configured embedded device there is only 1 mode set between power on and a fully functional system. Under normal circumstances a PC will set the mode several times during initialization which causes screen flicker and latency that is undesirable for an embedded device. With seamless mode set, the firmware sets the mode and the driver adopts the existing mode without altering the hardware state. This feature can be combined with splash screen or splash video for optimal effect. EFI and the EPOG feature do not support this feature.
Splash Screen	<p>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.</p> <p>The splash screen is limited to 50kb in size and JPG and BMP formats. For Quickboot, only BMP format is allowed.</p>
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)	<p>This feature provides an option to skip checking the EDID for optimizing the boot time.</p> <p><b>Note:</b> It is not applicable to EPOG.</p>
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	Splash Screen BG Color Red must be between 0x0 and 0xFF.
Splash Screen BG Color Green	Splash Screen BG Color Green must be between 0x0 and 0xFF.
Splash Screen BG Color Blue	Splash Screen BG Color Blue must be between 0x0 and 0xFF.
Splash Screen X (upper left corner x coordinate)	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)	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.
Splash Video Offset	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.



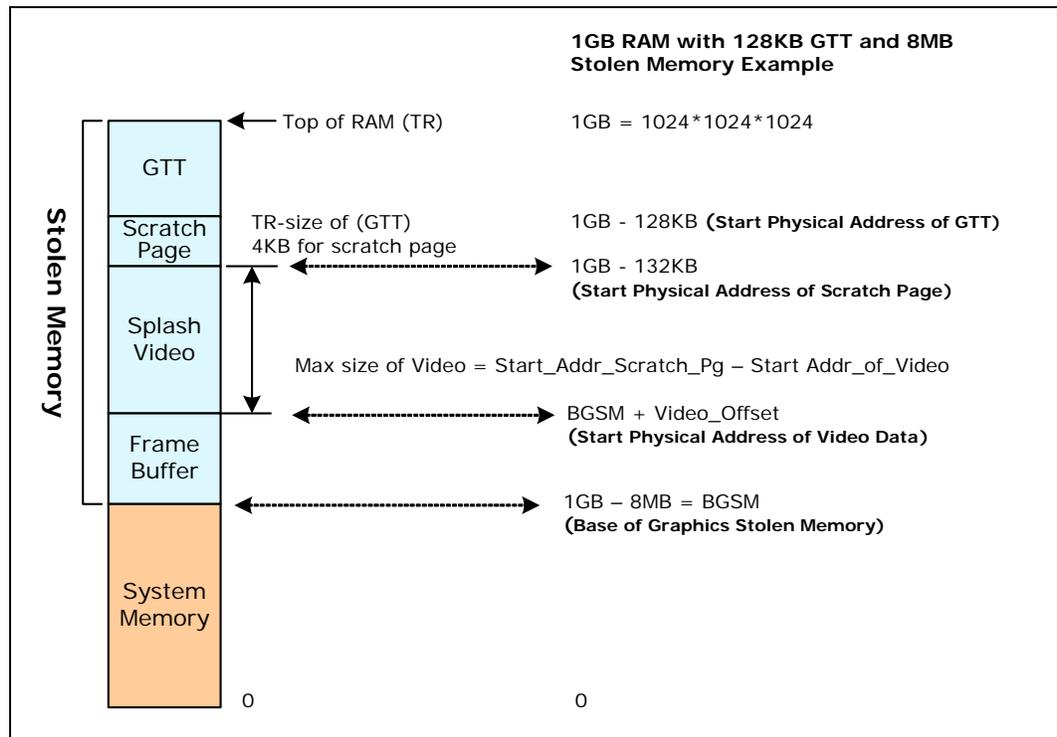
**Table 15. EFI GOP Options (Sheet 2 of 2)**

Fastboot Settings	Description
Splash Video Pixel Format	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	The width, in pixels, of the Splash Video image in memory.
Splash Video Source Height	The height, in pixels, of the Splash Video image in memory.
Splash Video Source Pitch	The pitch, in bytes, of the Splash Video image in memory. Pitch must be $\geq$ bytes per pixel * source width.
Splash Video Destination X	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	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	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	The width of the screen. This number must currently be the same as SrcWidth.

### 3.5.4.1 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:

$$\text{Start\_Phy\_Ram\_Addr} = \text{BGSM} + \text{Video\_Offset}$$

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.

To calculate BGSM, 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.

$$\text{BGSM} = 1 \text{ GB} - 8 \text{ MB} = \text{0x4000 0000} - \text{0x80 0000} = \text{0x3F80 0000}$$

### 3.5.4.2 How to Select the Video\_Offset

Determine the size of the maximum resolution of the framebuffer.

$$\text{Size} = \text{framebuffer\_height} * \text{framebuffer\_pitch}$$

where **framebuffer\_pitch** = framebuffer\_width \* Bytes\_per\_Pixel (page aligned)

For example, 1024x768 at 32-bit BPP:

$$\text{Size} = 768 * (1024 * 4) = 3145728 = \text{0x30 0000}$$

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

$$\text{Size} = 768 * (8192) = 6291456 = \text{0x60 0000}$$

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.

$$\text{Max Size of Splash Video} = \text{Size of Stolen Memory} - \text{Max Frame buffer size} - \text{Size of GTT} - \text{Size of Scratch Page (4 KB)}$$

#### Notes:

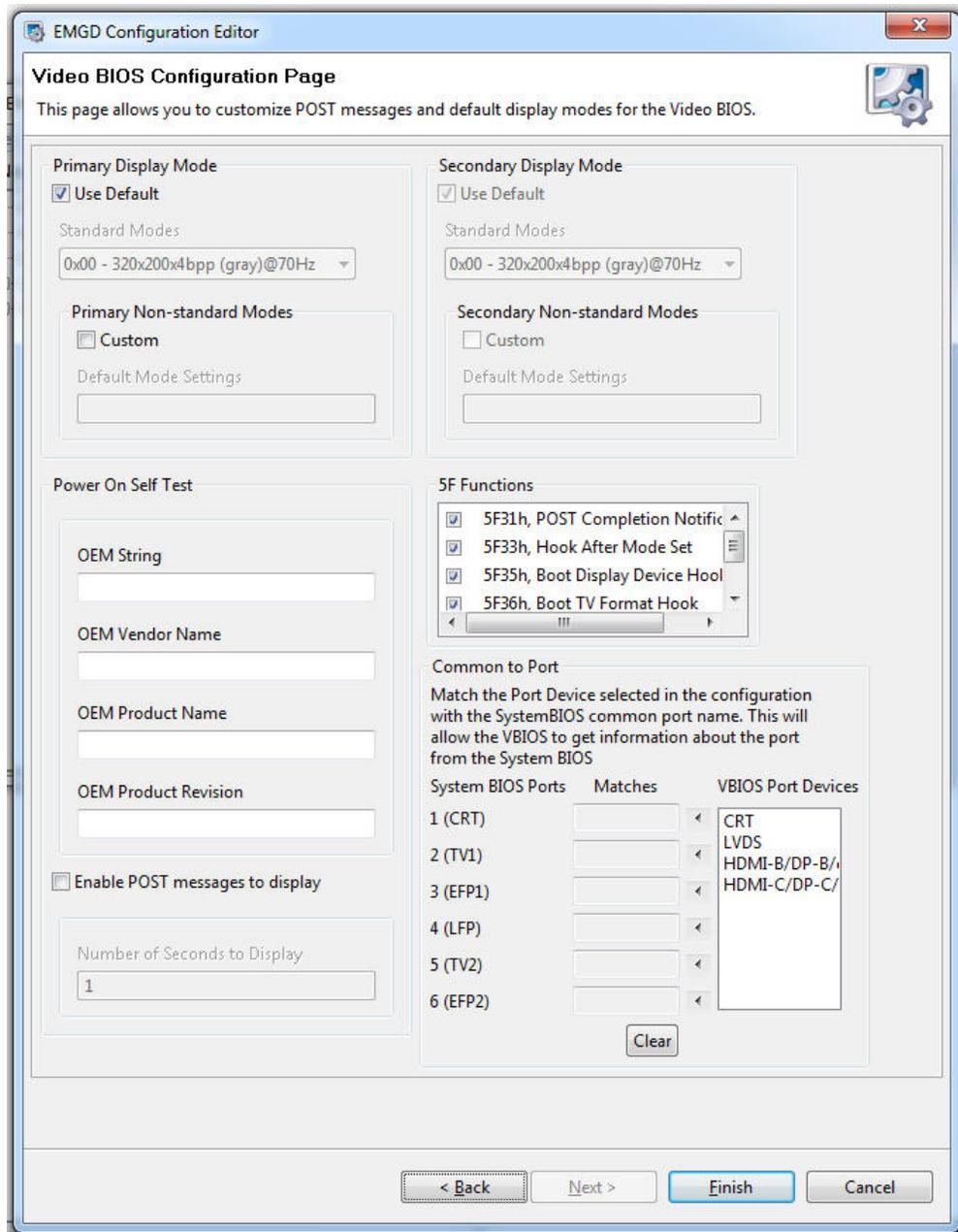
1. For the Splash Video option the stolen memory MUST be a minimum of **8 MB**. This is selected in the BIOS menu.
2. The recommended Video Offsets for the splash video are **0x600000** and **0x700000**.
3. If the Size of the Video frame is more than **1 MB**, please choose **0x600000**.

### 3.5.5 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 16. 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 <b>Custom</b> check box and enter the mode number in the box. For a complete list of customized VGA and VESA modes, refer to <a href="#">Table 25, “Supported VGA Video Display Modes” on page 72</a> and <a href="#">Table 26, “VESA Modes Supported by Video BIOS” on page 74</a> .
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 <b>Custom</b> check box and enter the mode number in the box. For a complete list of customized VGA and VESA modes, refer to <a href="#">Table 25, “Supported VGA Video Display Modes” on page 72</a> and <a href="#">Table 26, “VESA Modes Supported by Video BIOS” on page 74</a> .
5F Functions	These settings allow you to enable or disable the five System BIOS 15h interrupt hooks. (Please see <a href="#">Appendix C, “Intel® 5F Extended Interface Functions”</a> for more information on 5F functions.) All five functions are enabled by default.
Common to Port	<p>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). The VBIOS makes this association when the VBIOS calls the System BIOS Intel® 5F interrupt functions.</p> <p>This setting consists of six numbers, where each number is associated with one of the System BIOS displays:</p> <p>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</p> <p>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.</p> <p>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 <b>CRT</b> 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.</p> <p><b>Notes:</b> 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.</p> <p>The Display Detect field on the Chipset Configuration page must be set to <b>Enable</b> in order for the Common to Port values to be used.</p>
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 16. 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.

### 3.6 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 17. Intel® EMGD Package Editor Setting Options**

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	<p>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.</p> <p>To select a configuration, click the check box next to the configuration name. You can select all available configurations by clicking <b>Select All</b> located below the Configurations by block and clear all configurations by clicking <b>Clear All</b>.</p> <p>The Configuration Name column shows the name of each configuration and the Chipset column shows the chipset associated with each configuration.</p> <p>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.</p>
Default Configuration	<p>The Default Configuration list box allows you to select a default configuration from the configurations you selected in the Configurations block.</p> <p>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 <b>None</b>. See also <a href="#">Section 5.2.1, "Universal INF Configuration" on page 76</a>.</p>
Target OS	<p>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:</p> <ul style="list-style-type: none"> <li>• <a href="#">"Entering Windows OS Options" on page 42</a></li> <li>• <a href="#">"Generating a VBIOS Package" on page 44</a></li> <li>• <a href="#">"Entering EFI Options" on page 45</a></li> </ul>
Microsoft Windows Settings	<p>If you are creating a package for a Microsoft Windows* platform, click the <b>Microsoft Windows Settings</b> button for additional settings that may be required for your configuration. Please see <a href="#">"Entering Windows OS Options" on page 42</a> for descriptions of these settings.</p>
General Driver Windows Embedded Compact 7 Settings	<p>To specify general setting for Windows Embedded Compact 7, click this button. See <a href="#">Section 3.6.2, "Entering General Windows Embedded Compact 7 Options" on page 44</a>.</p>

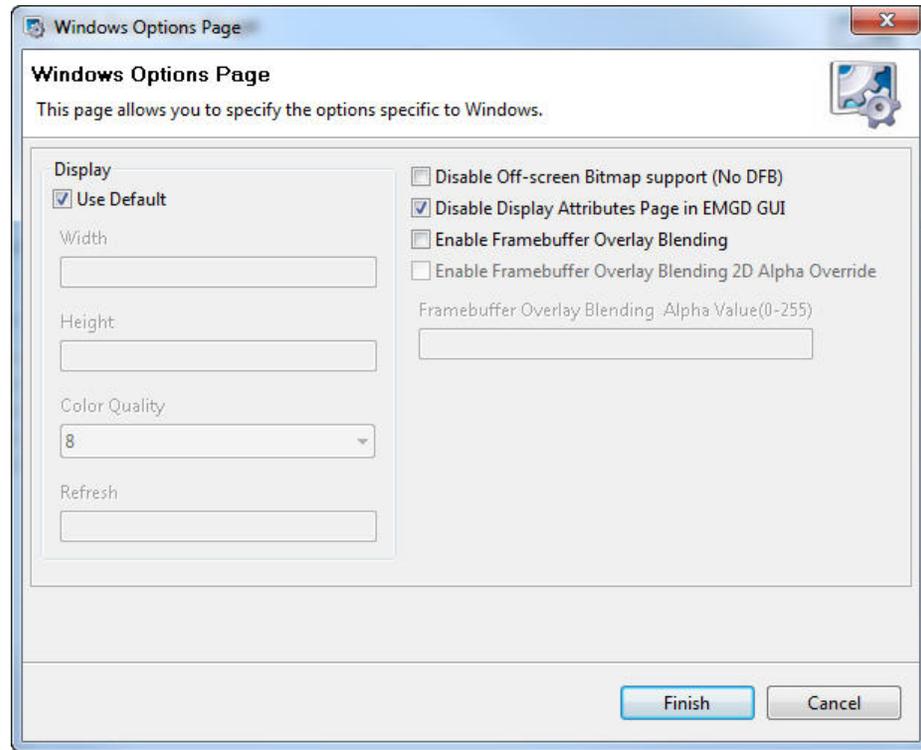
If you are not creating a VBIOS package, click **Finish**. When you click **Finish**, CED creates a package that can be used for generating an installation.

### 3.6.1 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 18. 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 <b>Use Default</b> check box or to select the default width, height, color quality, and refresh rate for the configuration.
Disable Off-Screen Bitmap support (No DFB)	Setting No DFB causes the driver to enable using off-screen Bitmap support potentially using more memory, but speeding up bit map handling. Default is enabled.
Disable Display Attributes Page in EMGD GUI	Some OEMs may choose to limit the features in the standard EMGD GUI interface. Setting this option removes the display attributes from the GUI presented to the end user. Default is this feature disabled, thus allowing full GUI features.

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

Windows OS Option	Description
Enable Framebuffer Overlay Blending	When checked, this option enables overlay blending with the framebuffer on both display outputs (if in VEXT mode) and when display mode resolution is 32-bit XRGB. This is used to enable some advanced “special effects” within the overlay plane at the expense of performance. Default is Blending in Overlay disabled for best performance of graphics/video in the overlay.
Enable Framebuffer Overlay Blending 2D Alpha Override	This option applies only to Windows XP. When checked, it enables an override to the frame buffer overlay blending 2D alpha.  <b>Note:</b> Checking the Framebuffer 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 <b>Enable Framebuffer Overlay Blending 2D Alpha Override</b> option and then enter the alpha value. This alpha override will cause performance impact when a lot of 2D blitting operations take place.
Framebuffer Overlay Blending Alpha Value	The valid range is from 0x00 to 0xFF (0-255).

### 3.6.2 Entering General Windows Embedded Compact 7 Options

The General Driver Windows Embedded Compact 7 Settings Page allows you to enter Windows Embedded Compact 7 OS-specific options into the configuration. When you click **General Driver Windows Embedded Compact 7 Settings** from the Intel® EMGD Package Page, the following page appears.

Figure 19. General Driver Windows Embedded Compact 7 Settings Page

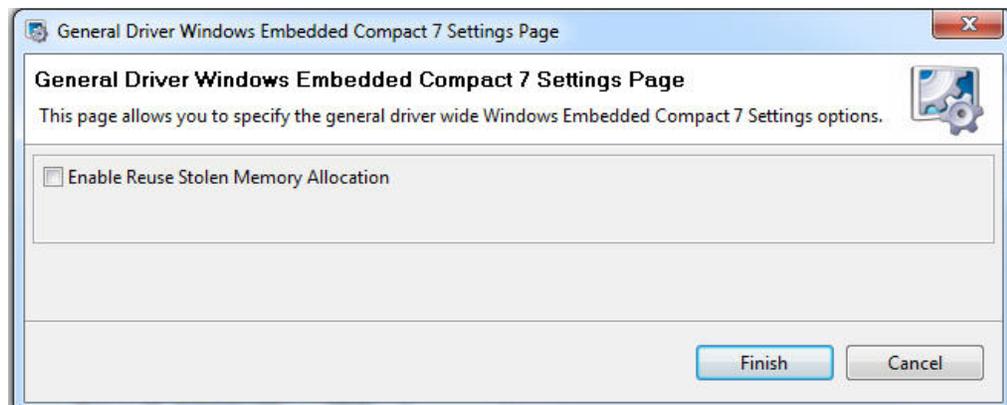


Table 19. Driver Windows Embedded Compact 7 Settings Options

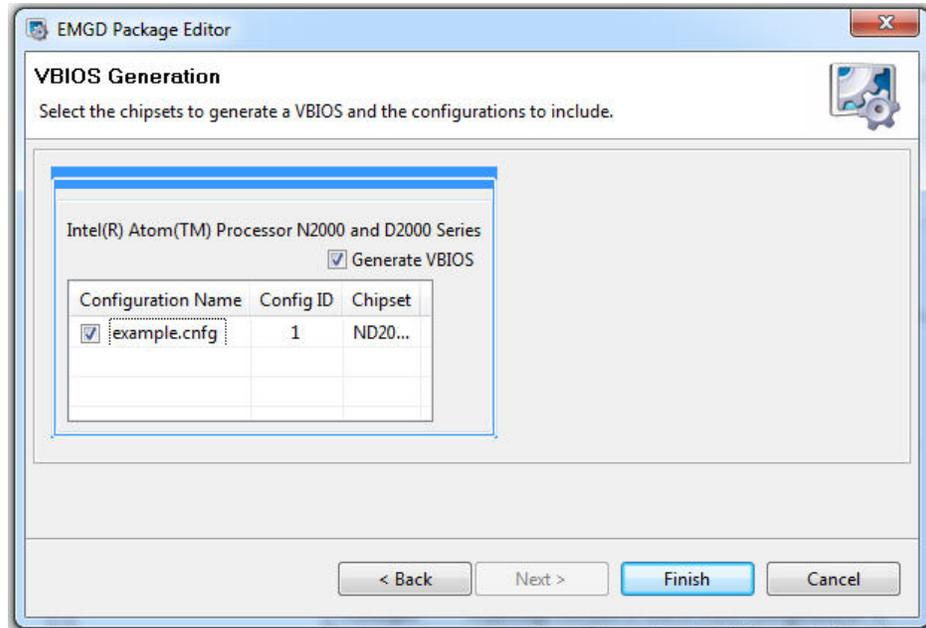
Windows Embedded Compact 7 OS Option	Description
Enable Reuse Stolen Memory Allocation	The dynamic memory option allows you to choose whether you want to use the memory stolen by the BIOS, or if you want to scrap that memory and re-allocate memory dynamically.

### 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

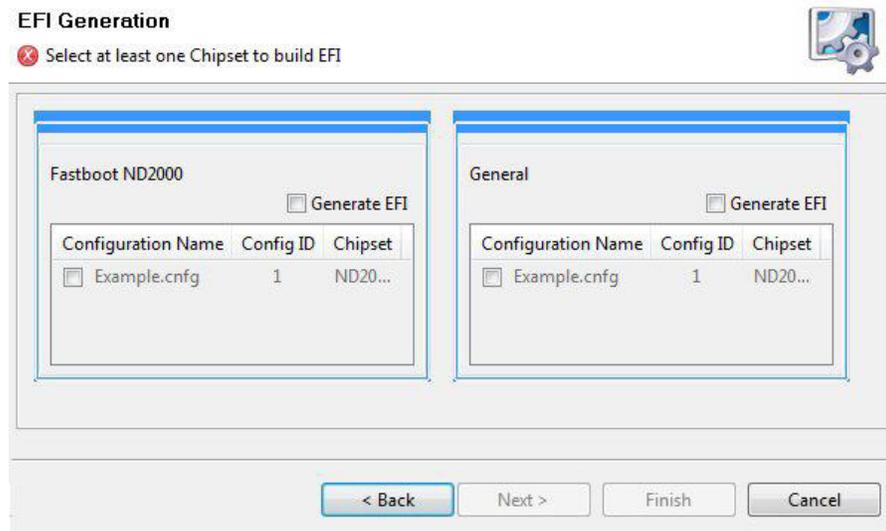


To generate a VBIOS, click the **Generate VBIOS** check box and select the configurations to include. After selecting the chipset and the configurations, click **Finish**. CED generates a package that includes both the OROM and the TSR for the chipsets and the configurations you selected.

### 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 `CBCHAxix.ROM` to `CBCHAxix_IEGD_EFI.ROM` where `xxx` = the release version of Standard BIOS  
OR  
Copy `CBFBAxix.ROM` to `CBFBAxix_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 `CBCHAxix_IEGD_EFI.ROM` or `CBFBAxix_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 `CBCHAxix.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.



### 3.7 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_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:

- “Configuring and Installing Microsoft Windows Drivers” on page 76
- “Configuring and Building Intel® EMGD for Microsoft Windows\* Embedded Compact 7” on page 89

### 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 20. GMCH Device 2, Function 1 BIOS Setting**

OS	Chipset
	Intel® Atom™ Processor N2000 and D2000 Series
Microsoft Windows* XP and Microsoft Windows XPe*	Disabled
Microsoft Windows* Embedded Compact 7	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 21. GMS Settings

Chipset	GMS Settings
Intel® Atom™ Processor N2000 and D2000 Series	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 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.

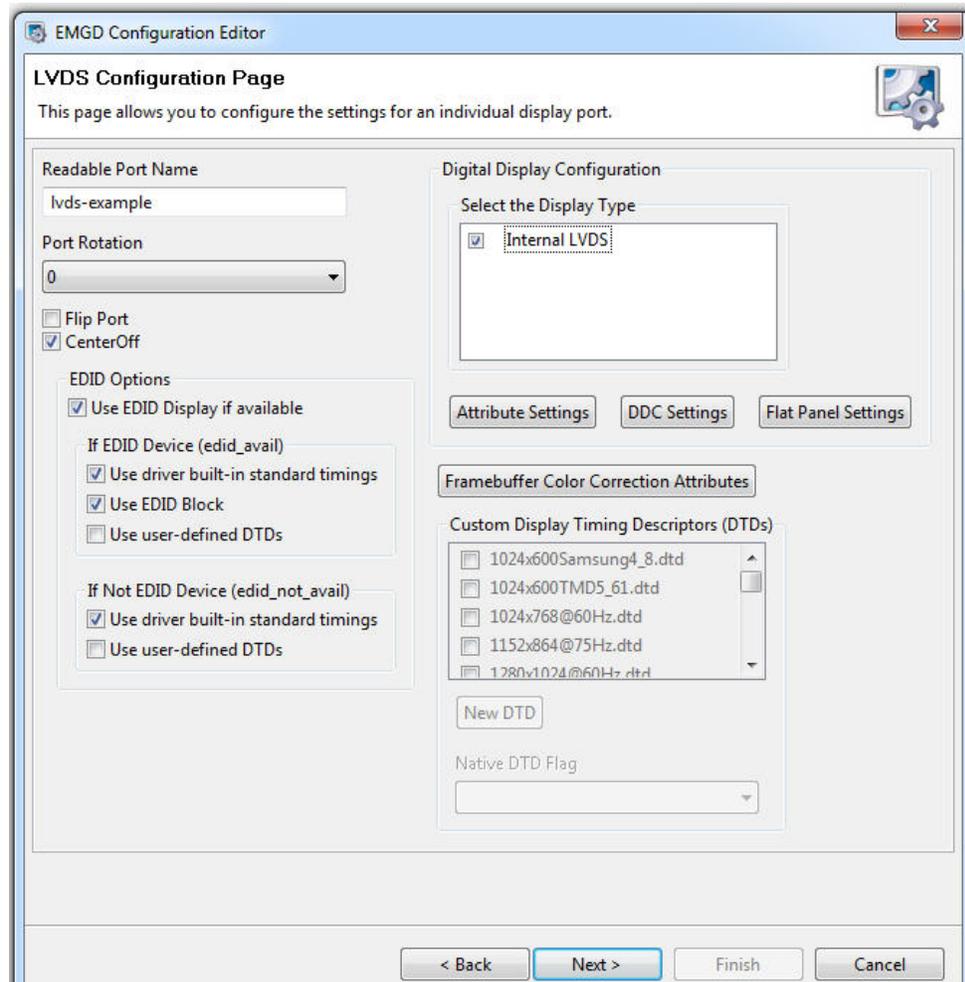
To create a configuration and configure the LVDS options:

1. Create a custom DTD as described in [Section 3.4, “Creating a New Customized DTD” on page 20](#).
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**.



- In the Custom Display Timing Descriptors (DTDs) list, select the DTD you created in Section 3.4, “Creating a New Customized DTD” on page 20 for example, *test\_LVDS*.

**Figure 22. LVDS Configuration Page**

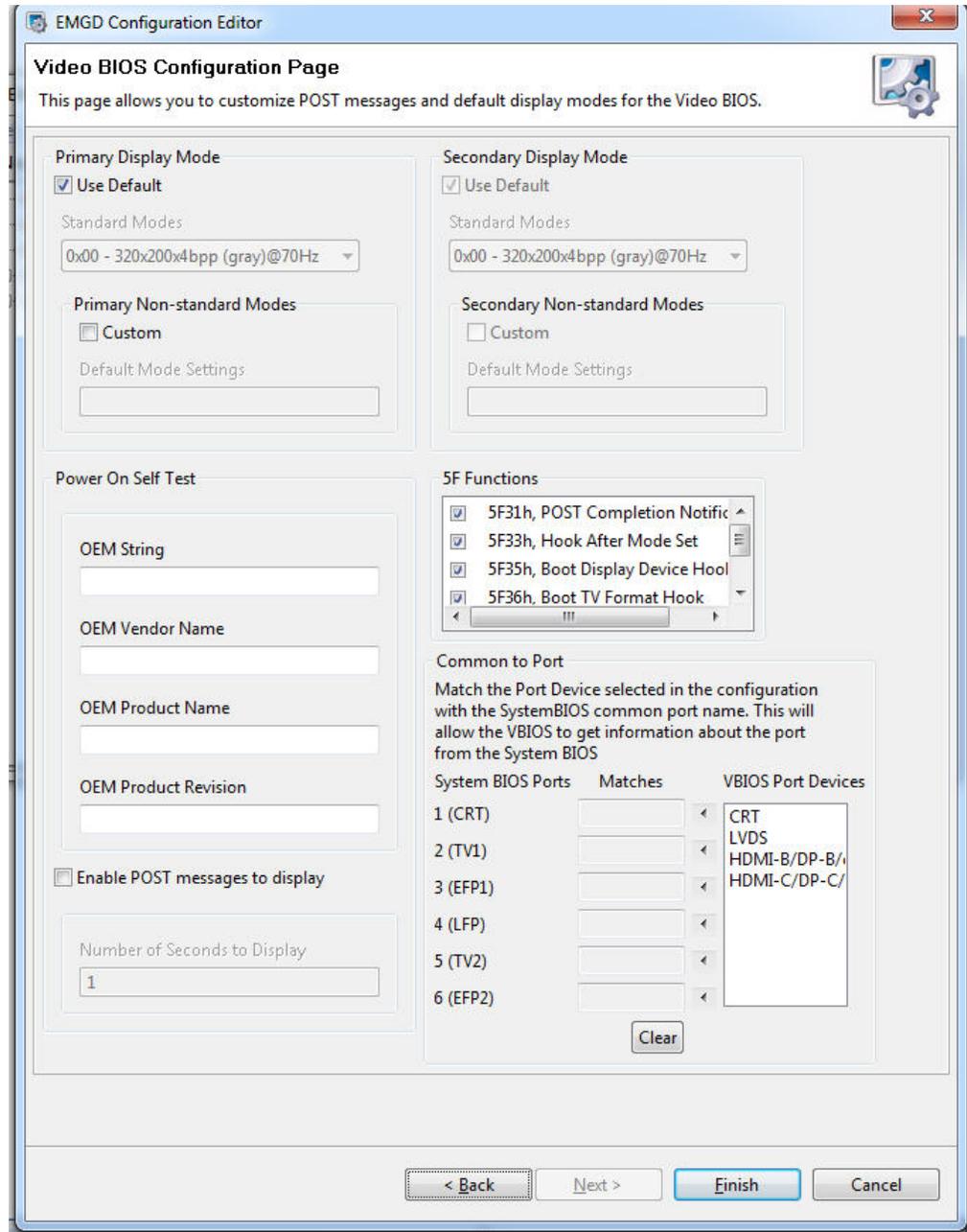


- Click **Next**.
- (Optional) Configure Fastboot options as described in “Configuring EFI GOP” on page 35.
- Click **Next**.

To set the custom mode:

- From the Intel® EMGD CED screen (similar to the example below), in the Primary Display Mode section, clear the **Use Default** checkbox.
- In the Primary Non-standard Modes section, select the checkbox for **Custom**.
- 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 22. 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 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., <b>Extended Desktop</b> in the <b>Display Properties</b> sheet within Microsoft Windows).



Table 22. Parameter Configuration Format (Sheet 2 of 7)

Name	Range/Value	Description
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 <a href="#">Section 3.12, "Display Detection and Initialization"</a> on page 58 for detailed information on this parameter.
PortOrder	PortOrder must be specified as a quoted string containing five digits. The valid values are: 2 - HDMI-B port 3 - HDMI-C port 4 - Integrated LVDS port 5 - CRT (VGA) 6 - DisplayPort-B (DP-B) 7 - DisplayPort-C (DP-C) 8 - embedded DisplayPort (eDP) 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 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 <a href="#">Table 35, "Default Search Order"</a> on page 118. Please see <a href="#">Section 3.12, "Display Detection and Initialization"</a> on page 58 for more information on using PortOrder in combination with the Display Detect feature.
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. <b>Note:</b> This parameter is only for Microsoft Windows*.
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.
FbBlendAlphaValue	The valid range is from 0x00 to 0xFF.	The Alpha value used for the frame buffer blend.
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 <a href="#">Section 3.5.5, "Configuring the Video BIOS and EFI"</a> on page 38 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
OverlayNoClip	0 - Off (Default) 1 - On	Enables/disables the colorkey feature.



Table 22. 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 22. Parameter Configuration Format (Sheet 4 of 7)

Name	Range/Value	Description															
int15	5 digits	<p>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)</p> <p>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</p> <p>(Please see <a href="#">Appendix C, "Intel® 5F Extended Interface Functions"</a> for more information on 5F functions.)</p> <p>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</p> <p>For example,            int15 = 11001</p> <p>Enables 5F31h, 5F33h, and 5F38h hooks only. The 5F35h and 5F36h hooks are disabled.</p> <p>The default is 11111, enable all five hooks.</p>															
port	2 - HDMI-B port 3 - HDMI-C 4 - Integrated LVDS port 5 - Analog 6 - DP-B 7 - DP-C 8 - eDP	Used to define port specific settings.															
rotation	<table border="1" data-bbox="574 1266 927 1444"> <thead> <tr> <th>Degrees</th> <th>Windows</th> <th>WEC</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0x00</td> </tr> <tr> <td>90</td> <td>90</td> <td>0x5A</td> </tr> <tr> <td>180</td> <td>180</td> <td>0xB4</td> </tr> <tr> <td>270</td> <td>270</td> <td>0x10E</td> </tr> </tbody> </table> <p>Default: 0 degrees</p>	Degrees	Windows	WEC	0	0	0x00	90	90	0x5A	180	180	0xB4	270	270	0x10E	Rotation of the display.
Degrees	Windows	WEC															
0	0	0x00															
90	90	0x5A															
180	180	0xB4															
270	270	0x10E															
flip	<table border="1" data-bbox="574 1554 927 1654"> <thead> <tr> <th>Flip</th> <th>Windows</th> <th>WEC</th> </tr> </thead> <tbody> <tr> <td>off</td> <td>0</td> <td>0x00</td> </tr> <tr> <td>on</td> <td>1</td> <td>0x01</td> </tr> </tbody> </table> <p>Default: off</p>	Flip	Windows	WEC	off	0	0x00	on	1	0x01	Flip of the display.						
Flip	Windows	WEC															
off	0	0x00															
on	1	0x01															



Table 22. Parameter Configuration Format (Sheet 5 of 7)

Name	Range/Value	Description
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.
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 <a href="#">Section 3.13, “Advanced EDID Configuration”</a> on page 60 for detailed information.
fpinfo		Panel-specific information.
bklmethod	Range [0-3] 0 – no backlight 1 – Port Driver 2 – GMCH 3 – ICH  <b>Note:</b> 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 -0xffff]. 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 clock/data active. Zero indicates no delay required.
bkltt2		(T2) Time delay between clock/data active and backlight enable.
bkltt3		(T3) Time delay between backlight disable and clock/data inactive.
bkltt4		(T4) Time delay between clock/data inactive and VDD inactive.
bkltt5		(T5) Minimum delay between VDD inactive, and active.
gpiopinvee	Valid ICH GPIO pin, 0 indexed	GPIO connection for panel power.
gpiopinvdd	For example: gpiopinvee = 3 gpiopinvee = 5 gpiopinenable = 1	GPIO connection for backlight power on/off sequencing signal.
gpiopinbkl		GPIO to enable backlight signal.



Table 22. Parameter Configuration Format (Sheet 6 of 7)

Name	Range/Value	Description
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 <code>Flags</code> 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 [0-0x7ffffff]	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.
v_size	Range 0-4096 [12 bits]	Vertical size. Currently not supported.



**Table 22. Parameter Configuration Format (Sheet 7 of 7)**

Name	Range/Value	Description
flags	Range [32 bits] Valid values: <u>bit 31</u> 0 - Non-interlaced 1 - Interlaced  <u>bit 27</u> 0 - vertical sync polarity active low 1 - vertical sync polarity active high  <u>bit 26</u> 0 - horizontal sync polarity active low 1 - horizontal sync polarity active high  <u>bit 25</u> 0 - blank sync polarity active high 1 - blank sync polarity active low  <u>bit 17</u> 0 - Normal DTD 1 - Panel/display Native DTD  <u>All other bits</u> 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 device for the port. See <a href="#">Appendix B, "Port Driver Attributes"</a> 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 <a href="#">Appendix B, "Port Driver Attributes"</a> .



## 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.

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 23](#) 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: `cdv` for Atom N2000/D2000 Series, etc.):

```
HKR, AII\<ConfigID>\General, DisplayDetect, %REG_DWORD%, 1
```

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

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), 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 1.15 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 v1.15 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, HDMI-B)  
 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 HDMI-B, so the Secondary display is "HDMI-B."
  - 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 "HDMI-B."
  - c. The driver cannot detect the presence of a display connected to the Internal LVDS. 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 other detectable ports or devices are connected, then put LVDS in the PortOrder after the other detectable ports. For example:  
 PortOrder = "24000" (HDMI-B, 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" (HDMI-B)  
 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

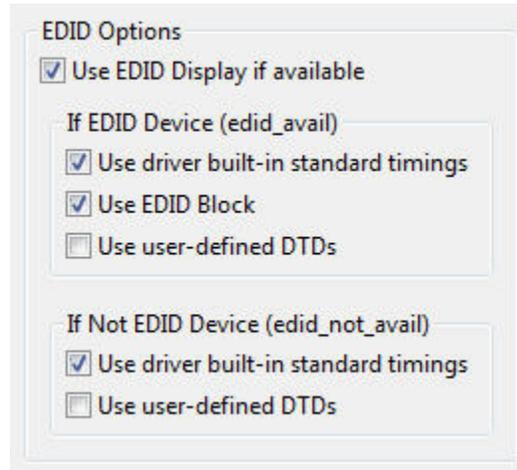
The table below provides a list of displays that are detectable by Intel® EMGD.

**Table 23. Detectable Displays**

Transmitter	Display Type	Detectable by Intel® EMGD?
Integrated LVDS	LVDS	No (assumed attached)
CH7022	VGA	Yes
Integrated DisplayPort	DP Display	Yes
Integrated eDP	eDP flat panel	No (assumed attached)

### 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, DisplayPort, LVDS, HDMI, etc. 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 22](#) 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 61](#).

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 24. Sample Advanced EDID Configurations**

Configurations	CED Settings	Description
<ol style="list-style-type: none"> <li>1. Use only filtered built-in and any EDID-DTDs when the display has EDID information</li> <li>2. Use all built-in modes when the display does not contain EDID information</li> </ol>	edid = 1 edid_avail = 3 edid_not_avail = 1	Default values.
<ol style="list-style-type: none"> <li>1. Use only filtered built-in modes and EDID-DTDs when the display has EDID.</li> <li>2. Use only user-DTDs otherwise.</li> </ol>	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.
<ol style="list-style-type: none"> <li>1. Use only user-DTDs regardless of connected display. (Typically used for a custom panel that only supports user-defined DTDs.)</li> <li>2. Use limited set of timings when a panel EDID is present, but the Intel® EMGD cannot read the EDID information.</li> </ol>	edid = 0 edid_avail = (any value) edid_not_avail = 4	Only user-DTDs defined in CED are used.
<ol style="list-style-type: none"> <li>1. Use EDID-DTDs for an EDID display.</li> <li>2. Use user-DTDs for a non-EDID display.</li> </ol>	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.
<ol style="list-style-type: none"> <li>1. Use only EDID-DTDs and user-DTDs for an EDID display.</li> <li>2. Use user-DTDs only for a non-EDID display.</li> </ol>	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 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 84).

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 is not available for the clone display.

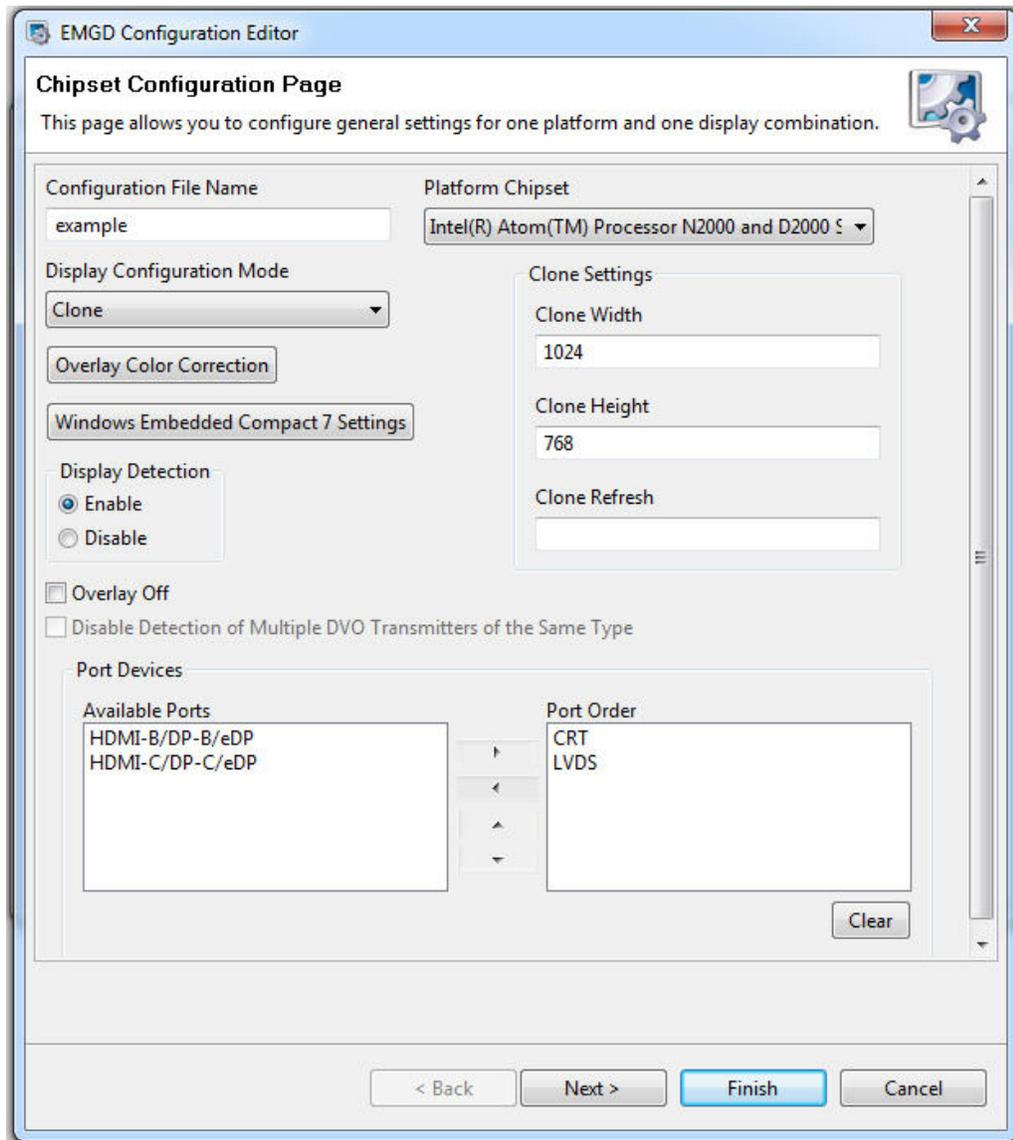


Extended Clone mode uses four CED parameters:

- 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

### 3.14.1 Clone Mode CED Configuration

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





## 3.15 Scaling and Centering Configurations

This release supports the following scaling and centering configurations:

- Internal LVDS Scaling With EDID Panels
- Alignment in Clone mode

See the following topics for configuration details:

- [“Internal LVDS Scaling with EDID Panels”](#)
- [“Centering Primary Display”](#)
- [“Alignment in Clone Mode”](#)

### 3.15.1 Internal LVDS Scaling with EDID Panels

The Intel® EMGD uses a user-supplied DTD with the native flag set (also known as native DTD) as native timing.

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.

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. 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 port drivers 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.

### 3.15.2 Centering Primary Display

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 3.15.3](#)), 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).

### 3.15.3 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 is one type of scaling in Clone mode, as described below.
  - *Hardware 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 3.15.2](#) explains how to force the primary display to center — thus allowing both displays to center.





## 4.0 Video BIOS 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 1.15 version of the VBIOS includes support for the following chipsets:

- Intel® Atom™ Processor N2000 and D2000 Series

Enabling the SMSW instructions used when Intel® EMGD VBIOS sets up its caching functions increases the boot speed during POST and system boot. 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 N2000 and D2000 Series

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)
  - `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 24](#)).

For further VBIOS build guidelines, see [Section 4.3.3, “Building the VBIOS”](#) on page 69.

See also the following topics:

- [“Selecting the Build Folder”](#)
- [“Configuring the Video BIOS”](#)
- [“Building the VBIOS”](#)

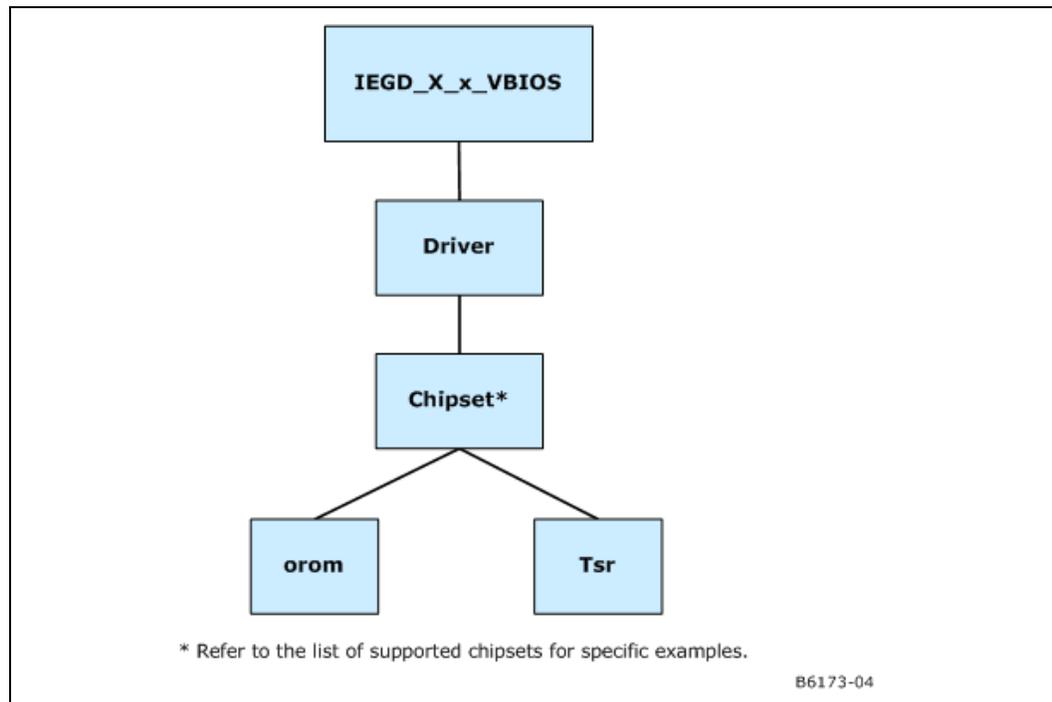
### 4.3.1 Selecting the Build Folder

The 1.15 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 24](#) shows the directory structure for the Video BIOS libraries contained within CED.



**Figure 24. 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, etc.). 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`

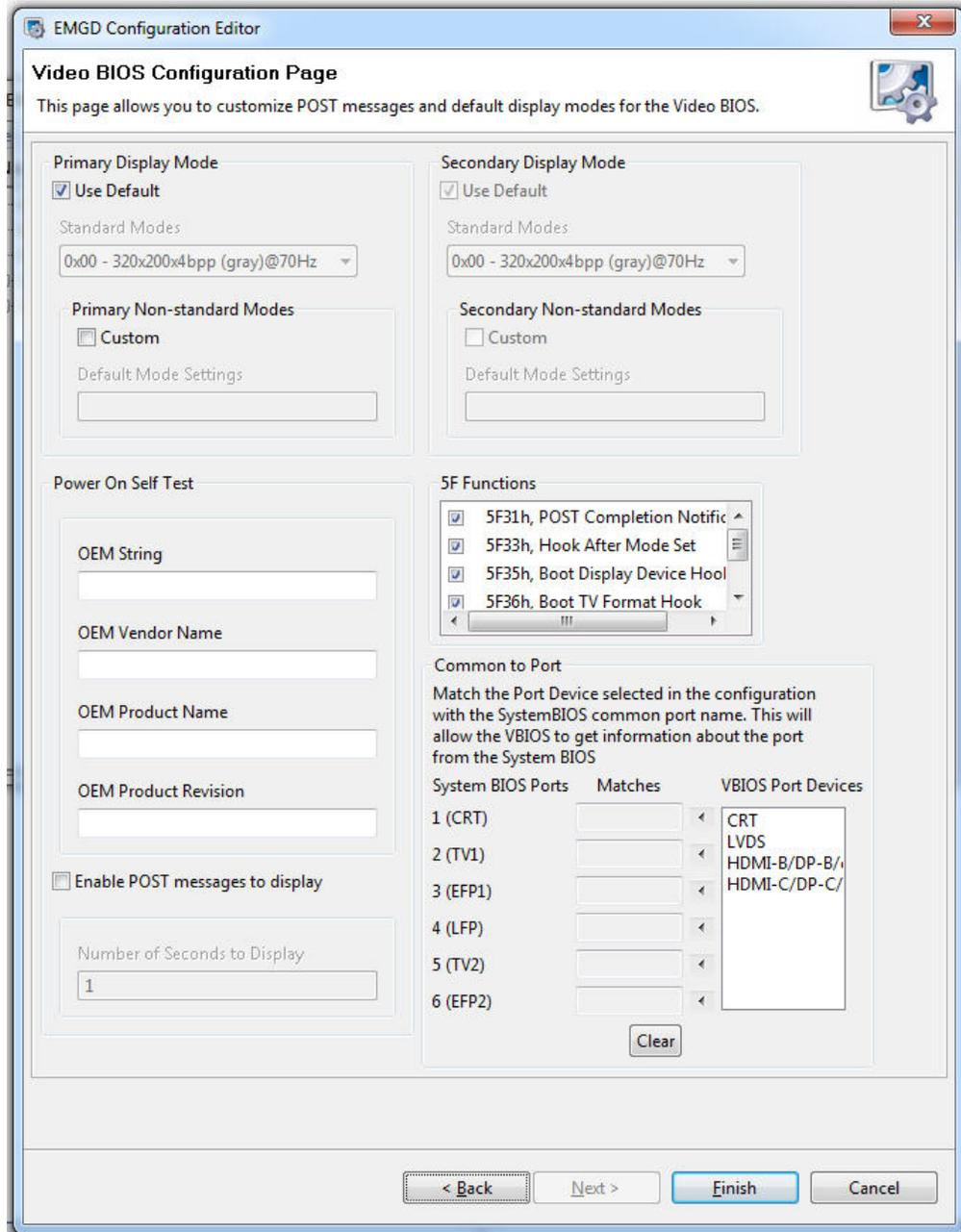
This setting allows you to specify which resolution to use for certain VGA modes. Because only one VGA mode can be supported on both displays, this setting applies to the primary display mode (`default_mode_0`). For example, mode 3 specifies three possible resolutions: 640x200, 640x350, and 720x400.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



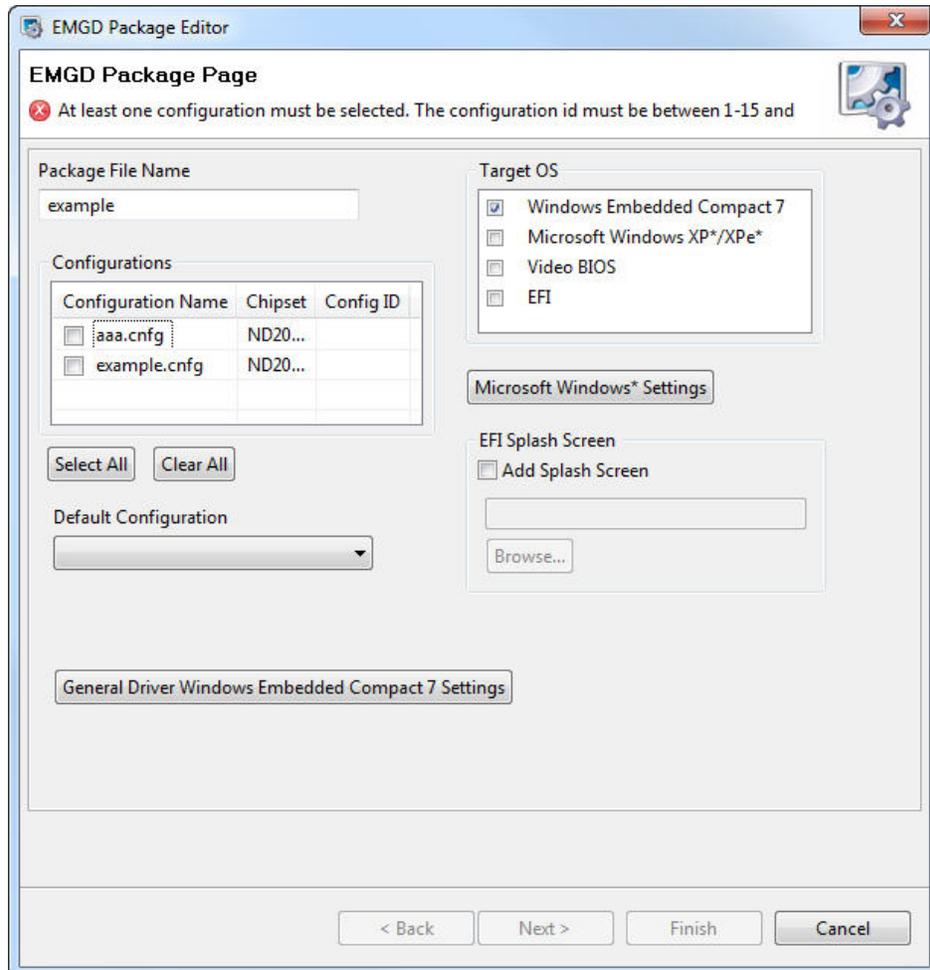
### 4.3.3 Building the VBIOS

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

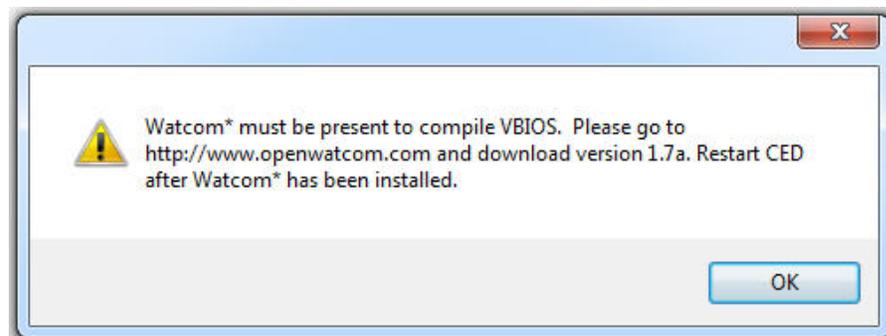
1. Define your configuration via CED, being sure to complete the Video BIOS Configuration Page.

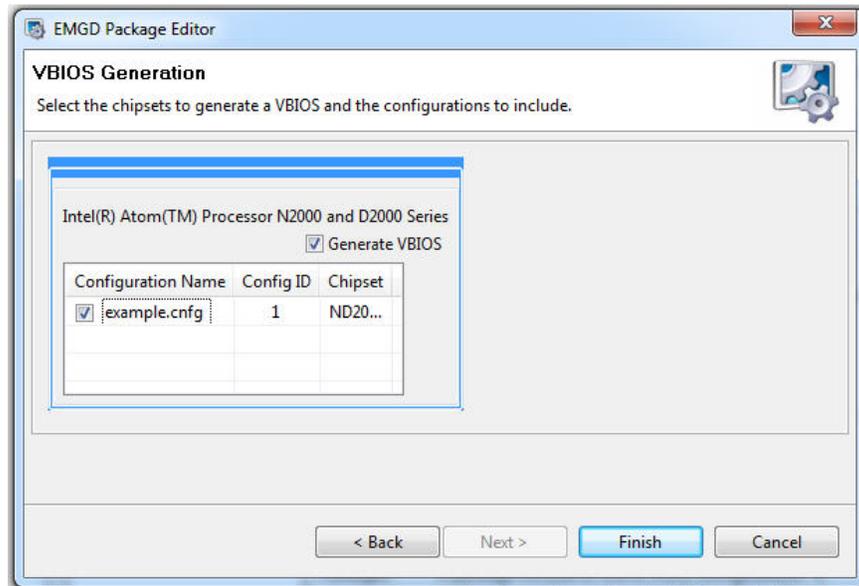


- When defining the package, be sure to select **Video BIOS** as Target OS.

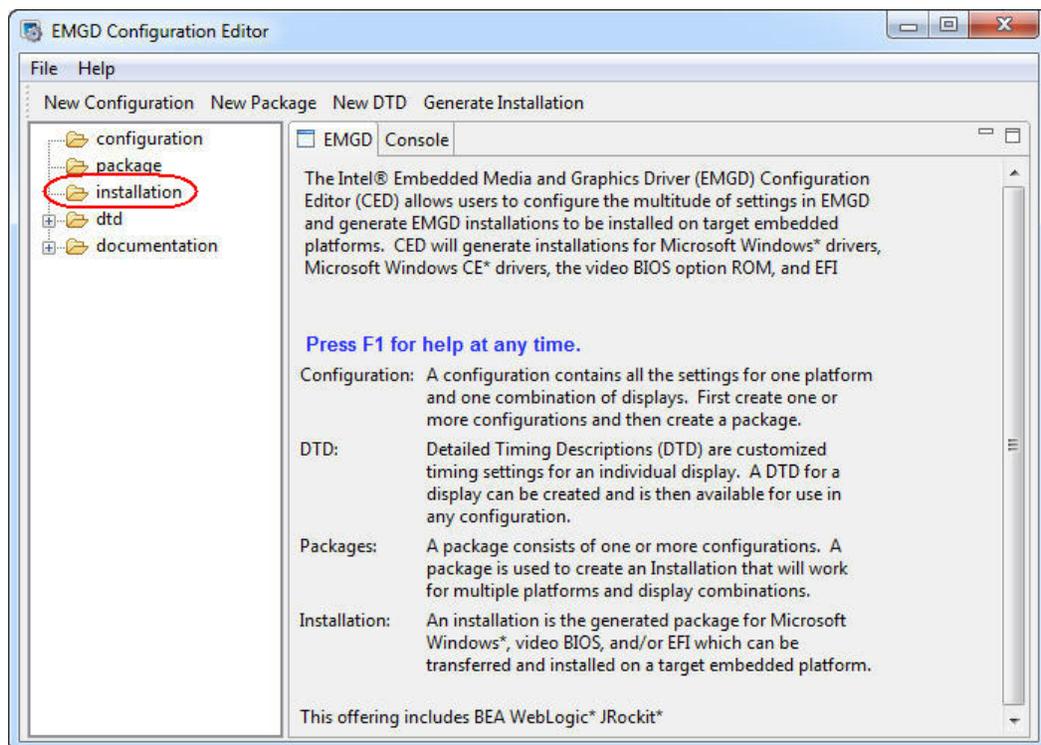


- Generate the installation. The following message will appear if the Open Watcom\* C/C++ compiler has not been installed on the user build system.





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 25](#) and [Table 26](#) 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 25. 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 25. 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 26. 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 26. 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 DOS.

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<sup>1</sup>:

- 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 41](#) 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 27](#) 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. The first number in the port order represents the primary display.

Notice the configuration limitations, for instance HDMI is only available on port B, and embedded DisplayPort (eDP) uses port C.

**Table 27. Dual Display Parameter Setting**

Dual Display Combination	Port Order
Internal LVDS and HDMI-B	42000
HDMI-B and Internal LVDS	24000
HDMI-B and Analog CRT	25000
HDMI-B and eDP	28000
HDMI-B and DP-C	27000
Internal LVDS and Analog CRT	45000
Internal LVDS and DP-B	46000
Internal LVDS and DP-C	47000
Analog CRT and Internal LVDS	54000
Analog CRT and DP-B	56000
Analog CRT and DP-C	57000
Analog CRT and HDMI-B	52000
Analog CRT and Internal LVDS	54000
DP-B and Internal LVDS	64000
DP-B and DP-C	67000
DP-B and eDP	68000
DP-C and Internal LVDS	74000

### 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 22](#) 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_SoftwareDeviceSettings_cdv]
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_N2000/D2000"
HKR, ALL\1\General, DisplayConfig, %REG_DWORD%, 1
HKR, ALL\1\General, DisplayDetect, %REG_DWORD%, 1
HKR, ALL\1\General, PortOrder, %REG_SZ%, "52678400"
HKR, ALL\1\General, DxvaOptions, %REG_DWORD%, 0
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\5\General, name, %REG_SZ%, "ANALOG"
HKR, ALL\1\Port\5\General, Rotation, %REG_DWORD%, 0
HKR, ALL\1\Port\5\General, Edid, %REG_DWORD%, 1
HKR, ALL\1\Port\5\General, EdidAvail, %REG_DWORD%, 3
HKR, ALL\1\Port\5\General, EdidNotAvail, %REG_DWORD%, 4
HKR, ALL\1\Port\5\General, CenterOff, %REG_DWORD%, 1
HKR, ALL\1\Port\2\General, name, %REG_SZ%, "HDMI-B"
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\1\Port\6\General, name, %REG_SZ%, "DP-B"
HKR, ALL\1\Port\6\General, Rotation, %REG_DWORD%, 0
HKR, ALL\1\Port\6\General, Edid, %REG_DWORD%, 1
HKR, ALL\1\Port\6\General, EdidAvail, %REG_DWORD%, 3
HKR, ALL\1\Port\6\General, EdidNotAvail, %REG_DWORD%, 4
HKR, ALL\1\Port\6\General, CenterOff, %REG_DWORD%, 1
HKR, ALL\1\Port\7\General, name, %REG_SZ%, "DP-C"
HKR, ALL\1\Port\7\General, Rotation, %REG_DWORD%, 0
HKR, ALL\1\Port\7\General, Edid, %REG_DWORD%, 1
HKR, ALL\1\Port\7\General, EdidAvail, %REG_DWORD%, 3
HKR, ALL\1\Port\7\General, EdidNotAvail, %REG_DWORD%, 4
HKR, ALL\1\Port\7\General, CenterOff, %REG_DWORD%, 1
HKR, ALL\1\Port\8\General, name, %REG_SZ%, "eDP"
HKR, ALL\1\Port\8\General, Rotation, %REG_DWORD%, 0
HKR, ALL\1\Port\8\General, CenterOff, %REG_DWORD%, 1
```



```

HKR, ALL\1\Port\8\General , Edid, %REG_DWORD%, 1
HKR, ALL\1\Port\8\General , EdidAvail, %REG_DWORD%, 3
HKR, ALL\1\Port\8\General , EdidNotAvail, %REG_DWORD%, 4
HKR, ALL\1\Port\8\FpInfo , bkltmethod, %REG_DWORD%, 1
HKR, ALL\1\Port\8\FpInfo , BkltT1, %REG_DWORD%, 200
HKR, ALL\1\Port\8\FpInfo , BkltT2, %REG_DWORD%, 1
HKR, ALL\1\Port\8\FpInfo , BkltT3, %REG_DWORD%, 200
HKR, ALL\1\Port\8\FpInfo , BkltT4, %REG_DWORD%, 50
HKR, ALL\1\Port\8\FpInfo , BkltT5, %REG_DWORD%, 500
HKR, ALL\1\Port\8\Dtd\1 , PixelClock, %REG_DWORD%, 69300
HKR, ALL\1\Port\8\Dtd\1 , HorzActive, %REG_DWORD%, 1366
HKR, ALL\1\Port\8\Dtd\1 , HorzSync, %REG_DWORD%, 48
HKR, ALL\1\Port\8\Dtd\1 , HorzSyncPulse, %REG_DWORD%, 32
HKR, ALL\1\Port\8\Dtd\1 , HorzBlank, %REG_DWORD%, 160
HKR, ALL\1\Port\8\Dtd\1 , VertActive, %REG_DWORD%, 768
HKR, ALL\1\Port\8\Dtd\1 , VertSync, %REG_DWORD%, 3
HKR, ALL\1\Port\8\Dtd\1 , VertSyncPulse, %REG_DWORD%, 5
HKR, ALL\1\Port\8\Dtd\1 , VertBlank, %REG_DWORD%, 22
HKR, ALL\1\Port\8\Dtd\1 , Flags, %REG_DWORD%, 0x20000

HKR,, No_D3D, %REG_DWORD%, 1

HKR,, PortDrivers, %REG_SZ%, "analog lvds hdmi dp"

;-----
[iegd_ICDSoftwareSettings]
HKLM, "SOFTWARE\Microsoft\Windows NT\CurrentVersion\OpenGLDrivers\iegd", DLL,
%REG_SZ%, iegdglga
HKLM, "SOFTWARE\Microsoft\Windows NT\CurrentVersion\OpenGLDrivers\iegd",
DriverVersion, %REG_DWORD%,

0x00000001
HKLM, "SOFTWARE\Microsoft\Windows NT\CurrentVersion\OpenGLDrivers\iegd", Flags,
%REG_DWORD%,

0x00000001
HKLM, "SOFTWARE\Microsoft\Windows NT\CurrentVersion\OpenGLDrivers\iegd",
Version, %REG_DWORD%,

0x00000002

;=====
[Strings]

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

iCDV0="Atom™ N2000/D2000 Series Embedded Media and Graphics Driver"
iCDV1="Atom™ N2000/D2000 Series Embedded Media and Graphics Driver"
iCDV2="Atom™ N2000/D2000 Series Embedded Media and Graphics Driver"
iCDV3="Atom™ N2000/D2000 Series Embedded Media and Graphics Driver"

;-----
; 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 = 0x0
```

## 5.2.5 Dynamic Port Driver Configuration

The Intel® Embedded Media and Graphics Driver supports many combinations and detection orders of displays. The support for these ports is 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_cdv]
analog.sys
lvds.sys
hdmi.sys
dp.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
iegdccagt.cpa = 1
iegdccagt.vp = 1
iegdccdis.dll = 1
analog.sys = 1
analog.vp = 1
lvds.sys = 1
lvds.vp = 1
hdmi.sys = 1
hdmi.vp = 1
dp.sys = 1
dp.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\_cdv] block:  
 HKR,, PortDrivers, %REG\_SZ%, "analog lvds hdmi dp"

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 IEGD 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 Intel® Embedded Media and Graphics Driver on Microsoft Windows

You can install and uninstall 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 Intel® Embedded Media and Graphics Driver. [Section 5.4, “Uninstalling the Current Version of the Driver” on page 83](#) provides instructions for uninstalling the current version of 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 automatic reboot without the reboot dialog box stopping the installation, use the option `"/nr"` following the `setup.exe` command, for example, **setup.exe /nr**. The end user will be responsible to do their own reboot.

## 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:

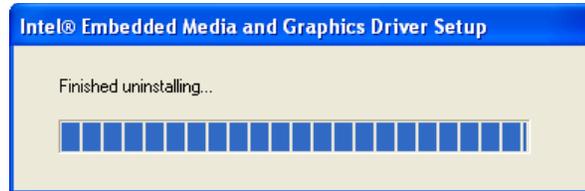
- The Intel® Embedded Media and Graphics Driver
- The `.inf` and `.pnf` files from the `windows\inf` folder.
- The `DisplayPage.dll` and `qt-mt332.dll` from the `windows\system32` folder
- Data registry items by running `regsvr32.exe` with the `uninstall` option.

**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**.

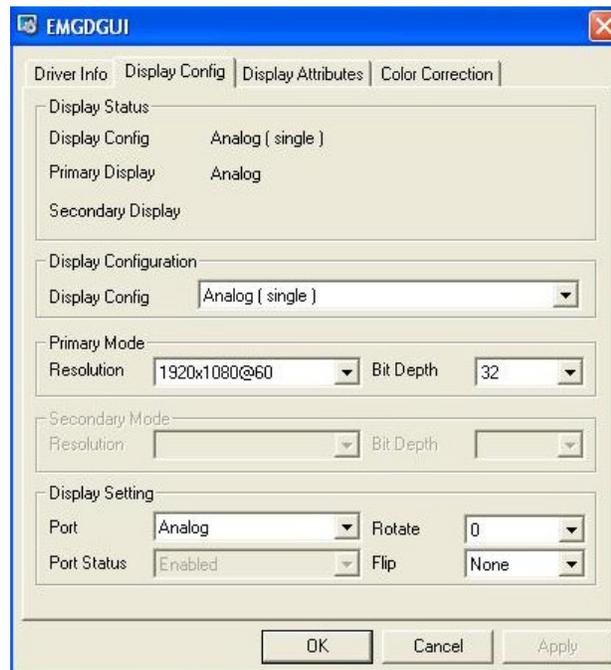


Figure 25. Example Runtime Configuration GUI — Driver Info Tab



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

Figure 26. Example Runtime Configuration GUI — Display Config Tab





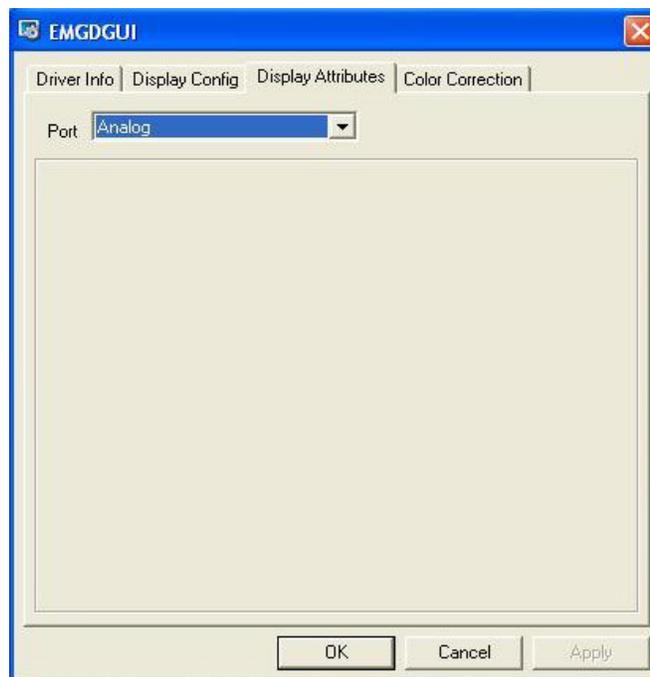
The **Display Status** section of the above dialog shows the current configuration for the **Primary** and **Secondary** displays.

3. 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.
4. 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.
5. 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.

6. 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 27. Example Runtime Configuration GUI — Display Attributes Tab**



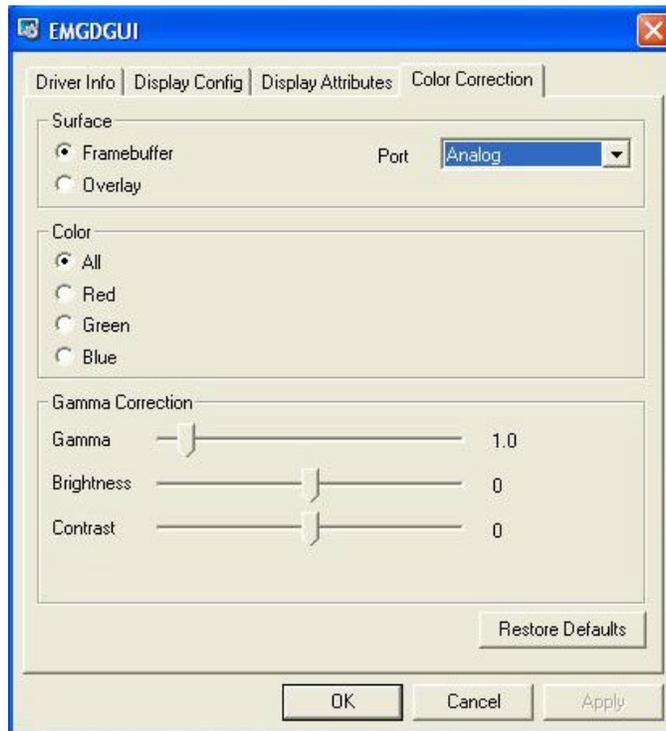


The figure above shows the attributes that can be changed for the selected port in the **Port** drop-down list. You can change the Port Driver by selecting the appropriate one for your device. The attributes that appear on this tab depend upon the selected port driver. Please see [Appendix B, "Port Driver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

*Note:* This tab can be set to NOT display with a configuration option in CED! See CED configuration options for details.

7. Click the **Color Correction** tab to view and change color corrections. [Figure 28](#) shows a sample Color Correction tab screen.  
Color Correction is available for both overlays and framebuffer.

**Figure 28. Example Runtime Configuration GUI — Color Correction Tab**



**Table 28. 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 29. 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 3.2 Alpha PPR (see [Section 6.1.2.1, "Installation and Setup" on page 90](#) for download location)

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

- Intel® Atom™ Processor N2000 and D2000 Series

*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 software.

#### 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); v3.2 Alpha PPR 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 from the Intel EDC website: [edc.intel.com](http://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
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 WEC7\_EMGD\_DRIVER=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:

```
ENDIF BSP_DISPLAY_RAGEXL
; include the path to the iegd.reg file in the release
package
;-----
; IEGD/EMGD
;-----
; [HKEY_LOCAL_MACHINE\System\GDI\DisplayCandidates]
;"Candidate3"="Drivers\Display\Intel"
```



```

;#include $(PLATFORM_DRIVERS_DIR)\EMGD\iegd.reg
;#include
"C:\WINCE700\platform\INTEL_CS\SRC\DRIVERS\CDV_WEC7\Driver\
emgd.reg"
;#include
C:\WINCE700\platform\INTEL_CS\SRC\DRIVERS\CDV_WEC7\Driver\
Codecs\emgd_filters.reg
ENDIF BSP_NODISPLAY !

```

Add in the following code:

```

IF WEC7_EMGD_DRIVER
[KEY_LOCAL_MACHINE\System\GDI\DisplayCandidates]
"Candidate3"="Drivers\Display\Intel"
#include
"C:\WINCE700\platform\INTEL_CS\SRC\DRIVERS\CDV_WEC7\Driver\
emgd.reg"
;#include
C:\WINCE700\platform\INTEL_CS\SRC\DRIVERS\CDV_WEC7\Driver\
Codecs\emgd_filters.reg

ENDIF WEC7_EMGD_DRIVER

```

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 WEC7_EMGD_DRIVER
ddi_emgd.dll
$( _TARGETPLATROOT )\SRC\DRIVERS\CDV_WEC7\Driver\ddi_emgd.dll
NK SHK
;isr_emgd.dll
$( _TARGETPLATROOT )\SRC\DRIVERS\CDV_WEC7\Driver\isr_emgd.dll
NK SHK
analog.dll
$( _TARGETPLATROOT )\SRC\DRIVERS\CDV_WEC7\Driver\analog.dll
NK SHK
lvds.dll
$( _TARGETPLATROOT )\SRC\DRIVERS\CDV_WEC7\Driver\lvds.dll
NK SHK
hdmi.dll
$( _TARGETPLATROOT )\SRC\DRIVERS\CDV_WEC7\Driver\hdmi.dll
NK SHK
dp.dll
$( _TARGETPLATROOT )\SRC\DRIVERS\CDV_WEC7\Driver\dp.dll
NK SHK

; aac_dec_filter.dll
$( _TARGETPLATROOT )\SRC\DRIVERS\CDV_WEC7\Driver\Codecs\aac_dec_
filter.dll NK SH

```



```
; ac3_dec_filter.dll
$( _TARGETPLATROOT ) \SRC \DRIVERS \CDV_WEC7 \Driver \Codecs \ac3_dec_
filter.dll      NK SH
; h264_dec_filter.dll
$( _TARGETPLATROOT ) \SRC \DRIVERS \CDV_WEC7 \Driver \Codecs \h264_dec_
_filter.dll      NK SH
; mp3_dec_filter.dll
$( _TARGETPLATROOT ) \SRC \DRIVERS \CDV_WEC7 \Driver \Codecs \mp3_dec_
filter.dll      NK SH
; mpeg2_dec_filter.dll
$( _TARGETPLATROOT ) \SRC \DRIVERS \CDV_WEC7 \Driver \Codecs \mpeg2_de
c_filter.dll      NK SH
; mpeg2_spl_filter.dll
$( _TARGETPLATROOT ) \SRC \DRIVERS \CDV_WEC7 \Driver \Codecs \mpeg2_sp
l_filter.dll      NK SH
; mpeg4_dec_filted.dll
$( _TARGETPLATROOT ) \SRC \DRIVERS \CDV_WEC7 \Driver \Codecs \mpeg4_de
c_filter.dll      NK SH
; mpeg4_spl_filter.dll
$( _TARGETPLATROOT ) \SRC \DRIVERS \CDV_WEC7 \Driver \Codecs \mpeg4_sp
l_filter.dll      NK SH

ENDIF WEC7_EMGD_DRIVER
```

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\_MULTIMON
  - 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. VPlayer does not support video resizing or drag-and-drop functionality.

### 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 90](#).



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.
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.
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 90](#) 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 90](#)), 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 30. [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.	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.

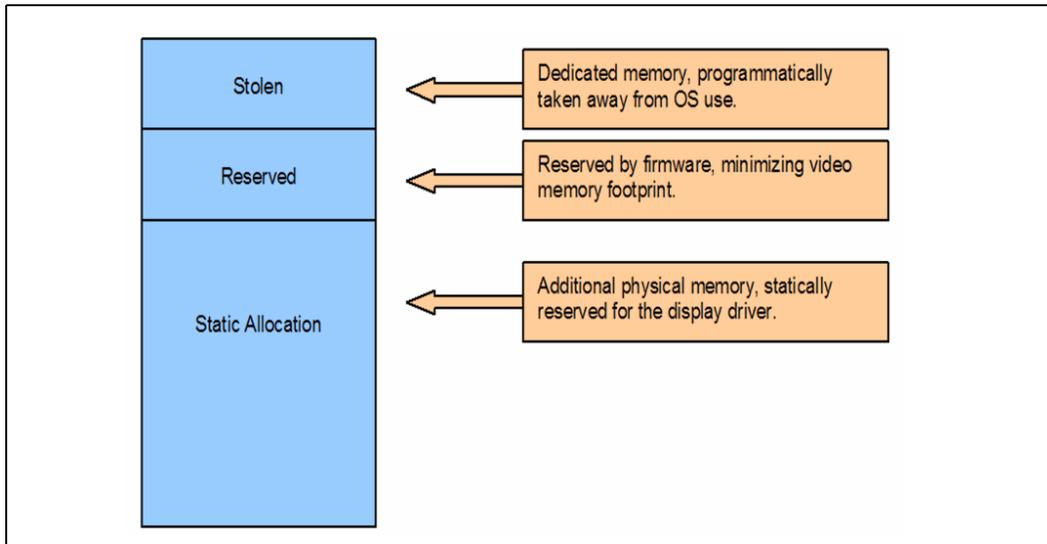
*Note:* Reserved memory is only available on the Microsoft Windows Embedded Compact 7 operating system, and must be accounted for in the `config.bib` memory layout file.

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 96](#) for further details on how to configure the static memory model.

[Figure 29](#) shows a typical memory map, using a static memory model.

Figure 29. 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.



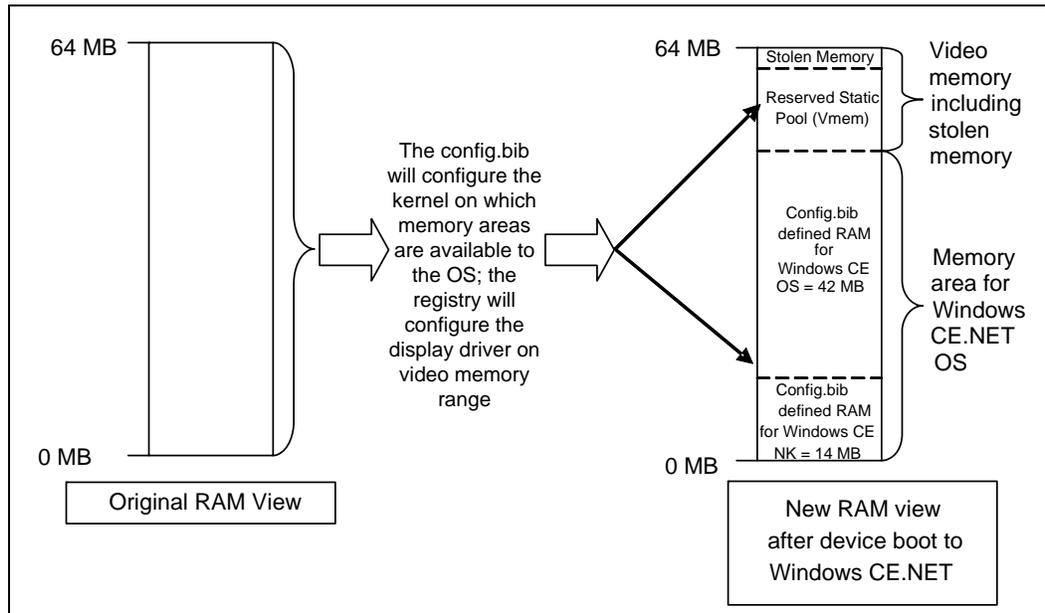
Besides the registry entry, the Platform Builder working project also needs to be updated to ensure that the kernel does not try to access this stolen memory. Two items in the config.bib of the project workspace need editing: the NK image/RAM memory partitioning and the memory reservation list. Using 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_N2000/
D2000. 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 31. [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 <i>ReservedMemoryBase</i> and <i>ReservedMemorySize</i> are non-zero, then Video memory allocation uses the static model.	The <i>ReservedMemoryBase</i> plus the <i>ReservedMemorySize</i> 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 31. [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. 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 <a href="#">Section 6.2.3.1</a> .



### 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 32. PortOrder Information

Port Number	Chipsets
2	HDMI-B
3	HDMI-C
4	Internal LVDS Port
5	Analog
6	DP-B
7	DP-C
8	eDP

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 HDMI-B 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 HDMI-B 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 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 - HDMI-B port
; 4 - HDMI-C port
; 5 - Analog port
; 6 - DP-B
; 7 - DP-C

```



### 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...
- 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 6.2.6, “Sample emgd.reg File” on page 106](#) 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 Chronitel 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 22, "Parameter Configuration Format" on page 51](#) 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 Miscellaneous Configuration Options

This section covers registry settings not in [HKEY\_LOCAL\_MACHINE\Drivers\Display\Intel].

### 6.2.5.1 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.

## 6.2.6 Sample emgd.reg File

```
;***** BEGIN INTEL DISPLAY DRIVER REGISTRY ENTRY *****
;*****
; This file was created based on user variable specified in the CED.
; DriverVer=
;*****

[HKEY_LOCAL_MACHINE\System\GDI\Drivers]
    "Display"="ddi_emgd.dll"
; [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]
    "PcfVersion"=dword:700
    "ConfigId"=dword:1

;*****
; The Following Sections Provide Per-Config
; & Per-Port Registry Settings
;*****

[HKEY_LOCAL_MACHINE\Drivers\Display\Intel\ALL\1]
    "name"="Sample"
[HKEY_LOCAL_MACHINE\Drivers\Display\Intel\ALL\1\General]
;-----
; Select Display configuration, single, twin ...
```



```

;-----
"DisplayConfig"=dword:1
;-----
; Select if you want to enable Display Detection
;-----
"DisplayDetect"=dword:1
;-----
; Select Port Order
;-----
"PortOrder"="52740000"
"Width"=dword:400
"Height"=dword:300
"Depth"=dword:20
"Refresh"=dword:3c
"MaxFbSize"=dword:800000
"MinVidSurfX"=dword:10
"MinVidSurfY"=dword:10
"OverlayDualVext"=dword:0
"NO_D3D"=dword:0
;-----
; Config 1 - Analog Port
;-----
; Following are the registry
; entries for port's general config
;-----
[HKEY_LOCAL_MACHINE\Drivers\Display\Intel\ALL\1\Port\5\General]
"Name"="Analog"
"Edid"=dword:1
"EdidAvail"=dword:3 ; STD TIMINGS + EDID TIMINGS + USER TIMINGS
"EdidNotAvail"=dword:1 ; STD TIMINGS + USER TIMINGS
;-----
; Following are the registry entries
; for port's DVO I2C settings
;-----
[HKEY_LOCAL_MACHINE\Drivers\Display\Intel\ALL\1\Port\5\DVO]
;-----
; 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\5\FPInfo]
;-----
; Following are the registry entries
; for ports first custom DTD mode to add
;-----

; Following are the registry entries for the port device' display
attribute parameters
;
;-----

```



```
[HKEY_LOCAL_MACHINE\Drivers\Display\Intel\ALL\1\Port\5\Attr]
;-----
; Config 1 - HDMI-B Port
;-----
; Following are the registry
; entries for port's general config
;-----
[HKEY_LOCAL_MACHINE\Drivers\Display\Intel\ALL\1\Port\2\General]
  "Name"="HDMI-B"
  "Edid"=dword:1
  "EdidAvail"=dword:3 ; STD TIMINGS + EDID TIMINGS + USER TIMINGS
  "EdidNotAvail"=dword:1 ; STD TIMINGS + USER TIMINGS
;-----
; Following are the registry entries
; for port's DVO I2C settings
;-----
[HKEY_LOCAL_MACHINE\Drivers\Display\Intel\ALL\1\Port\2\DVO]
;-----
; 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]
;-----
; Following are the registry entries
; for ports first custom DTD mode to add
;-----

; Following are the registry entries for the port device' display
attribute parameters
;
;-----
[HKEY_LOCAL_MACHINE\Drivers\Display\Intel\ALL\1\Port\2\Attr]
;-----
; Config 1 - DP-C Port
;-----
; Following are the registry
; entries for port's general config
;-----
[HKEY_LOCAL_MACHINE\Drivers\Display\Intel\ALL\1\Port\7\General]
  "Name"="DP-C"
  "Edid"=dword:1
  "EdidAvail"=dword:3 ; STD TIMINGS + EDID TIMINGS + USER TIMINGS
  "EdidNotAvail"=dword:1 ; STD TIMINGS + USER TIMINGS
;-----
; Following are the registry entries
; for port's DVO I2C settings
;-----
```

## § §



## Appendix A Example INF File

---

```

;*****
; Filename: iegd.inf
; $Revision: 1.552.4.1.2.80 $
; $Id: iegd.inf,v 1.552.4.1.2.80 2012/04/14 04:17:18 lab_bldmstr 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 = 04/14/2012,1.12.0.2450

;=====
[SourceDisksNames]
1=%DiskDesc%,,,""

[SourceDisksFiles]
iegdmini.sys = 1
iegdckey.vp = 1
iegdmsys.vp = 1
iegdccagt.cpa = 1
iegdccagt.vp = 1
iegddis.dll = 1
analog.sys = 1
analog.vp = 1
lvds.sys = 1
lvds.vp = 1
hdmi.sys = 1
hdmi.vp = 1
dp.sys = 1
dp.vp = 1

;iegd3dga.dll = 1
;iegdglga.dll = 1
;libGLÉS_CM.dll = 1
;libGLÉSv2.dll = 1
;sdvo.sys = 1
;sdvo.vp = 1

;=====
[DestinationDirs]
DefaultDestDir = 11; System directory
iegd.Display_cdv = 11
iegd.Miniport = 12; Drivers directory
iegd.Copp = 12
iegd.Copp_cdv = 12
iegd.PortDrvs_cdv = 12

```



```
;=====
[Manufacturer]
%Intel%=Intel.Mfg

;=====
[Intel.Mfg]
%Intel% %iCDV0% = iegd_cdv, PCI\VEN_8086&DEV_0BE0
%Intel% %iCDV1% = iegd_cdv, PCI\VEN_8086&DEV_0BE1
%Intel% %iCDV2% = iegd_cdv, PCI\VEN_8086&DEV_0BE2
%Intel% %iCDV3% = iegd_cdv, PCI\VEN_8086&DEV_0BE3

;=====
[iegd_cdv.GeneralConfigData]
MaximumNumberOfDevices = 2
MaximumDeviceMemoryConfiguration = 256

;=====
[iegd_cdv]
CopyFiles = iegd.Miniport, iegd.Display_cdv, iegd.PortDrvs_cdv, iegd.Copp_cdv

;=====
[iegd.Miniport]
iegdmini.sys

[iegd.Copp]
iegdckey.vp
iegdmsys.vp
sdvo.vp
lvds.vp
iegdcagt.cpa
iegdcagt.vp

[iegd.Copp_cdv]
iegdckey.vp
iegdmsys.vp
analog.vp
lvds.vp
hdmi.vp
dp.vp
iegdcagt.cpa
iegdcagt.vp

[iegd.Display_cdv]
iegd_dis.dll

[iegd.PortDrvs_cdv]
analog.sys
lvds.sys
hdmi.sys
dp.sys

[iegd.Null]

;=====
[iegd_cdv.Services]
AddService = iegdmini, 0x00000002, iegd_Service_Inst, iegd_EventLog_Inst
AddService = analog, ,analog_Service_Inst, iegd_EventLog_Inst
AddService = lvds, ,lvds_Service_Inst, iegd_EventLog_Inst
AddService = hdmi, ,hdmi_Service_Inst, iegd_EventLog_Inst
AddService = dp, ,dp_Service_Inst, iegd_EventLog_Inst

;=====
```



```
[iegd_Service_Inst]
ServiceType      = 1
StartType        = %SERVICE_DEMAND_START%
ErrorControl     = 0
LoadOrderGroup  = Video
ServiceBinary    = %12%\iegdmini.sys

[analog_Service_Inst]
DisplayName      = "analog"
ServiceType      = %SERVICE_KERNEL_DRIVER%
StartType        = %SERVICE_DEMAND_START%
ErrorControl     = %SERVICE_ERROR_IGNORE%
ServiceBinary    = %12%\analog.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

[hDMI_Service_Inst]
DisplayName      = "hDMI"
ServiceType      = %SERVICE_KERNEL_DRIVER%
StartType        = %SERVICE_DEMAND_START%
ErrorControl     = %SERVICE_ERROR_IGNORE%
ServiceBinary    = %12%\hDMI.sys

[dp_Service_Inst]
DisplayName      = "dp"
ServiceType      = %SERVICE_KERNEL_DRIVER%
StartType        = %SERVICE_DEMAND_START%
ErrorControl     = %SERVICE_ERROR_IGNORE%
ServiceBinary    = %12%\dp.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_cdv.SoftwareSettings]
AddReg = iegd_SoftwareDeviceSettings_cdv

;-----
[iegd_SoftwareDeviceSettings_cdv]
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_N2000/D2000"
HKR, ALL\1\General , DisplayConfig, %REG_DWORD%, 1
HKR, ALL\1\General , DisplayDetect, %REG_DWORD%, 1
HKR, ALL\1\General , PortOrder, %REG_SZ%, "52678400"
HKR, ALL\1\General , DxvaOptions, %REG_DWORD%, 0
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\5\General , name, %REG_SZ%, "ANALOG"
HKR, ALL\1\Port\5\General , Rotation, %REG_DWORD%, 0
HKR, ALL\1\Port\5\General , Edid, %REG_DWORD%, 1
HKR, ALL\1\Port\5\General , EdidAvail, %REG_DWORD%, 3
HKR, ALL\1\Port\5\General , EdidNotAvail, %REG_DWORD%, 4
HKR, ALL\1\Port\5\General , CenterOff, %REG_DWORD%, 1
HKR, ALL\1\Port\2\General , name, %REG_SZ%, "HDMI-B"
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\1\Port\6\General , name, %REG_SZ%, "DP-B"
HKR, ALL\1\Port\6\General , Rotation, %REG_DWORD%, 0
HKR, ALL\1\Port\6\General , Edid, %REG_DWORD%, 1
HKR, ALL\1\Port\6\General , EdidAvail, %REG_DWORD%, 3
HKR, ALL\1\Port\6\General , EdidNotAvail, %REG_DWORD%, 4
HKR, ALL\1\Port\6\General , CenterOff, %REG_DWORD%, 1
HKR, ALL\1\Port\7\General , name, %REG_SZ%, "DP-C"
HKR, ALL\1\Port\7\General , Rotation, %REG_DWORD%, 0
HKR, ALL\1\Port\7\General , Edid, %REG_DWORD%, 1
HKR, ALL\1\Port\7\General , EdidAvail, %REG_DWORD%, 3
HKR, ALL\1\Port\7\General , EdidNotAvail, %REG_DWORD%, 4
HKR, ALL\1\Port\7\General , CenterOff, %REG_DWORD%, 1
HKR, ALL\1\Port\8\General , name, %REG_SZ%, "eDP"
HKR, ALL\1\Port\8\General , Rotation, %REG_DWORD%, 0
HKR, ALL\1\Port\8\General , CenterOff, %REG_DWORD%, 1
HKR, ALL\1\Port\8\General , Edid, %REG_DWORD%, 1
HKR, ALL\1\Port\8\General , EdidAvail, %REG_DWORD%, 3
HKR, ALL\1\Port\8\General , EdidNotAvail, %REG_DWORD%, 4
HKR, ALL\1\Port\8\FpInfo , bkltmethod, %REG_DWORD%, 1
HKR, ALL\1\Port\8\FpInfo , BkltT1, %REG_DWORD%, 200
HKR, ALL\1\Port\8\FpInfo , BkltT2, %REG_DWORD%, 1
HKR, ALL\1\Port\8\FpInfo , BkltT3, %REG_DWORD%, 200
HKR, ALL\1\Port\8\FpInfo , BkltT4, %REG_DWORD%, 50
```



```

HKR, ALL\1\Port\8\FpInfo , BkltT5, %REG_DWORD%, 500
HKR, ALL\1\Port\8\Dtd\1 , PixelClock, %REG_DWORD%, 69300
HKR, ALL\1\Port\8\Dtd\1 , HorzActive, %REG_DWORD%, 1366
HKR, ALL\1\Port\8\Dtd\1 , HorzSync, %REG_DWORD%, 48
HKR, ALL\1\Port\8\Dtd\1 , HorzSyncPulse, %REG_DWORD%, 32
HKR, ALL\1\Port\8\Dtd\1 , HorzBlank, %REG_DWORD%, 160
HKR, ALL\1\Port\8\Dtd\1 , VertActive, %REG_DWORD%, 768
HKR, ALL\1\Port\8\Dtd\1 , VertSync, %REG_DWORD%, 3
HKR, ALL\1\Port\8\Dtd\1 , VertSyncPulse, %REG_DWORD%, 5
HKR, ALL\1\Port\8\Dtd\1 , VertBlank, %REG_DWORD%, 22
HKR, ALL\1\Port\8\Dtd\1 , Flags, %REG_DWORD%, 0x20000

HKR,, No_D3D, %REG_DWORD%, 1

HKR,, PortDrivers, %REG_SZ%, "analog lvds hdmi dp"

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

;=====
[Strings]

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

iCDV0="Atom™ N2000/D2000 Series Embedded Media and Graphics Driver"
iCDV1="Atom™ N2000/D2000 Series Embedded Media and Graphics Driver"
iCDV2="Atom™ N2000/D2000 Series Embedded Media and Graphics Driver"
iCDV3="Atom™ N2000/D2000 Series Embedded Media and Graphics Driver"

;-----
; 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 116](#)
- [“HDMI Port Driver Attributes” on page 117](#)

*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 22, “Parameter Configuration Format” on page 51](#).

**Table 33. Standard Port Driver Attributes (Sheet 1 of 2)**

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.
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.



Table 33. Standard Port Driver Attributes (Sheet 2 of 2)

Attribute Name	Attribute ID Number	Description
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 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.
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 34. 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  <b>Note:</b> This setting changes the EMI characteristics, which can be measured with tuning equipment. The change will not necessarily be visible in the display.
DITHER	45	On and off Dithering	Dither=0 for 24-bit panels Dither=1 for 18-bit panels Default: <ul style="list-style-type: none"> <li>• dither = 1 for 18-bit panels</li> <li>• dither = 0 for 24-bit panels.</li> </ul>
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



Table 34. Internal LVDS Port Driver Attributes (Sheet 2 of 2)

Attribute Name	Attribute ID	Description	Possible Ranges
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.
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 Internal HDMI

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.2.2.3 HDCP

HDCP is supported through the Certified Output Protection Protocol\* (COPP) interface in Windows.



## B.3 Chipset and Port Driver-specific Installation Information

### B.3.1 Default Search Order

*Note:* See more information pertaining to port order in the description for [Section , “Port Devices \(Available Ports, Port Order\)”](#) on page 25.

**Table 35. Default Search Order**

Chipset	Default Search Order
Intel® Atom™ Processor N2000 and D2000 Series	CRT, LVDS

### B.3.2 Default GPIO Pin Pair Assignments

**Table 36. Default GPIO Pin Pair Assignments**

Chipset	Default GPIO Pin Pair for EDID	
	HDMI -B	LVDS
Intel® Atom™ Processor N2000 and D2000 Series	4	2

§ §



## 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 37. Summary of Intel 5F Extended Interface Functions (Sheet 1 of 2)**

Function	Function Name	Description	Page
<b>BIOS Extended Interface Functions</b>			
5F01h	Get Video BIOS Information	Gets VBIOS Build Information.	120
5F05h	Refresh Rate	Sets a new vertical refresh rate for a given mode and returns the current vertical refresh rate	120
5F10h	Get Display Memory Information	Returns information about the linear memory.	122
5F1Ch	BIOS Pipe Access	Sets the BIOS pipe access and returns the BIOS pipe access status.	122
5F29h	Get Mode Information	Returns information on the requested mode.	123
5F61h	Local Flat Panel Support Function	Supports local flat panel features.	123
5F68h	System BIOS Callback	Allows SoftBIOS to do any system callbacks through INT 15h	124
<b>Hooks for the System BIOS</b>			
5F31h	POST Completion Notification Hook	Signals the completion of video POST (Power On Self Test)	124
5F33h	Hook After Mode Set	Allows System BIOS to intercept Video BIOS at the end of a mode set.	124
5F35h	Boot Display Device Hook	Allows System BIOS to override video display default setting.	125
5F36h	Boot TV Format Hook	Allows System BIOS to boot TV in selected TV format state.	126



Table 37. 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.	126
5F40h	Config ID Hook	Allows System BIOS to supply a configuration ID that is passed to the driver.	127

## 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 Subfunction 5h of the 5f61h interface is 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.1.7 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 BIOS Firmware” on page 65](#). 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 BIOS Firmware” on page 65](#).

**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 section provides information on 2D and 3D support.

### D.1 2D Support

Intel® EMGD provides 2D capabilities on Windows through DirectX\*/GDI.

### D.2 3D Support

Intel® EMGD provides 3D capabilities through Direct3D.

§ §