OEM Diagnostic Tool for Graphics Controllers

External Product Specification (EPS)

Revision 7.5

April 16, 2008

Table of Contents

Table of Figures

Revision History

Rev	Draft/Changes	Date	Author
7.5	Addition of EagleLake new chipsets	April 16, 2008	Meenakshi Sundaram S
7.4	Updated EagleLake/Cantiga chipsets to their legal names	February 18, 2008	Meenakshi Sundaram S
7.3	Added EagleLake support	December 14, 2007	Meenakshi Sundaram S
7.2	Added Cantiga support	August 1, 2007	Meenakshi Sundaram S
7.1	Updated latest chipsets to their legal names	February 13, 2007	Meenakshi Sundaram S
7.0	Added Bearlake-B support	December 8, 2006	Craig Donovan
6.9	Added Crestline support	June 2, 2006	Craig Donovan
6.8	Updated latest chipsets to their legal names. Added section 4.1 detailing FreeDOS usage.	April 27, 2006	Craig Donovan
6.7	Added 3D Register test to replace de-featured 3D Render test for Broadwater / Crestline & future platforms.	April 7, 2006	Craig Donovan
6.6	Added new corporate logos.	January 31, 2006	Craig Donovan
6.5	Added Broadwater to supported product lists.	December 5, 2005	Craig Donovan
6.4	Updated latest chipsets to their legal names.	September 12, 2005	Craig Donovan
6.3	Fixed typo on page 11	February 17, 2005	Craig Donovan
6.2	Changed references to Lakeport and added Calistoga	January 25, 2005	Craig Donovan
6.1	Updated the “Installation Directory Structure” figure	December 2, 2004	Craig Donovan
6.0	Added Intel Lakeport to the supported product lists	November 10, 2004	Craig Donovan
5.9	Changed Intel® Copper River references to Intel® E7221 Chipset.	June 21, 2004	Craig Donovan
5.8	Updated Intel® Copper River Chipset references.	April 29, 2004	Craig Donovan
5.7	Added Intel® Copper River Chipset support.	April 22, 2004	Craig Donovan
5.6	Updated latest chipsets to their legal names.	April 6, 2004	Craig Donovan
5.5	Added Mobile Intel® 915GMS & 910GML text. Fixed 2004 dates in this table for Revs 5.4, 5.3, and 5.2. Updated log file text in appendixes.	April 5, 2004	Craig Donovan
5.4	Added information about Intel® 915G & Intel® 915GM code base unification..	March 26, 2004	Craig Donovan
5.3	Clarified “-o color xx” option wording.	February 17, 2004	Craig Donovan
5.2	Updated Grantsdale-G references to Intel® 915G.	February 10, 2004	Craig Donovan
5.1	Updated document header formats to properly support Adobe Acrobat PDF creation with bookmarks.	November 7, 2003	Craig Donovan
5.0	Added Grantsdale & Alviso to supported platforms. Updated command line limitation to include DOS/4GW	September 16, 2003	Craig Donovan

Rev	Draft/Changes	Date	Author
	issues.		
4.9	Removed extra return code from “-o oem pcd” section.	May 28, 2003	Craig Donovan
4.8	Updated “-o oem pcd” section.	May 23, 2003	Craig Donovan
4.7	Updated Video Modes table. Moved Bootable CD section.	March 25, 2003	Craig Donovan
4.6	Typo correction. Moved Appendix B.	March 17, 2003	Craig Donovan
4.5	Added Installation, and Common Issues & Errors sections	March 14, 2003	Craig Donovan
4.4	Updated internal name references to Intel® 865G. Changed executable and support files to i865diag.*	January 17, 2003	Craig Donovan
4.3	Updated internal name references to Intel® 852xM/855xM. Changed executable and support files to i855diag.*	December 10, 2002	Craig Donovan
4.2	Fixed color option syntax example line.	September 16, 2002	Craig Donovan
4.1	Added notes about text displays during graphics tests.	August 6, 2002	Craig Donovan
4.0	Updated Hardware cursor section with new diagram.	July 9, 2002	Craig Donovan
3.9	Changed internal name references to Intel® 845G/GL.	March 21, 2002	Craig Donovan
3.8	Added references to Intel® 865G/852xM/855xM chipsets.	February 22, 2002	Craig Donovan
3.7	Added video mode 0x116 to documentation.	December 11, 2001	Craig Donovan
3.6	Added Intel® 845G/GL support	November 16, 2001	Craig Donovan
3.5	Updated “auto” and “add” modes.	October 9, 2001	Craig Donovan
3.4	Added “auto” mode to “-o crc_file xxx” option. Added “-o oem xxx” switch to support PC-Doctor®. Updated Appendixes with Intel® 830M files.	October 3, 2001	Craig Donovan
3.3	Fixed 9 parameter switch limit of diag.bat. Added ‘loop’ parameter to diag.bat file. DIAG.BAT is now universal across all platforms (Intel® 830/815/810)	June 20, 2001	Craig Donovan
3.2	Added burst_mask option to control memory burst tests.	June 12, 2001	Craig Donovan
3.1	Added “y” key reference to Subjective on|off switch paragraph	March 29,2001	Craig Donovan
3.0	Changed internal name references to Intel® 830M.	February 9, 2001	Craig Donovan
2.9	Added Intel® 830M support.	October 26, 2000	Dennis Kush
2.8	Added color option description	August 24, 2000	Dennis Kush
2.7	Added delay option description	June 30, 2000	Dennis Kush
2.6	Updated the render_cursor test description	May 5, 2000	Dennis Kush
2.5	Updated the list of supported devices. Added 2 new options.	April 26, 2000	Dennis Kush
2.4	Added a usage note to the local_memory option. Added note about lagging ‘h’ specified for hex values. Removed line in render cursor test description that said no logging would be done unless subjective was selected.	October 21, 1999	Dennis Kush
2.3	Updated descriptions of graphics memory tests and render BLIT test. Added description of local_memory option.	October 20, 1999	Dennis Kush
2.1	Insert notes about batch file parameters and MS DOS command line lengths (sec 5.1).	August 26, 1999	Dennis Kush

Rev	Draft/Changes	Date	Author
	Clarify syntax and usage of hex and decimal values when specified in –o options that require their use (sec 6.3).		
2.0	Updated sample INI, Log, and Metrics files. Changed syntax of "-o mode " configuration option.	August 10, 1999	Dennis Kush
1.1	Render cursor description incorrect. Cursor colors 0 & 1 were swapped.	May 24, 1999	Dennis Kush
1.0	Updated render_3d and render_cursor test descriptions. Updated metrics file sample output. Updated crc_file setting description Added note in software requirement section about running in "True MS DOS Mode" and also running with EMM386.EXE.	May 2, 1999	Dennis Kush
0.9	Added References section Included OEM and PRD to Section 1.2 Added further metrics information and sample metrics log file Clarified the meaning of "xxx" in "IxxxDIAG.LOG" and "IxxxDIAG.INI"	Feb 11, 1999	Kevin Shekleton
0.1	Created	Feb 08, 1999	Kevin Shekleton

Introduction

1.1 Purpose/Scope

The purpose of this document is to define the external product specification for the OEM Diagnostic Tool. The intended audience is OEM manufacturers who want to test the integrated graphics controller portion of the supported chipset devices.

1.2 Definitions, Acronyms, and Abbreviations

This section defines all terms necessary to interpret this document.

EPS	External Product Specification. Software interface specification from an external perspective. Typically outlines modules and their calling interface.
GC	Graphics Controller.
DVM	Dynamic Video Memory.
OEM	Original Equipment Manufacturer
PRD	Products Requirement Document

Figure 1 - Definitions, Acronyms and Abbreviations

1.3 References

This section provides a complete list of all documents referenced elsewhere in the document.

2. Supported Devices

The OEM Diagnostic Tool currently supports the following devices:

- Intel® 810 Chipset family
  - Intel® 82810 Internal Graphics device
  - Intel® DC-100 Internal Graphics device
  - Intel® 82810E Internal Graphics device
- Intel® 815 Chipset family
  - Intel® 82815 Internal Graphics device
  - Intel® SolanoEM Internal Graphics device
- Intel® 830M Chipset family
  - Intel® 830M Internal Graphics device
- Intel® 845G/GL Chipset family

  - Intel® 845G/GL Internal Graphics device
- Intel® 852xM and Intel® 855xM Chipset families
  - Intel® 852GM, Intel® 852GME, Intel® 855PM, Intel® 855GM, and Intel® 855GME Internal Graphics devices
- Intel® 865G Chipset family
  - Intel® 865G Internal Graphics device
- Intel® 915G Express Chipset Family, Mobile Intel® 915GM Chipsets, and the Intel® E7221 Chipset
  - Intel® 82915G Express Chipset Family, Mobile Intel® 915GM/915GMS/910GML Extreme Chipsets and the Intel® E7221 Chipset
- Intel® 945G Express Chipset family, Mobile Intel® 945GM Express Chipsets
  - Mobile Intel® 945GM Express Chipsets, Intel® 945G Internal Graphics devices
- Intel® G965 Express Chipset family, Intel® Q965 Express Chipset family, and the Intel® Q963 Express Chipset family
  - Intel® G965, Intel® Q965, and Intel® Q963 Chipset devices
- Intel® PM965 Express Chipset family, Intel® GM965 Express Chipset family, and the Intel® GL960 Express Chipset family
- Intel® G33 Express Chipset family, Intel® G31 Express Chipset family, Intel® P35 Express Chipset family and the Intel® Q33 Express Chipset family
- Mobile Intel® 45 Express Chipset family
- Intel(R) 4 Series Internal Chipset, Intel(R) Q45/Q43 Express Chipset, Intel(R) G45/G43 Express Chipset and the Intel(R) G41 Express Chipset

3. Hardware Requirements

The OEM Diagnostic Tool requires that the device be a supported Intel Graphics Controller.

NOTE: At this time the OEM Diagnostics utility does not support legacy free PC's. The DOS4GW DOS extender requires an active 8042-keyboard controller to operate properly. The DOS extender will therefore fail if used on a system that uses a USB keyboard, floppy drive and mouse. At this time there is no work around for this issue.

4. Software Requirements

Once the hardware requirements are met, the user must have the OEM Diagnostic Tool and the Tenberry DOS4GW DOS extender. The OEM Diagnostic Tool comes with the DOS4GW DOS extender so there is no need to purchase a copy of DOS4GW. MS DOS 6.22 or later must be installed and all of the files included in the OEM Diagnostic Tool GC package must be in the same directory. A release of the OEM Diagnostic Tool GC will include all of the required software to test the supported hardware.

There is a unique OEM Diagnostic executable for each of the supported chipsets:

- I810DIAG.EXE for the Intel® 810 Chipset family.
- I815DIAG.EXE for the Intel® 815 Chipset family.
- I830DIAG.EXE for the Intel® 830M Chipset family.
- I845DIAG.EXE for the Intel® 845G/GL Chipset family.
- I855DIAG.EXE for the Intel® 852xM and Intel® 855xM Chipset families.
- I865DIAG.EXE for the Intel® 865G Chipset family.
- I915DIAG.EXE for the Intel® 82915G Express Chipset,
  Mobile Intel® 915GM/915GMS/910GML Extreme Chipset families and
  Intel® E7221 Chipset.
- I945DIAG.EXE for the Intel® 945G Express Chipset family, Mobile Intel® 945GM Express Chipsets family.
- I965DIAG.EXE for the Intel Crestline & Intel® G965 Chipset family, Intel® Q965 Chipset family, Intel® Q963 Chipset family
- I33DIAG.EXE for the Intel G33/G31/P35 Express Chipset family and Intel® Q33 Express Chipset family.
- I45DIAG.exe for the Intel ® 45 Express Chipset family, Intel ® 4 Series Internal Chipset, Intel(R) Q45/Q43 Express Chipset, Intel(R) G45/G43 Express Chipset and the Intel(R) G41 Express Chipset

This utility must be executed in “True MS DOS Mode” only. It will not execute in a “MS-DOS window”.

Because of a conflict between the DOS4GW DOS extender and EMM386.EXE, you must remove execution of EMM386.EXE from your system.

4.1 Free DOS alternative to MS-DOS 6.22

The OEM Diagnostics utility suite is only validated and supported on MS DOS v6.22 or greater. However, the development team has performed informal testing with a DOS alternative called FreeDOS. This version is available on the internet at www.freedos.org. If this version of DOS is used and issues are encountered, please verify the issue by running the utility on a supported version of Microsoft DOS before requesting support.

5. Installation

The OEM Diagnostics utility is distributed in a self-extracting ZIP file format. Once the distribution package has been downloaded, it MUST be executed under a Windows environment to obtain the actual binaries and documentation. The distribution package EXE will NOT run from DOS.

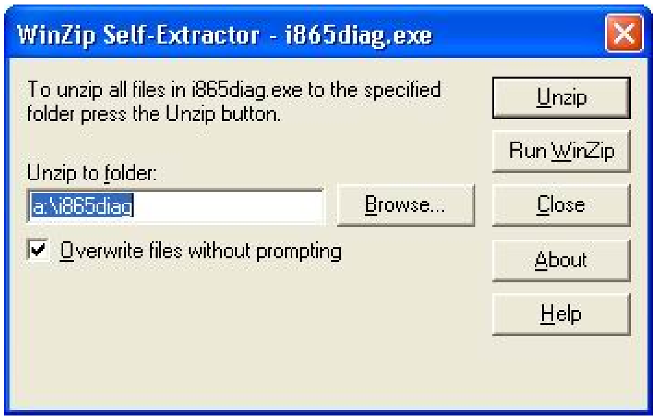
5.1 Installation Package Extraction

5.1.1 Overview:

- Download the distribution package to a machine running a Windows operating system.
- Execute the file named "iXXXdiag.exe" on the Windows machine to extract the contents of the self-extracting zip file.
- In the resultant dialog box, accept the default destination drive and directory or browse to another. **Figure 2**
- Press the "Unzip" button to extract the DOS binaries and documents now**. Figure 2**
  OR
  If you have WinZip installed, press the "WinZip" button to perform the extraction using WinZip instead.
- Copy the extracted files from the destination media to the test machine running MS DOS 6.22 or greater.
- The extracted files can now be executed in a true MS DOS environment.

5.1.2 Details

Using a Windows based machine (Windows 95, 98, NT, 2000 or XP), execute the installation package named **"iXXXdiag.exe".** The 'X' characters should be replaced with the chipset numbers or letters representing your specific chipset. In this case it is "865". **Figure 2** shows a typical WinZip Self-Extractor dialog box. Use the **[Browse]** button to choose a destination drive and directory for the resultant files, if the default is unacceptable. The example shown will extract the package contents to a subdirectory called "**i865diag**" on the A: drive when the "Unzip" button is pressed.


WinZip Self-Extractor - i865diag.exe

To unzip all files in i865diag.exe to the specified folder press the Unzip button.

Unzip to folder: a:\i865diag

☑ Overwrite files without prompting

Figure 2 - WinZip Self Extractor Dialog

Note: The distribution package CANNOT be extracted to the root directory of a drive by specifying ***"A:\"*** *in the* ***[Unzip to folder:]*** *text box.*
A subdirectory **<u>MUST</u>** be specified or an error will occur. This is an issue with the WinZip Self-Extractor application and not with the OEM Diagnostics utility. If you need the utility in the root directory, extract the contents to the default path and move the resultant files and directories manually, using Windows Explorer or use the full version of WinZip which is described later.

If WinZip is installed on your machine the **[Run WinZip]** button will execute the full version of WinZip and display the distribution package contents. From the WinZip application, the user can then extract the package contents. The full WinZip application can extract the package contents to the root directory of any drive.

Most of the documentation provided with this utility requires a graphical user interface such as Windows or Linux to view. HTML and PDF files may not be viewable in a true DOS environment and will require Windows or Linux to view them properly. Text documents however can be viewed directly in DOS using any appropriate text editor. All files generated by this utility when running under DOS, such as LOG and INI files, can be viewed by any DOS, Windows, or Linux text editor.

5.1.3 Installation Directory Structure

The distribution package contains a built-in directory structure which is recreated upon extraction to the destination media. DOS executables and all files required for execution are extracted to the **"\BIN"** directory. The DOC subdirectory is NOT required for the proper execution of the applications contained in "**\BIN**" directory. If additional space is needed on the destination media, the entire **"\DOC"** directory can be removed. This is especially useful when the destination media is a standard 1.44MB floppy disk

All documentation files are extracted to the **"\DOC**" subdirectory and are further categorized into subdirectories based on the application name. For example, the documentation for the OEM Diagnostic utility will be located in **"\DOC\OEMDIAG"** directory and the **"VESA"** utility documents will be located in the **"\DOC\VESA"** directory. This example assumes the distribution package was extracted to the root directory of the media. **Figure 3** shows the resultant directory structure of an extraction to the A: drive. Notice all the binaries are located in **"A:\IxxxDIAG\bin"**. Change to this directory to run the any of the utility applications. The **"DOS4GW.EXE"** file is not part of the utility suite and is actually a support file required by **"IxxxDIAG.EXE"**. It will not function without it.

The example directory structure will be similar across all versions of OEM Diagnostics. To obtain the actual distribution package contents, refer to "README.TXT" file located in the appropriate **"\DOC"** subdirectory.

Additional files may appear in the **"\BIN"** directory after any of the utilities have been run. Each utility will generate various output files for later review and editing. Generated files may have the following extensions: **LOG, BIN, CRC**. All generated files can be viewed with any DOS, Windows or Linux text editor.

```
A:.
└───IxxxDIAG
    ├───bin
    │     810CHIPL.BMP
    │     810CHIPS.BMP
    │   * DA.BAT
    │   * DETADD.EXE
    │   * DETADD.INI
    │     DIAG.BAT
    │     DOS4GW.EXE
    │     IxxxDIAG.EXE
    │     I865DIAG.INI
    │     VESA.EXE
    │     VESA.INI
    │     VGA.EXE
    │     VGA.INI
    │
    └───doc
        ├─*─detadd
        │     * OPTIONS.TXT
        │     * README.TXT
        │     * RELNOTES.HTM
        │     * VERSION.TXT
        │
        ├───oemdiag
        │       README.TXT
        │       RELNOTES.HTM
        │       USERGIDE.PDF
        │       VERSION.TXT
        │
        ├───vesa
        │       OPTIONS.TXT
        │       README.TXT
        │       RELNOTES.HTM
        │       VERSION.TXT
        │
        └───vga
                OPTIONS.TXT
                README.TXT
                RELNOTES.HTM
                VERSION.TXT

Executables are shown in BLUE
* 945G and later releases only
```

Figure 3 - Installation Directory Structure

5.1.4 Absolute Minimum File Requirements for Executables

This list is being provided for OEM's that require the maximum amount of disk space on the media where OEM Diagnostic utilities will reside. The list will be helpful in cases where a boot floppy has been created and other OS components or utilities need to be on the same media as OEM Diagnostics. By eliminating unnecessary files included with OEM Diagnostics additional disk space will be available.

It is HIGHLY RECOMMENDED that at least 50K be available on the execution media to support the writing of LOG files that the OEM Diagnostics utilities automatically generate.

Below is a list of the utilities included with the OEM Diagnostic Utility suite. Also listed are the absolute minimum files required to execute each utility. Although all INI files can be excluded from the execution directory this is NOT recommended. Without an INI file for each utility, all test parameters that differ from the program's defaults must be specified on the command line. Since the command line has a set character limit, specifying all the required parameters on the command line may not provide the end user with the functionality desired.

Utility File Name	Description	Required Support Files	Optional Support Files
IxxxDIAG.EXE [300 KB]	Main diagnostic utility	DOS4GW.EXE [270 KB] (**4**) 810CHIPS.BMP [3 KB] (**3**) 810CHIPL.BMP [16 KB] (**3**)	DIAG.BAT [4 KB] (**1**) IxxxDIAG.INI [2 KB] (**2**)
DETADD.EXE [60 KB](**5**)	DetADD utility	NONE	DETADD.INI [2 KB](**2**)
VESA.EXE [91 KB]	VESA utility	NONE	VESA.INI [2 KB](**2**)
VGA.EXE [92 KB]	VGA utility	NONE	VGA.INI [2 KB] (**2**)
			[Approximate File Size in KB] (See footnote info below)

Figure 4 - Minimum File Requirements for Executables table

Notes:

File sizes indicated are approximate and will vary slightly by platform and version.

INI file sizes represent the approximate size at the time of release. OEM modified INI files may increase or decrease INI file sizes.

(1) “loop” parameter and “PASS/FAIL” screen displays will not function without this file.

(2) Without the “INI” file, program defaults will be used. All test parameters must be submitted via command line. Missing “INI” file error message maybe displayed if missing, but basic execution not affected.

(3) 2D test (render_blt) may fail without this file. Video mode dependent.

(4) Program will not execute without this file.

(5) DetADD utility only supports Intel® G965/Q965/Q963/945G/945GM/915G/910G/E7221 Chipset products. Support will be provided for all future platforms.

6. Executing the OEM Diagnostic Tool

6.1 Execution with command line arguments

The OEM Diagnostic Tool accepts arguments given directly through the command line. The arguments are used to enable or disable specific tests as well as configure how the OEM Diagnostic Tool operates. See Section 6 for descriptions and correct syntax of the arguments.

Note: **Applies to all versions.** The DOS/4GW extender limits the available command line length. Passing parameters on the command line where there are between 103 and 106 characters, will cause truncation of the commands after the 103rd character. This scenario will most likely cause OEM Diagnostics to generate an error due to the truncation of the last parameter. If more than 106 characters are passed, the DOS/4GW executable will generate an exception fault or an error message and then terminate. There is no work around for this issue besides limiting the maximum command parameters to 102 characters. This limit equates to approximately 6 average length commands composed of 3 values each such as “-t render_blt on”. On Intel® 915G and later platforms, the DIAG.BAT file has been modified to generate a

DOS/4GW error message and terminate instead of generating an exception fault and terminating. This batch file will be included with earlier platforms releases when new releases are produced.

Note: **Applies to versions earlier than and equal to: 810 v2.7, 815 v1.3, and 830M v0.4.1.** If you use the provided DIAG.BAT file to execute the utility, ensure that no more than 9 parameters are specified on the command line or use the configuration file (see next section). This is a limitation of MS DOS batch files.

Note: **Applies to all versions equal to or greater than: 810 v2.8, 815 v1.4, 830M v0.4.2, and all later platforms.** The MS DOS batch file limitation of 9 parameters specified on the command line has been removed. The DIAG.BAT file was updated so that this limitation is no longer an issue except for the DOS/4GW limitation already mentioned. In addition the batch file supports a 'loop' parameter. This parameter causes the OEM Diagnostics to continually execute until there is a failure or until the user aborts testing using the CTRL-BREAK key sequence. The new batch file supports all platforms and all versions without editing. Just replace the current DIAG.BAT file on any supported system and execute it. The new batch file does not affect the functionality of the IxxxDIAG.INI file.

Syntax examples:

diag –t all on loop	// Runs all tests until failure or user abort.
diag loop –t all on	// Same as above. Placement of 'loop' parameter irrelevant.
diag loop	// Runs all tests specified in the INI file until failure or user abort.
diag –t all on LOOP	// **ILLEGAL –** 'loop' parameter must be all lower case.
i855diag –t all on loop	// **ILLEGAL –** 'loop' parameter is not supported in the executable.

This batch file will be included in all future releases of OEM Diagnostics.

NOTE: The MS DOS command line is typically limited to 128 characters in length. If you need to specify more than 128 characters on the command line, it is recommended that you use the configuration file. See next section for an explanation of this file.

6.2 Using a configuration file to pass arguments

The "IxxxDIAG.INI" configuration file (where "xxx" represents the Intel® Chipset) may be used in place of or in conjunction with command line arguments. The syntax for passing arguments from the configuration file to the Diagnostic Tool is the same as using command line arguments. However, using the configuration file simplifies the process of multiple tests. See Section 6 for descriptions and correct syntax of the arguments. It is important to note that switches from the "IxxxDIAG.INI" configuration file (if present) will be read sequentially followed by arguments from the command line. So, the last setting of a particular test is the one that is used.

6.3 Execution without arguments

If no command line or configuration file arguments are used, all tests will be performed and all logs will be recorded in the "IxxxDIAG.LOG" file (where "xxx" represents the Intel® Chipset). All setting defaults will be used.

6.4 Optimizing Execution Speed

By default OEM Diagnostics will run all available tests with an approximate run time of 5 minutes. Several factors can cause OEM Diagnostics to execute more slowly than expected. The execution media, video BIOS and the tests themselves will cause significant changes in the total execution time of OEM Diagnostics. The following sections discuss various methods of decreasing execution time.

6.4.1 Improving Application Performance

In some situations, an OEM may want to decrease testing time, without significantly affecting the validity of the testing process. Below is a list of methods to decrease testing time without a significant impact to the validity of the testing process.

Please refer to Sections 6, 7 and 10 for more information about particular tests and options.

- **Do not execute the OEM Diagnostics from a floppy disk.** OEM Diagnostics writes data to several log files after the completion of each subtest. Since writing to a floppy is very slow, changing the execution location to a hard drive or network share can cut the execution time in half. See Figure 5 for a list of relative execution times versus the execution media.

- **Reduce the number of tested video modes tested using the "-o mode nnn" option.** The 3 graphics tests, Hardware Cursor, 2D Blit and the 3D Render tests make up a significant part of the total test time. By default each test uses all applicable video modes, as reported by the video BIOS, to perform these tests. The OEM can decrease testing time by limiting which video modes are tested with the **"-o mode"** option. A typical example would be to include the **"-o mode 111,114,117"** command in the INI file to limit the graphics tests. In this example only the 3 listed modes will be tested. When using this option, be careful not to limit testing to an unavailable video mode. At least 1, available and appropriate video mode MUST be specified. If a test is executed and there are no available modes, due to the use of the "-o mode" option, an error will be generated. Also note that the graphics UMA size setting, available in CMOS, will affect the number video modes available to the system. A setting of 16Meg will typically have more video modes available than a UMA size of 1Meg. Appropriate video modes are shown in the table below:

OEM Diagnostics Tests	Tests all video modes with resolutions greater than or equal to:
Hardware Cursor (render_cursor)	640 x 400 pixels with color depths of **15 or 16 bits**.
2D Blit (render_blt)	640 x 400 pixels with a color depth of **16 bits**.
3D Render (render_3d)	640 x 400 pixels with a color depth of **16 bits**.

Figure 5 - Appropriate Video Modes for Graphics Tests

- **Reduce the number of available video modes with a CMOS setting.** As shown above, the number of video modes available will affect the number of graphics tests executed. Some BIOS's will allow the user to alter the amount of graphics memory allocated to graphics. This CMOS setting is typically called UMA Size, and has values such as 1M, 4M, 16M, 32M etc. Reducing the size of the UMA will, in most cases, reduces the number of available video modes and therefore decreases the number of graphics tests performed.

- **Limit the number of iterations that the Memory Burst uses with the "-o burst_mask nnn" option.** The memory burst test is the longest memory test executed. It executes 7 times with byte block sizes of 1, 2, 4, 8, 16, 32 and 64 bytes. A byte block size of 1 executes the slowest and a size of 64 executes the fastest. Removing the slower byte block sizes will decrease the memory burst test time. For example, using the **"-o burst_mask 96"** option, will only test using the byte block sizes 32 & 64 and skip the byte block sizes 1, 2, 4, 8 and 16.

- **Options to reduce the amount of memory tested during memory tests.** The two longest memory tests, memory stuck bit and memory burst tests are affected by several options. By changing the amount of memory tested, typical default is 12 Mbytes (0xBFFFFF), the testing time will be reduced.

  - **"memory_test_start" option:** Default is 0. Increase to a value less than 0xBFFFFF to decrease the amount of memory tested.

  - **"memory_test_end" option:** Default is 0xBFFFFF. Decrease to a value greater than 0 to decrease the amount of memory tested.

  - **"memory_test_percent" option:** Default is 100. Changing to 50 will only test half as much memory as the default.

- **Eliminate unnecessary tests.** Some OEM's use several tools to validate platforms besides OEM Diagnostics. Since all tools are not created equal, there may be some redundancy when it comes to testing memory or graphics. If for example another application is used to test memory, the OEM Diagnostic's memory tests could be disabled to save execution time. Simply edit the INI file or place the appropriate "-t *test_name* off" command on the command line to disable a test. To shut off more than a couple of tests, it is recommended that the INI file be edited instead of using the command line.
  The "INI" file lists all the available tests with each line preceded by a ';' semicolon. The semicolon is used to remark-out commands and options. To disable any test, simply locate the test name in the "INI" file and remove the semicolon remark character. See the examples below:

;-t render_blt off	(command has no effect – uses the ';')
-t render_3d off	(render_3d test turned off)

6.4.2 Installation Media versus Execution Speed

The speed in which the OEM Diagnostic tool executes is directly related to the speed of the media where it is installed. Since the tool writes to a log file after each sub-test, the overall execution speed is determined by the relative speed of the media were the log files are written. The table below shows the relative execution speed based on the media in which the utility is installed.

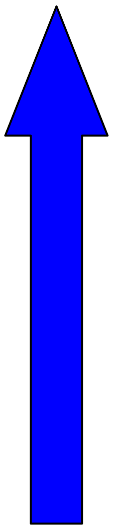
Speed	Media	Notes	Limitations
	MS DOS Ram Drive	Typically offers the fastest execution speed.	Log files written to ram drive should be copied to a non-volatile media before system shut down.
	Hard Drive	Fastest non-volatile media	
	Bootable CD	**Limited Support**	May have to provide an alternate, writable location for main log file generation. Use the "-f logfile" option. See next section for more info.
	CD	**Not Supported**	Execution is not supported because the application will attempt to write LOG and CRC files to the READ-ONLY media. See next section for more info.
	External Hard Drive		Support based on system configuration.
	Network Share		Network setup dependant.
	USB 2.0 Flash Drive (not write protected)	Maybe faster than hard drive medias.	Support based on system configuration. Have not formally tested this media yet.
	USB 1.1 Flash Drive (not write protected)	Fairly slow execution due to USB 1.1 bandwidth.	Support based on system configuration.
	Floppy Drive	Lowest speed, but offers low cost, low maintenance testing.	Limited media space available.

Figure 6 - Relative Execution Times vs. Media table

6.4.2.1 CDR’s and other READ-ONLY Medias

OEM Diagnostics must reside on a media that allows reading and writing. Although the main log file can be redirected to another writable location with the “-f” option, there are no options for redirecting the output to the METRICS.LOG and IxxxDIAG.CRC files. These files are by design, written to the execution directory.

Failures that will be encountered if attempting to run from READ-ONLY media include:
PROGRAM ABORT – Unable to open metrics file.
ABNORMAL TERMINATION
CRC Checksum error. 2D & 3D tests fail with log file indicating CRC correct value is 00000000.

6.4.2.2 Bootable CD’s

Bootable CD’s will suffer from the same problems that regular CDR’s do if created improperly. If the user simply takes a bootable CD project and adds the utility files to a directory on that CD, it will fail just as a regular CD would. This is because once the CD has booted the system; the media is still just another READ-ONLY media. OEM Diagnostics will fail as soon as it attempts to write its log files back to the device.

When a bootable CD starts a system, it performs a little bit of magic during the boot process. Typically the boot CD contains an image of a bootable floppy which is loaded into memory from the CD. This image boots just like a floppy drive, but off the CD itself. In fact, this boot image will appear to the system as the A: drive but in reality is a RAM Disk containing the boot image loaded from the CD. When the system finally completes its boot process, a quick investigation will reveal that the A: drive is in fact a RAM drive containing everything from the drive image on the CD. If the system had an actual floppy drive, which is usually the A: drive, it will be bumped up to become B: drive.

The trick to making OEM Diagnostics execute from a CD is to first create a complete bootable floppy containing the OEM Diagnostics utility and all the require boot files. You will then use this bootable floppy to create the bootable CD. This of course will require the use some third party software like Roxio’s Easy CD Creator, or Nero. Once the floppy is created, attempt to boot a system with it. Once booted, verify you that you can execute the OEM Diagnostic utility without any errors from the floppy disk. If all goes well, you can now use your CD Authoring software to create the bootable CD from this floppy. Below are the steps involved.

- Start by creating a DOS 6.22 or greater boot disk using the DOS command “**format a: /s**”
- Copy the OEM Diagnostic binaries from the “**\BIN**” directory to the floppy drive.
- Edit the floppy boot files, “AUTOEXEC.BAT” and “CONFIG.SYS” to support any required functionality. You must add support for CD Rom access if you want to access any files on the CD Rom not contained on the floppy. Once the CD Rom boots the system, the floppy image will become the A:\ drive and the CD Rom will be in accessible unless you have loaded a driver in the “CONFIG.SYS” file and “MSCDEX” application in the “AUTOEXEC.BAT” file.
- Test your boot floppy to verify that everything works as intended. Also if you need CD Rom access, verify this is operational too.
    - Special Note about the “**iXXXdiag.crc**” file. While testing the floppy a “**CRC**” file will be created. Unless you intend to use this file as the reference CRC file, for future testing, it should be deleted from the floppy before creating the boot CD. If you do not delete the file, the values contained in it will be used to validate all systems you test with the bootable CD.
    - When you are done testing the floppy delete all the “***.LOG**” files from the “\BIN” directory on the floppy.
- Use your CD Authoring software to start a Bootable CD project. At some point, you should be prompted to insert the boot floppy you created earlier.

- Add any additional files to the CD, but remember you will not be able to access them unless your boot floppy contains the correct drivers for CD access under DOS.
- Burn the CD and test it.

7. OEM Diagnostic Tool GC Arguments

7.1 General

Note: When specifying hex numbers as an input parameter to the arguments, use of a lagging 'h', (for example: 1234h) is invalid and will generate an invalid option failure message. Use a leading '0x' instead (0x1234).

`-i[ni] <ini_file_name> (Valid with I815DIAG.EXE version 1.1 or later)`

Specify name of INI file. Default is "IxxxDIAG.INI". Specifying this inside an INI file will have no effect.

`-f[ilename] <log_file_name>`

Specifies the name of the text file in which the diagnostic's output will be logged. If this switch is not used, the log file name defaults to "`IxxxDIAG.LOG`" (where "`xxx`" represents the Intel® Chipset).

Syntax Example: `-f OUTPUT.LOG`

`-?` or `-h[elp]`

This switch brings up the command line help. All of the commands are listed with brief descriptions.

Syntax Example: `-?`

`; <comment>`

A '`;`' at the beginning of a line in the `IxxxDIAG.INI` configuration file will cause the diagnostic to skip the remainder of that line. This allows you to include comments in the configuration file.

Syntax Example: `;Below test inserted on 2/1/99`

7.2 Tests

The `-t <test_name> on|off` selectively enables or disables the various OEM Diagnostics Tool tests. By default, all test are enabled. Below are the test switches that can be used with the `-t` switch. Tests are listed in the order that they are executed.

`all`

Using this will tell the Diagnostic Tool to use all of the tests on your Intel hardware. This is very useful, allowing a user to run all of the tests without having to specify each individual test through switches. By default, this option is turned on.

Syntax Example: `-t all on`

`pci_info`

This test scans the PCI bus and reports all of the devices that are present.

Syntax Example: `-t pci_info on`

`cpu_info`

This test reports the processor's family, model, stepping, as well as the amount of system memory.

Syntax Example: `-t cpu_info on`

`memory_info`

This test gives detailed information about the graphics device's graphics memory.

Syntax Example: `-t memory_info on`

`bios_info`

This option reports information about the video BIOS, including version number and supported video modes.

Syntax Example: `-t bios_info on`

`ddc_info`

This tests the graphics adapter and monitor's DDC (Display Data Channel) information. You must have a monitor that supports DDC in order to execute this test and receive a PASS indication.

Syntax Example: `-t ddc_info on`

`bios_crc`

This test calculates a checksum of the adapter video BIOS and compares it to the value stored in the shadowed BIOS space.

Syntax Example: `-t bios_crc on`

`memory_data_line`

This test checks the graphics memory data path by performing a walking 1s and 0s test on the last address in the memory range. This address will be either DVM or local memory, if installed.

The test performs the following (this description valid starting with version 2.3 of utility):

```
Set pattern equal to 0x00000001.
Loop until pattern equals zero.
    Write pattern to address, then read and compare.
    Write ones complement of pattern to address, then read and compare.
    Shift pattern left one place, filling places right of the one with zero.
```

Syntax Example: `-t memory_data_line on`

`memory_address`

This test performs a check on the graphics memory address bus. The range of addresses tested will be 4 Mbyte if local memory is installed or 12 Mbyte if local memory is not installed. Memory at address offsets 0, 1, 2, 4, … (end of memory) is first initialized to zero. Then a pattern of 0xFF is walked through the above listed offsets and tested.

The test performs the following (this description valid starting with version 2.3 of utility):

```
#define PATTERN 0xff
#define PASS 0
#define FAIL 1
unsigned long Add1, Add2, dwTestSize, dwStartAddr, fault = PASS;
unsigned char cv, *pc = (unsigned char *) dwStartAddr;
```

```
// Initialize tested locations
for( Add1 = 0; Add1 < dwTestSize; Add1 = (Add1) ? (Add1 << 1) : 1 )
    pc[Add1] = 0;

// Walk the address bus
for( Add1 = 0; (Add1 < dwTestSize) && !fault; Add1 = (Add1) ? (Add1 << 1) : 1 )
{
    pc[Add1] = PATTERN;    // Set address to test to pattern.
    for( Add2 = 0; (Add2 < dwTestSize) && !fault; Add2 = (Add2) ? (Add2 << 1) : 1 )
    {
        cv = (Add1 == Add2) ? PATTERN : 0;
        fault = (cv == pc[Add2] ) ? PASS : FAIL;
    }
    pc[Add1] = 0; // Set address to test back to zero.
}
```

`memory_stuck_bit`

This test checks the display adapter's graphics memory by writing a series of patterns to the memory, and making sure the values are properly written to memory. This test checks all installed local memory on the adapter and up to 12MB of DVM, when applicable.

The test performs the following for each address (this description valid starting with version 2.3 of utility):

write 0xAAAAAAA, then read and compare.
write 0x55555555, then read and compare.
write 0xFFFFFFFF, then read and compare.
write 0x00000000, then read and compare.

Syntax Example: `-t memory_stuck_bit on`

```
00000000>>
Performing memory stuck bit test...
<<00C00000
```

Figure 7 - Typical Memory Stuck Bit / Burst Test Display

`memory_burst`

This test checks the local memory by writing in blocks of 1, 2, 4, 8, 16, 32, and 64 bytes, then verifying that the block was written correctly. This test may take several minutes to complete. This test checks all installed memory on the adapter and up to 12MB of DVM, when applicable.

The test performs the following for each address:

For each block size, where block size = 1, 2, 4,. . .,64; throughout the range of memory
    Initialize block with 0x5A, read and compare
    Initialize block with 0xA5, read and compare

Syntax Example: `-t memory_burst on`

`registers`

This test checks the display adapter's registers by writing 1s and 0s to verify they are working correctly. It also takes into account any side effects one register has on another one.

Syntax Example: `-t registers on`

`command_exec`

This test attempts to execute a command from each of the two different possible sources:
1. high-priority ring(s) (HPR)
2. low-priority ring(s) (LPR).

Syntax Example: `-t command_exec off`

`render_cursor`

Note: Intel® 852 & Intel® 855 and later platform releases will display mode information in the upper left corner of the screen. Subjective mode prompts, are also displayed centered at the bottom of the screen.
This test exercises the Intel devices' hardware cursor overlaying functionality. For each supported video mode, a gradient background will be displayed, blue at the top to red at the bottom. The cursor will be rendered as a four-color rectangle in the middle of the screen. One rectangle is possible (three on I830DIAG.EXE). These rectangles are described below:

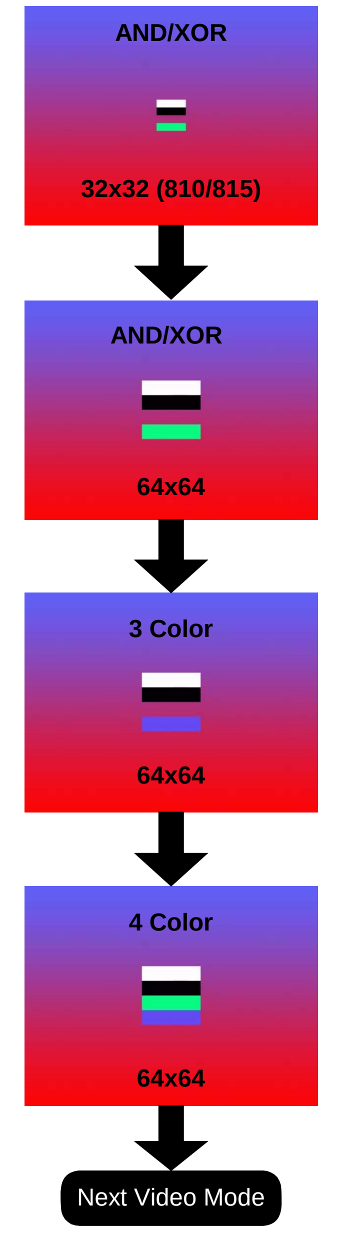


Figure 8 - Hardware Cursor Test Display

32 x 32 x 2bpp AND/XOR 2-plane mode (Valid on 810/815 only):

32 pixels wide by 32 scan lines in the center of the screen. The first 8 scan lines of the rectangle will be displayed as cursor color 0 (white). The second 8 scan lines will be displayed as cursor color 1 (black). The third set of 8 scan lines will be transparent (pixels of the main display image behind the cursor shows through. The fourth set of 8 scan lines will be transparent but inverted (pixels of the main display image behind the cursor shows through with inverted color.

64 x 64 x 2bpp AND/XOR 2-plane mode (valid with I830DIAG.EXE, I815DIAG.EXE version 1.1 or later):

64 pixels wide by 64 scan lines in the center of the screen. The first 16 scan lines of the rectangle will be displayed as cursor color 0 (white). The second 16 scan lines will be displayed as cursor color 1 (black). The third set of 16 scan lines will be transparent (pixels of the main display image behind the cursor shows through. The fourth set of 16 scan lines will be transparent but inverted (pixels of the main display image behind the cursor shows through with inverted color.

64 x 64 x 2bpp 3-color and transparency mode (valid with I830DIAG.EXE, I815DIAG.EXE version 1.1 or later):

64 pixels wide by 64 scan lines in the center of the screen. The first 16 scan lines of the rectangle will be displayed as cursor color 0 (white). The second 16 scan lines will be displayed as cursor color 1 (black). The third set of 16 scan lines will be transparent (pixels of the main display image behind the cursor shows through. The fourth set of 16 scan lines will be cursor color 3 (blue).

64 x 64 x 2bpp 4-color mode (valid with I830DIAG.EXE, I815DIAG.EXE version 1.1 or later):

64 pixels wide by 64 scan lines in the center of the screen. The first 16 scan lines of the rectangle will be displayed as cursor color 0 (white). The second 16 scan lines will be displayed as cursor color 1 (black). The third set of 16 scan lines will be cursor color 2 (green). The fourth set of 16 scan lines will be cursor color 3 (blue).

```
Syntax Example: -t render_cursor on
```

`render_blt`

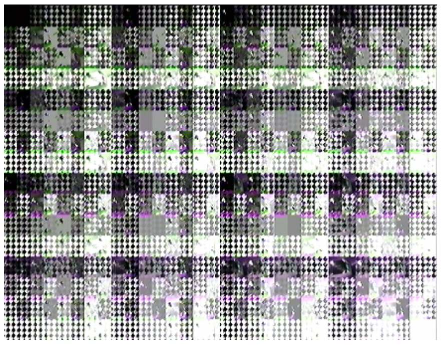


Figure 9 - Render Blit Test Display (2D)

Note: Intel® 852 & Intel® 855 and later platform releases will display mode information in the upper left corner of the screen. Subjective mode prompts, are also displayed centered at the bottom of the screen.
This test exercises the Intel devices' 2D Block Transfer (BLT) acceleration engine. For each video mode that is selected, the screen is divided into 256 rectangles, each one containing one of the Microsoft*-defined combinations of a striped background, a bitmap image, and monochrome pattern. A checksum of the resulting image is compared to a reference value to evaluate the accuracy of the rendering. The mode(s) in which the BLT functionality is tested can be specified by the "`-o mode`" option (see Section 7.3).

What the RENDER_BLT test does is run through all 256 possible raster operations (ROP) and displays them on the screen in a 16 x 16 series of boxes. Three images are operated on and combined using AND/OR/XOR/NOT combinations. Actual modes vary depending on platform and memory configuration.

Using mode 111h, 640 x 480, each box is 30 scan lines by 40 pixels.
Using mode 114h, 800 x 600, each box is 37 scan lines by 50 pixels.
Using mode 127h, 640 x 400, each box is 30 scan lines by 25 pixels.

The three images are: Source (which is the BMP chip image), Destination (which is 32 equally spaced, alternating light/dark gray vertical bars), and a Pattern (white circle against a black background, with 2 black pixels in the upper right).

Some of the easily discernable boxes are:
The upper left box (row 0, col. 0) is ROP #0, which is an all black image.
The lower right box (15, 15) is ROP #FFh, which is an all white image.
The lower left box (15, 0) is ROP #F0h, which is the white circle pattern on a black background. (mode 111h, you will see a 4 x 5 matrix of circles in the box, different resolutions produce different matrix sizes).
The upper right box (0, 15) is ROP #0Fh, which is the inverse of ROP #F0h.
ROP #AAh (10, 10) is the original destination image.
ROP #55h (5, 5) is the inverse of the destination image.
ROP #CCh (12, 12) is the original source image (clipping may occur at different resolutions).
ROP #33h (3, 3) is the inverse of the source image.

The following inside the red (or possibly gray) box is an example of the Pattern:

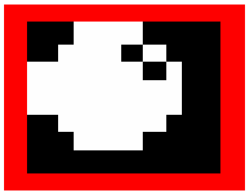


Figure 10 - Render Blit Pattern

Syntax Example: `-t render_blt on`

render_3d_reg (3D Test for Intel® G965/Q965/Q963 Chipset family and later platforms)

This option tests various 3D graphics registers to verify the operation of the 3D graphics engine. It does not provide a "visual" test and its output is similar to the standard register test described earlier. For actual graphics output validation, the 2D Blit test is provided and should be utilized instead of the de-featured 3D Render test.

Syntax Example: `-t render_3d_reg on`

render_3d (de-featured on Intel® G965/Q965/Q963 Chipset family and later platforms)

Special Intel® E7221 Platform Note: *Due to hardware constraints, the Render 3D test will not pass the software CRC check. It is recommended that either the Render 3D test NOT be run on this platform or to use the "crc_file" "create" or "add" options to effectively disable the software CRC check during the Render 3D test.*
Note: Intel® 852 & Intel® 855 and later platform releases will display mode information in the upper left corner of the screen. Subjective mode prompts, are also displayed centered at the bottom of the screen.
This option tests the Intel devices' 3D polygon rendering pipeline. Three multicolored, dithered, Gouraud-shaded triangles are rendered to the screen. The one in the upper-left corner of the screen demonstrates fogging, and should show a gradient coloring going from green at the bottom of the screen to black at the top. The triangle in the upper-right corner of the screen tests specular highlighting, and should have a gradient coloring from black at the bottom to white at the top. Finally, the center triangle demonstrates alpha transparency, and should be colored in red at the top, fading towards green at the bottom left and blue at the bottom right. These triangles are displayed once for each mode specified with the "`-o mode`" option (see Section 7.3), or in all supported 16-bit extended modes if none are specified by the user.

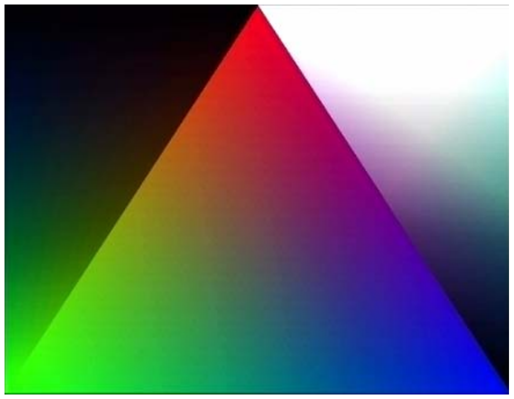


Figure 11 - Render 3D Test Display

Syntax Example: `-t render_3d on`

7.3 Settings

The `-o[ther] <setting_name> <value(s)>` selectively enables or disables the various OEM Diagnostic Tool settings. Below are the switch settings that can be used with the `-o` argument. Underlined options represent the default setting.

`burst_mask <decimal value>` (VALID WITH I830DIAG VERSION 0.4.3 OR LATER AND ALL LATER PLATFORMS)

This setting specifies which byte block sizes should be tested. The default for this option is 127, which specifies all byte block sizes. To determine the value, simple add up the block sizes to test. For example, if only block sizes 4 and 8 are to be tested, use 12 as the decimal value. Only values from 1 to 127 are valid. Any value outside this range will generate an error.

In the below example since 96 equals 32 + 64, only the 32 and 64 byte block sizes will execute.

Syntax Example: `-o burst_mask 96`

`color on|off` (I815DIAG VERSION 1.1 OR LATER, AND ALL LATER PLATFORMS)

This setting specifies whether the diagnostic should change the background color before exiting. If this option is set to on and all of the desired tests pass, the background will change to green. If this option is set to on, and any test fails, the background will change to red. If this option is set to off or the user aborts the testing, the background will be black. If an error has been detected on the command line or in the INI file, the exiting screen color will change only if this option has been parsed before the error occurs. The default for this option is "`off`".

Syntax Example: `-o color on`

`crc_file create|add|read|auto ("auto" VALID WITH I830DIAG VERSION 0.5.2+, and ALL LATER PLATFORMS)`

This setting allows the user to create or add to the file that contains the reference checksum values for rendering tests. Note that CRC values are only written for tests and video modes that are selected by the user during that execution of the diagnostic. The default for this option is "`auto`" on 830M platforms and later. The default for 810/815 platforms and earlier platforms is "read".

Special Intel® E7221 Platform Note: *Due to hardware constraints the Render 3D test will not pass the software CRC check. It is recommended that either the Render 3D test NOT be run on this platform or to use the "crc_file" "create" or "add" options to effectively disable the software CRC check during the Render 3D test.*

create: Use the "`create`" mode to completely wipe the current CRC file and start a new one. If the "`create`" mode is left on, the 2D and 3D tests will never fail due to a CRC error, because the CRC check file is recreated upon every execution. This mode of operation is not recommended. Use the "`create`" mode to generate a new CRC file when starting a new system, or when a system is reporting a CRC error. For the latter, change the mode to "`read`" or "`auto`" after the first run and execute diagnostics to verify the rendering tests now pass the CRC check.

add: The "`add`" mode simply adds new CRC values to the current CRC file. If the CRC file does not exist, one is created. This mode acts like a CRC value log and duplicate CRC values will be generated. In this mode all CRC tests will pass because values are just added to the log and assumed good. For any given 2D or 3D video mode only the last CRC value in the CRC file is used in verifying a video mode when in "`read`" or "`auto`" modes.

read: Read mode complements the "`create`" mode. Once a CRC file is generated with "`create`", the mode should be switched to "`read`" mode. Any change in the 2D or 3D display can then be detected with the CRC check during the rendering tests. Note "`read`" mode never modifies the CRC file. So if the "`create`" option is used with the 2D rendering test, only the CRC value for 2D rendering test will be present in the CRC file. Later if the 3D rendering test is run in "`read`" mode it will fail because no CRC values for this test will be present. This is the default for platforms prior to 830M.

auto: This is the new default for platforms that support it. This mode encompasses all the previous modes depending on the status of the CRC file. If a CRC file does not exists, one is created and new CRC values are "`added`". If a CRC file exists AND a CRC value for the given mode is present, this mode performs like the "`read`" mode. If a CRC file exists AND the CRC value for the given mode is NOT present, it is added. HINT: To force the generation of a new CRC file simple delete the current file and run diagnostics again in "`auto`" mode. Then run diagnostics a second time to compare the CRC values created during the first execution.

It is recommended that the CRC file be generated once on a known good system and then propagated to all other test systems. It is also recommended that since the utility relies heavily on the video BIOS, that the CRC file is re-generated after the video BIOS has been updated.

Mode	Can create a new CRC file	Can add new CRC's if missing	Will always pass CRC check	Can be used continuously without further user interaction	Notes	Description
create	Yes	No	Yes	See note 4	1,3,4	Recreates the CRC file from scratch.

add	Yes	Yes	Yes	See note 4	1,3,4	Effectively creates a CRC value log.
read	No	No	No	No		Never modifies or writes to the CRC file.
auto	Yes	Yes	No	Yes	2	Preferred default mode for most users.

Figure 12 - CRC File Option Matrix

Note 1:	Fails if CRC file is READ ONLY with a read-only file access error.
Note 2:	If a mode is missing from a READ ONLY CRC file, the test will fail with read-only file access error. When a mode is missing from the CRC file that is NOT READ ONLY, it is added on the first execution, and read on subsequent executions.
Note 3:	Always passes CRC check, because CRC values are only added to the CRC file and not verified.
Note 4:	This mode can be used continuously without further user interaction but doing so effectively disables any CRC error checking on the video display.

Syntax Example: `-o crc_file create`

`delay <decimal value>`

This switch sets the amount of time to display the screen during the hardware cursor, 2d, and 3d rendering test. The default is 3 seconds. The minimum value for the hardware cursor test is 0 seconds and for the 2d and 3d rendering tests is 3 seconds.

Syntax Example: `-o delay 3`

`halt_on_error on|off`

This setting specifies whether the diagnostic should halt immediately upon encountering an error, or attempt to complete all desired testing and report all encountered errors at the end of testing. If all of the desired tests pass, “`PASS`” will be displayed to the screen. If any test fails, “`FAIL`” is displayed on the screen. Furthermore, all test successes and failures will be noted in the log file (“`IxxxDIAG.LOG`” by default). The default for this option is “`on`”.

Syntax Example: `-o halt_on_error on`

`local_memory on|off`

This setting specifies whether the diagnostic should used display cache (local) memory as graphics memory.. If this option is set to off, then this forces system memory to be used in place of local memory.

The default for this option is “`on`” for 810 versions prior to v2.8, and “`off`” for v2.8 and later.
The default for this option is “`on`” for 815 versions prior to v1.3, and “`off`” for v1.3 and later.
The default for this option is “`on`” for 830M versions prior to v0.5.2, and “`off`” for v0.5.2 and later.
All future platforms the default is “`off`”

NOTE: Set this option to off if you have a chipset that supports the use of display cache, but do not have the memory installed. Otherwise, the memory info test will fail.

NOTE (830M v0.4.3 or greater): Local memory usage on 830M differs from previous 81x platforms. If local memory is installed and activated in BIOS then it will be mapped to the beginning of the display cache. On systems with 16Meg of local memory approximately 11 meg is available for display cache. Approximately 1 meg of system memory will be mapped to the end of the display cache, for a total display cache size of 12Meg. For systems with more than 16Meg of local memory, the entire display cache should be mapped to local memory. Specifying the **local_memory off** when it is present and active prevents the use of local memory for the display cache and does not generate an error. In this case 12Meg of system memory will be used for display cache.

If the BIOS is set to use UMA 512K, 1Meg, or 8Meg instead of local memory, the **local_memory off** option must be used to prevent test failure as described previously.

Syntax Example: -o local_memory on

`memory_test_start <hex address>`

This setting specifies the address (relative to the start of graphics memory) of the first byte to test when doing memory testing. Valid addresses range from 0 hex to BFFFFF hex. If no address is specified, or if the address is outside of the valid range, this setting will default to 0 hex. The value specified is always interpreted as a hex value. The leading 0x is optional. This setting is used for the stuck bit and burst tests only.

Syntax Example: `-o memory_test_start 0 or –o memory_test_start 0x0`

`memory_test_end <hex address>`

This setting specifies the address (relative to the start of video memory) of the last byte to test when doing memory testing. Valid addresses range from 0 hex to BFFFFF hex, depending on the amount of local memory installed. If no address is specified, or if the address is outside of the valid range, this setting will default to the address of the last byte of graphics memory. The value specified is always interpreted as a hex value. The leading 0x is optional. This setting is used for the stuck bit and burst tests only.

Syntax Example: `-o memory_test_end BFFFFF or –o memory_test_end 0xBFFFFF`

`memory_test_percent < decimal value>`

This setting specifies how much of the memory that the memory tests will check. The default value is 100. The value must be from 1 to 100. The value specified is always interpreted as a decimal value. This setting is used for the stuck bit and burst tests only.

Syntax Example: `-o memory_test_percent 50`

`memory_test_distrib <decimal value>`

This setting specifies how the memory test is distributed across the memory. memory_test_distrib will take the desired memory_test_percent and distribute it into segments of X% where X% is the value of memory_test_distrib. This is only useful if the memory_test_percent is set to less than 100. The default value of memory_test_distrib is 100. The value specified is always interpreted as a decimal value. This setting is used for the stuck bit and burst tests only.

Syntax Example: `-o memory_test_distrib 10`

`metrics on|off`

The metrics setting will record the time taken for each test and output the results to the metrics log file, "`metrics.log.` See Appendix B for information on how the metrics log file organizes it's results. The default for this option is "`on`".

Syntax Example: `-o metrics on`

`mode <hex mode number> [,<hex mode number>,... ]`

This setting allows the user to specify one or more modes in which to conduct the rendering tests. Note that the 2d and 3d tests currently support 16-bit extended modes only, the hardware cursor test may also support 15-bit extended modes in addition to 16-bit modes. Also, each mode requires a minimal amount of VGA memory and must be supported by the video BIOS. If no modes are specified, all modes supported by a test will be used. The value specified is always interpreted as a hex value. A leading 0x (as in 0x127) is allowed. The default for this option is to use modes automatically detected as supported by the video BIOS.

*****Valid modes are:

VESA Mode	Resolution		Color Depth	VGA Memory Requirements
0x101	640	480	8	0.29 MB
0x103	800	600	8	0.46 MB
0x105	1024	768	8	0.75 MB
0x107	1280	1024	8	1.25 MB
0x111	640	480	16	0.59 MB
0x112	640	480	32	1.17 MB
0x114	800	600	16	0.92 MB
0x115	800	600	32	1.83 MB
0x117	1024	768	16	1.50 MB
0x118	1024	768	32	3.00 MB
0x11A	1280	1024	16	2.50 MB
0x11B	1280	1024	32	5.00 MB
0x13A	1600	1200	8	1.83 MB
0x13C	1920	1440	8	2.64 MB
0x14B	1600	1200	16	3.66 MB
0x14D	1920	1440	16	5.27 MB
0x15A	1600	1200	32	7.32 MB
0x15C	1920	1440	32	10.55 MB

Figure 13 - Video Mode Table

* Available modes may vary by vender and video BIOS versions.

NOTE: The mode values must be surrounded by double quotes when this option is specified on the command line.

INI file syntax example: `-o mode 117,111 or -o mode 0x117,0x111`

Command line syntax example: -o mode “111,117” or -o mode “0x111,0x117”

`oem std|pcd (VALID WITH I830DIAG VERSION 0.5.1 OR GREATER, AND ALL LATER PLATFORMS)`

The oem setting has been added to support various special OEM functional requests. It is only supported on the 830M and later platforms. Two functions are supported, STD and PCD (Standard, PC-Doctor®).

STD: Standard mode is the default and disables all other “oem” functions. Standard mode is activated automatically and is the program’s default setting. Command syntax checking is on and active.

PCD: The PCD function supports the PC-Doctor® application by changing the return codes that OEM Diagnostics reports back to the operating system. When in PCD mode, a screen message is displayed upon exiting the application in the form “PCD ssss (n)” where ‘ssss’ is either Pass, Failed, or User Break. The ‘n’ value represents the PC-Doctor® return code. There are no log entries to indicating this mode. The PC-Doctor®

verses the Standard return codes are shown below. Only the return codes shown in bold will be returned for any particular mode and are fully supported. It is recommended that when using PCD mode, the OEM Diagnostic (i8xxdiag.exe) utility be called directly without using the DIAG.BAT file, since the BAT file does not support the PCD return codes.

It is particularly important to verify the command syntax within the INI file and on the command line when using this option. When running OEM Diagnostics under the PC Doctor® application, all command syntax checking is disabled following the use of the "-o oem pcd" option. This is required because the PC Doctor® application may attempt to pass incompatible parameters to OEM Diagnostic application, causing it to fail. Any unknown commands contained in the INI file following the "-o oem pcd" option switch will be ignored. Furthermore, if an error exists in the INI file, legal commands following it may not execute properly or at all.

For proper execution it is recommended that the "–o oem pcd" option be placed at the top of the OEM Diagnostic INI file, and NOT as a command line switch passed from within the PC Doctor® application itself.

OEMDiags Result	Standard (STD)	PC-Doctor® (PCD)	PC-Doctor® defines
PASS	**0**	**1**	**PCDR_RESULT_PASSED**
FAIL	**2**	**2**	**PCDR_RESULT_FAILED**
N/A	N/A	3	PCDR_RESULT_NA
N/A	N/A	4	PCDR_RESULT_ABORTED
N/A	N/A	5	PCDR_RESULT_UNDEFINED
USER ABORT	**1**	**6**	**PCDR_RESULT_USERBREAK**

Figure 14 - Return Code Table

Syntax Example: `-o oem pcd`

`subjective on|off`

Note: Intel® 852 & Intel® 855 and later platform releases displays a user prompt when the "subjective on" option is activated. The prompt will be centered at the bottom of all graphics tests (2D, 3D, Hardware Cursor). Platforms earlier than Intel® 852 & Intel® 855 will NOT prompt the user!
If this switch is set to "`on`", the diagnostic will pause after every graphical rendering to wait for the user's subjective judgment of the rendering. Pressing the spacebar or the "y" key passes the test, and pressing the "`n`" key fails it. If this switch is set to "`off`", the diagnostic will display the rendering for a minimum of 3 seconds, then move on to the next test. CRC tests of the rendered images will be performed regardless of this switch setting. The default for this option is "`off`".

Syntax Example: `-o subjective off`

8. Appendix A: Common Issues & Solutions

- OEM Diagnostics fails with a CRC error when I execute the 2D and/or 3D rendering tests.

  This issue is usually caused by the reuse of a CRC file generated by a previous version of the OEM Diagnostic utility. Some changes to the Video BIOS, System BIOS and code changes between releases, will change the CRC values contained in this file. It is recommended that with each new version of the code that a new CRC file be generated. Updating the system or video BIOS may also require a new CRC file. To do this, simply delete the **"iXXXdiag.crc**", located in the **"\BIN"** directory, and rerun the render_blt and render_3d tests. Then run both test again to check. The first execution will WRITE a new set of values to the CRC File. The second execution will then READ values from the CRC file and compare them against the currently calculated values from this execution run. This testing sequence assumes that the version of OEM Diagnostics you are using supports AUTOMATIC CRC file creation. To verify your version supports this mode, please refer to **"crc_file"** option in section 7.3. To ensure that all possible video modes are tested and have CRC values generated, update your CMOS settings so that the maximum UMA memory is allocated for graphics. Make sure that all available video modes are enabled, by insuring that the **"-o mode xxx,yyy,nnn"** is NOT used on the command line or in the **"iXXXdiag.ini"** file. The command can be remarked out by placing a ';' at the front of the line containing the mode option in the INI file.
  Special Intel® E7221 Platform Note: *Due to hardware constraints the Render 3D test will not pass the software CRC check. It is recommended that either the Render 3D test NOT be run on this platform or to use the "crc_file" "create" or "add" options to effectively disable the software CRC check during the Render 3D test.*

  Related Topics:
  Section 7.2 Tests - Full description of available tests including the "**render_blt**" and "**render_3d**" tests.
  Section 7.3 Settings - Full description of program options such as "**mode**" and "**crc_file**".

- The OEM Diagnostic executable will not execute in DOS.

  This problem can be caused by several things. If you try to execute the Installation Package you obtained from the Intel ARMS web site in DOS, you are correct, it will not run. This file is a self extracting zip file which must be run under a Windows OS in order to extract the DOS components and documents. Another common problem is missing components in the utility directory. OEM Diagnostics requires the use of a DOS extender which may not be present in the execution directory. A file called DOS4GW.EXE must be present. If it is not, re-extract the files from the Installation Package file you downloaded from ARMS. If the file is still not present, contact Intel for an update or instructions.

  OEM Diagnostics has been designed and validated using Microsoft DOS 6.22 or greater. This includes the versions of DOS used with Windows 95/98. It is not guaranteed to run on anything but Microsoft DOS. Please verify you are using the correct version of DOS and that it is a Microsoft version.

  Related Topics:
  Section 4 Software Requirements – Detailed list of various software requirements.
  Section 5 Installation – Details the process of retrieving the application files from the distribution package.

9. Appendix B: Using OEM Diagnostic Tool Switches

Below are various examples of the correct execution of the OEM Diagnostic Tool using switches.

9.1 Command Line Switches

DIAG.BAT –t all off –t registers on –o halt_on_error on

The above execution would launch the Diagnostic Tool running only the registers test and would halt immediately if an error was encountered.

DIAG.BAT –t all on –t pci_info off –o halt_on_error on –o memory_test_end FF

The above execution will launch Diagnostic Tool and run all of the test excluding the PCI bus info test. The program would also halt immediately upon encountering an error and end memory testing at address offset 0xFFh.

9.2 Configuration File Switches

Below are correct ways to pass arguments to the Diagnostic Tool using the `IxxxDIAG.INI` configuration file.

```
;----- Start ixxxDIAG.INI ----
-f output.log
-t all on
; We don’t want to test memory burst just yet…
–t memory_burst off
–o halt_on_error off
;----- End ixxxDIAG.INI -----
```

The above configuration file would open a file “`output.log`” for writing all output to. OEM Diagnostic Tool would then proceed with all but the memory burst test. It would not stop on errors (if any) encountered.

9.3 Combined Command Line and Configuration File Switches

```
;----- Start ixxxDIAG.INI -----
-t all on
–o halt_on_error off
;----- End ixxxDIAG.INI -----

DIAG.BAT –t cpu_info off –o halt_on_error on
```

The above execution would run all test excluding the CPU info test and halt on any errors that might occur. Remember that the Diagnostic Tool first looks at the configuration file and then the command line switches before running. Although the configuration file had `halt_on_error` in the ‘`off`’ setting, the command line switch, being the last setting, turned it ‘`on`’.

10. Appendix C: Sample Configuration File

The following is a sample of the I915DIAG.INI configuration file:

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
;
;   $Workfile: I915DIAG.INI $
;
;   Purpose:
;       Intel(r) Graphics Controller OEM diagnostics option file.
;
;   Environment:
;       MSDOS 6.22, DOS protected mode.
;
;   Warnings and Restrictions:
;       Copyright (c) 2004 Intel Corporation, all rights reserved.
;
;       INTEL MAKES NO WARRANTY OF ANY KIND REGARDING THE CODE.  THIS CODE
;       IS LICENSED ON AN "AS IS" BASIS AND INTEL WILL NOT PROVIDE ANY
;       SUPPORT, ASSISTANCE, INSTALLATION, TRAINING OR OTHER SERVICES.
;       INTEL DOES NOT PROVIDE ANY UPDATES ENHANCEMENTS OR EXTENSIONS.
;       INTEL SPECIFICALLY DISCLAIMS ANY WARRANTY OF MERCHANTABILITY,
;       NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY OTHER
;       WARRANTY.  INTEL DISCLAIMS ALL LIABILITY, INCLUDING LIABILITY FOR
;       INFRINGMENT OF ANY RIGHTS, RELATING TO USE OF THE CODE.  NO
;       LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY
;       INTELLECTUAL PROPERTY RIGHTS IS GRANTED HEREIN.
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

-t all on
;-t pci_info off
;-t cpu_info off
;-t memory_info off
;-t bios_info off
;-t ddc_info off
;-t bios_crc off
;-t memory_data_line off
;-t memory_address off
;-t memory_stuck_bit off
;-t memory_burst off
;-t registers off
;-t command_exec off
;-t render_cursor off
;-t render_blt off
;-t render_3d off
;-o burst_mask 127
;-o crc_file create
;-o halt_on_error off
;-o local_memory on
;-o memory_test_start 0
;-o memory_test_end bfffff
;-o memory_test_percent 100
;-o memory_test_distrib 1
;-o oem std
;-o metrics off
;-o subjective on
-o mode 111,114,117
;-f i915diag.log
```

11. Appendix D: Sample Log File

The below sample log file was generated using an Intel® 915GM Chipset and the corresponding OEM Diagnostic tool using the configuration file specified in Appendix A. The actual output may vary depending on your individual hardware and software configuration:

```
##############################################################################
##          Mobile Intel(r) 915GM/915GMS/910GML Extreme Chipsets           ##
##                 Graphics Controller Diagnostic Utility                   ##
##                                Build 33                                   ##
##                   Copyright(c) 2004 Intel Corporation                     ##
##############################################################################


##############################################################################
PCI Device Information (pci_info) ############################################

Setting                 Value       Description
------------------------------------------------------------------------------
Class:                  0x06        Bridge device
Sub-Class:              0x00        Host/PCI bridge
Interface:              0x00        Host bridge
Device ID:              0x2590      Intel(r) 915GM/PM/GMS & Intel(r) 910GML
Processor to DRAM       Ctrlr
Vendor ID:              0x8086      Intel Corporation
Subsystem ID:           0x1999      Unknown
Subsystem Vendor ID:    0x8086      Intel Corporation
Revision ID:            0x01
Interrupt Line:         0x00        IRQ0
Base Addr. Registers:   0x00000000  Not Implemented
                        0x00000000  Not Implemented
                        0x00000000  Not Implemented
                        0x00000000  Not Implemented
                        0x00000000  Not Implemented
                        0x00000000  Not Implemented
PCI Express Device:     No          0 PCI Express devices found
PCI Cap List ID's:      0x09        Vender Specific


Setting                 Value       Description
------------------------------------------------------------------------------
Class:                  0x03        Display controller
Sub-Class:              0x00        VGA controller
Interface:              0x00        VGA-compatible controller
Device ID:              0x2592      Intel(r) Alviso Graphics Controller 0
Vendor ID:              0x8086      Intel Corporation
Subsystem ID:           0x1999      Unknown
Subsystem Vendor ID:    0x8086      Intel Corporation
Revision ID:            0x01
Interrupt Line:         0x07        IRQ7
Base Addr. Registers:   0xB0080000  Memory-Mapped Register
                        0x00001801  I/O Register
                        0xC0000008  Memory-Mapped Register
                        0xB0040000  Memory-Mapped Register
                        0x00000000  Not Implemented
                        0x00000000  Not Implemented
PCI Express Device:     No          0 PCI Express devices found
PCI Cap List ID's:      0x01        PCI Power Management Interface


Setting                 Value       Description
-----------------------------------------------------------------------------
Class:                  0x03        Display controller
Sub-Class:              0x80        Other
Interface:              0x00        Other display controller
Device ID:              0x2792      Intel(r) Alviso Graphics Controller 1
Vendor ID:              0x8086      Intel Corporation
Subsystem ID:           0x1999      Unknown
Subsystem Vendor ID:    0x8086      Intel Corporation
Revision ID:            0x01
Interrupt Line:         0x00        IRQ0
Base Addr. Registers:   0x00000000  Not Implemented
                        0x00000000  Not Implemented
                        0x00000000  Not Implemented
                        0x00000000  Not Implemented
                        0x00000000  Not Implemented
                        0x00000000  Not Implemented
PCI Express Device:     No          0 PCI Express devices found
PCI Cap List ID's:      0x01        PCI Power Management Interface


Setting                 Value       Description
-----------------------------------------------------------------------------
Class:                  0x04        Multimedia device
Sub-Class:              0x03        Unknown
Device ID:              0x2668      Unknown
Vendor ID:              0x8086      Intel Corporation
Subsystem ID:           0x1999      Unknown
Subsystem Vendor ID:    0x8086      Intel Corporation
Revision ID:            0x02
Interrupt Line:         0x07        IRQ7
Base Addr. Registers:   0xB0000004  Memory-Mapped Register
                        0x00000000  Not Implemented
                        0x00000000  Not Implemented
                        0x00000000  Not Implemented
                        0x00000000  Not Implemented
                        0x00000000  Not Implemented
PCI Express Device:     Yes         1 PCI Express device found
PCI Cap List ID's:      0x01        PCI Power Management Interface
                        0x05        Message Signaled Interrupts
                        0x10        PCI Express


Setting                 Value       Description
-----------------------------------------------------------------------------
Class:                  0x06        Bridge device
Sub-Class:              0x04        PCI/PCI bridge
Interface:              0x00        PCI-to-PCI bridge
Device ID:              0x2660      Intel(r) 82801FB/FBM PCI Express Root Port
Vendor ID:              0x8086      Intel Corporation
```

```
Subsystem ID:           0x0000      Unknown
Subsystem Vendor ID:    0x0000      Unknown
Revision ID:            0x02
Interrupt Line:         0x0A        IRQ10
Base Addr. Registers:   0x00000000  Not Implemented
                        0x00000000  Not Implemented
                        0x00020200  Memory-Mapped Register
                        0x200000F0  Memory-Mapped Register
                        0x0000FFF0  Memory-Mapped Register
                        0x0001FFF1  I/O Register
PCI Express Device:     Yes         1 PCI Express device found
PCI Cap List ID's:      0x10        PCI Express
                        0x05        Message Signaled Interrupts
                        0x0D        Not Defined
                        0x01        PCI Power Management Interface


Setting                 Value       Description
-----------------------------------------------------------------------------------
Class:                  0x06        Bridge device
Sub-Class:              0x04        PCI/PCI bridge
Interface:              0x00        PCI-to-PCI bridge
Device ID:              0x2662      Intel(r) 82801FB/FBM PCI Express Root Port
Vendor ID:              0x8086      Intel Corporation
Subsystem ID:           0x0000      Unknown
Subsystem Vendor ID:    0x0000      Unknown
Revision ID:            0x02
Interrupt Line:         0x07        IRQ7
Base Addr. Registers:   0x00000000  Not Implemented
                        0x00000000  Not Implemented
                        0x00030300  Memory-Mapped Register
                        0x200000F0  Memory-Mapped Register
                        0x0000FFF0  Memory-Mapped Register
                        0x0001FFF1  I/O Register
PCI Express Device:     Yes         1 PCI Express device found
PCI Cap List ID's:      0x10        PCI Express
                        0x05        Message Signaled Interrupts
                        0x0D        Not Defined
                        0x01        PCI Power Management Interface


Setting                 Value       Description
-----------------------------------------------------------------------------------
Class:                  0x06        Bridge device
Sub-Class:              0x04        PCI/PCI bridge
Interface:              0x00        PCI-to-PCI bridge
Device ID:              0x2664      Intel(r) 82801FB/FBM PCI Express Root Port
Vendor ID:              0x8086      Intel Corporation
Subsystem ID:           0x0000      Unknown
Subsystem Vendor ID:    0x0000      Unknown
Revision ID:            0x02
Interrupt Line:         0x0B        IRQ11
Base Addr. Registers:   0x00000000  Not Implemented
                        0x00000000  Not Implemented
                        0x00040400  Memory-Mapped Register
                        0x200000F0  Memory-Mapped Register
                        0x0000FFF0  Memory-Mapped Register
                        0x0001FFF1  I/O Register
PCI Express Device:     Yes         1 PCI Express device found
PCI Cap List ID's:      0x10        PCI Express
                        0x05        Message Signaled Interrupts
                        0x0D        Not Defined
                        0x01        PCI Power Management Interface


Setting                 Value       Description
-----------------------------------------------------------------------------
Class:                  0x06        Bridge device
Sub-Class:              0x04        PCI/PCI bridge
Interface:              0x00        PCI-to-PCI bridge
Device ID:              0x2666      Intel(r) 82801FB/FBM PCI Express Root Port
Vendor ID:              0x8086      Intel Corporation
Subsystem ID:           0x0000      Unknown
Subsystem Vendor ID:    0x0000      Unknown
Revision ID:            0x02
Interrupt Line:         0x0A        IRQ10
Base Addr. Registers:   0x00000000  Not Implemented
                        0x00000000  Not Implemented
                        0x00050500  Memory-Mapped Register
                        0x200000F0  Memory-Mapped Register
                        0x0000FFF0  Memory-Mapped Register
                        0x0001FFF1  I/O Register
PCI Express Device:     Yes         1 PCI Express device found
PCI Cap List ID's:      0x10        PCI Express
                        0x05        Message Signaled Interrupts
                        0x0D        Not Defined
                        0x01        PCI Power Management Interface


Setting                 Value       Description
-----------------------------------------------------------------------------
Class:                  0x0C        Serial bus controller
Sub-Class:              0x03        USB (Universal Serial Bus)
Interface:              0x00        USB (Universal host controller spec)
Device ID:              0x2658      Intel(r) 82801FB/FBM USB Universal Host
Controller
Vendor ID:              0x8086      Intel Corporation
Subsystem ID:           0x1999      Unknown
Subsystem Vendor ID:    0x8086      Intel Corporation
Revision ID:            0x02
Interrupt Line:         0x05        IRQ5
Base Addr. Registers:   0x00000000  Not Implemented
                        0x00000000  Not Implemented
                        0x00000000  Not Implemented
                        0x00000000  Not Implemented
                        0x00001821  I/O Register
                        0x00000000  Not Implemented
PCI Express Device:     No          0 PCI Express devices found
PCI Cap List ID's:      0x00        Not Implemented
```

```
Setting                 Value       Description
-----------------------------------------------------------------------------
Class:                  0x0C        Serial bus controller
Sub-Class:              0x03        USB (Universal Serial Bus)
Interface:              0x00        USB (Universal host controller spec)
Device ID:              0x2659      Intel(r) 82801FB/FBM USB Universal Host
Controller
Vendor ID:              0x8086      Intel Corporation
Subsystem ID:           0x1999      Unknown
Subsystem Vendor ID:    0x8086      Intel Corporation
Revision ID:            0x02
Interrupt Line:         0x0A        IRQ10
Base Addr. Registers:   0x00000000  Not Implemented
                        0x00000000  Not Implemented
                        0x00000000  Not Implemented
                        0x00000000  Not Implemented
                        0x00001841  I/O Register
                        0x00000000  Not Implemented
PCI Express Device:     No          0 PCI Express devices found
PCI Cap List ID's:      0x00        Not Implemented


Setting                 Value       Description
-----------------------------------------------------------------------------
Class:                  0x0C        Serial bus controller
Sub-Class:              0x03        USB (Universal Serial Bus)
Interface:              0x00        USB (Universal host controller spec)
Device ID:              0x265A      Intel(r) 82801FB/FBM USB Universal Host
Controller
Vendor ID:              0x8086      Intel Corporation
Subsystem ID:           0x1999      Unknown
Subsystem Vendor ID:    0x8086      Intel Corporation
Revision ID:            0x02
Interrupt Line:         0x0B        IRQ11
Base Addr. Registers:   0x00000000  Not Implemented
                        0x00000000  Not Implemented
                        0x00000000  Not Implemented
                        0x00000000  Not Implemented
                        0x00001861  I/O Register
                        0x00000000  Not Implemented
PCI Express Device:     No          0 PCI Express devices found
PCI Cap List ID's:      0x00        Not Implemented


Setting                 Value       Description
-----------------------------------------------------------------------------
Class:                  0x0C        Serial bus controller
Sub-Class:              0x03        USB (Universal Serial Bus)
Interface:              0x00        USB (Universal host controller spec)
Device ID:              0x265B      Intel(r) 82801FB/FBM USB Universal Host
Controller
Vendor ID:              0x8086      Intel Corporation
Subsystem ID:           0x1999      Unknown
Subsystem Vendor ID:    0x8086      Intel Corporation
Revision ID:            0x02
Interrupt Line:         0x07        IRQ7
```

```
Base Addr. Registers:   0x00000000  Not Implemented
                        0x00000000  Not Implemented
                        0x00000000  Not Implemented
                        0x00000000  Not Implemented
                        0x00001881  I/O Register
                        0x00000000  Not Implemented
PCI Express Device:     No          0 PCI Express devices found
PCI Cap List ID's:      0x00        Not Implemented


Setting                 Value       Description
-----------------------------------------------------------------------------
Class:                  0x0C        Serial bus controller
Sub-Class:              0x03        USB (Universal Serial Bus)
Interface:              0x20        USB2 (Intel enhanced host controller
interface)
Device ID:              0x265C      Intel(r) 82801FB/FBM USB2 Enhanced Host
Controller
Vendor ID:              0x8086      Intel Corporation
Subsystem ID:           0x1999      Unknown
Subsystem Vendor ID:    0x8086      Intel Corporation
Revision ID:            0x02
Interrupt Line:         0x05        IRQ5
Base Addr. Registers:   0xB0004000  Memory-Mapped Register
                        0x00000000  Not Implemented
                        0x00000000  Not Implemented
                        0x00000000  Not Implemented
                        0x00000000  Not Implemented
                        0x00000000  Not Implemented
PCI Express Device:     No          0 PCI Express devices found
PCI Cap List ID's:      0x00        Not Implemented


Setting                 Value       Description
-----------------------------------------------------------------------------
Class:                  0x06        Bridge device
Sub-Class:              0x04        PCI/PCI bridge
Interface:              0x01        Subtractive decode PCI-to-PCI bridge
Device ID:              0x2448      Intel(r) 82801 PCI Bridge
Vendor ID:              0x8086      Intel Corporation
Subsystem ID:           0x0000      Unknown
Subsystem Vendor ID:    0x0000      Unknown
Revision ID:            0xD2
Interrupt Line:         0xFF        IRQ255
Base Addr. Registers:   0x00000000  Not Implemented
                        0x00000000  Not Implemented
                        0x20060600  Memory-Mapped Register
                        0x22802020  Memory-Mapped Register
                        0xB020B010  Memory-Mapped Register
                        0x0001FFF1  I/O Register
PCI Express Device:     No          0 PCI Express devices found
PCI Cap List ID's:      0x0D        Not Defined


Setting                 Value       Description
-----------------------------------------------------------------------------
```

```
Class:                  0x06        Bridge device
Sub-Class:              0x01        PCI/ISA bridge
Interface:              0x00        ISA bridge
Device ID:              0x2641      Intel(r) 82801FBM LPC Interface Controller
Vendor ID:              0x8086      Intel Corporation
Subsystem ID:           0x1999      Unknown
Subsystem Vendor ID:    0x8086      Intel Corporation
Revision ID:            0x02
Interrupt Line:         0x00        IRQ0
Base Addr. Registers:   0x00000000  Not Implemented
                        0x00000000  Not Implemented
                        0x00000000  Not Implemented
                        0x00000000  Not Implemented
                        0x00000000  Not Implemented
                        0x00000000  Not Implemented
PCI Express Device:     No          0 PCI Express devices found
PCI Cap List ID's:      0x00        Not Implemented


Setting                 Value       Description
-----------------------------------------------------------------------------
Class:                  0x01        Mass-storage controller
Sub-Class:              0x01        IDE controller
Interface:              0x8A        Master IDE Device
Device ID:              0x266F      Unknown
Vendor ID:              0x8086      Intel Corporation
Subsystem ID:           0x1999      Unknown
Subsystem Vendor ID:    0x8086      Intel Corporation
Revision ID:            0x02
Interrupt Line:         0xFF        IRQ255
Base Addr. Registers:   0x00000001  I/O Register
                        0x00000001  I/O Register
                        0x00000001  I/O Register
                        0x00000001  I/O Register
                        0x00001811  I/O Register
                        0x00000000  Not Implemented
PCI Express Device:     No          0 PCI Express devices found
PCI Cap List ID's:      0x00        Not Implemented


Setting                 Value       Description
-----------------------------------------------------------------------------
Class:                  0x0C        Serial bus controller
Sub-Class:              0x05        SMBus (System Management Bus)
Device ID:              0x266A      Intel(r) 82801FB/FBM SMBus Controller
Vendor ID:              0x8086      Intel Corporation
Subsystem ID:           0x1999      Unknown
Subsystem Vendor ID:    0x8086      Intel Corporation
Revision ID:            0x02
Interrupt Line:         0x0A        IRQ10
Base Addr. Registers:   0x00000000  Not Implemented
                        0x00000000  Not Implemented
                        0x00000000  Not Implemented
                        0x00000000  Not Implemented
                        0x000018C1  I/O Register
                        0x00000000  Not Implemented
```

```
PCI Express Device:     No          0 PCI Express devices found
PCI Cap List ID's:      0x00        Not Implemented


Setting                 Value       Description
---------------------------------------------------------------------------------
Class:                  0x02        Network controller
Sub-Class:              0x00        Ethernet controller
Device ID:              0x1229      Intel(r) 82557 Fast Ethernet LAN Controller
Vendor ID:              0x8086      Intel Corporation
Subsystem ID:           0x100C      Intel(r) 82544 T Gigabit Ethernet Controller
Subsystem Vendor ID:    0x8086      Intel Corporation
Revision ID:            0x08
Interrupt Line:         0x0B        IRQ11
Base Addr. Registers:   0xB0200000  Memory-Mapped Register
                        0x00002001  I/O Register
                        0xB0100000  Memory-Mapped Register
                        0x00000000  Not Implemented
                        0x00000000  Not Implemented
                        0x00000000  Not Implemented
PCI Express Device:     No          0 PCI Express devices found
PCI Cap List ID's:      0x01        PCI Power Management Interface


Setting                 Value       Description
---------------------------------------------------------------------------------
Class:                  0x02        Network controller
Sub-Class:              0x00        Ethernet controller
Device ID:              0x1069      Intel(r) PRO/100 VM Network Connection
Vendor ID:              0x8086      Intel Corporation
Subsystem ID:           0x1999      Unknown
Subsystem Vendor ID:    0x8086      Intel Corporation
Revision ID:            0x02
Interrupt Line:         0x0B        IRQ11
Base Addr. Registers:   0xB0201000  Memory-Mapped Register
                        0x00002041  I/O Register
                        0x00000000  Not Implemented
                        0x00000000  Not Implemented
                        0x00000000  Not Implemented
                        0x00000000  Not Implemented
PCI Express Device:     No          0 PCI Express devices found
PCI Cap List ID's:      0x01        PCI Power Management Interface



#################################################################################
CPU Information (cpu_info) ######################################################

      Vendor ID:              GenuineIntel
      Type:                   0
      Family:                 6
      Model:                  13
      Brand ID:               22
      Stepping:               2
      CPU Brand String:       Intel(R) Pentium(R) M processor 1.60GHz
      CPU Frequency:          1.20 GHz
      Core Ratio:             12
      Bus Speed:              100 MHz
      Front Side Bus:         400 MHz
      Mobile CPU:             Yes


##############################################################################
Memory Information (memory_info) #############################################

System Memory Info:
      Physical Memory Size:         256 MB
          Top of Low Usable DRAM:   247 MB
          Allocated to TSEG:        1 MB
          Allocated To Graphics:    8 MB

Physical Configuration:
      Channel A:
          DIMM 0:                   128 MB (single sided)
          DIMM 1:                   empty

      Channel B:
          DIMM 0:                   128 MB (single sided)
          DIMM 1:                   empty

      DRAM Technology:
          Channel A:                DDR-II (Dual Data Rate 2)
          Channel B:                DDR-II (Dual Data Rate 2)
          Addressing Mode:          Dual Channel Interleaved
          System Memory Frequency:  400 MHz

Graphics Controller Memory Mapped I/O Information:
      Location:                     B0080000
      Size:                         512 kB



##############################################################################
Video BIOS Information (bios_info) ###########################################

General Information:
  Build:                3302
  Date:                 03/29/2004
  Size:                 59392
  Checksum:             0xF7

VESA BIOS Extension Information:
  Version:              01.00
  PM Version:           1.0

(VBE) Supported Power Management States:
  Reduced On:           Yes
  Off:                  Yes
  Suspend:              Yes
  Standby:              Yes
```

```
Video BIOS Supported VESA Modes:
Mode    Width       Height    Depth   Frequencies (Hz)
--------------------------------------------------
0x101   640         480       8bpp    60
0x103   800         600       8bpp    60
0x105   1024        768       8bpp    60
0x107   1280        1024      8bpp    60
0x111   640         480       16bpp   60
0x112   640         480       32bpp   60
0x114   800         600       16bpp   60
0x115   800         600       32bpp   60
0x117   1024        768       16bpp   60
0x118   1024        768       32bpp   60
0x11A   1280        1024      16bpp   60
0x11B   1280        1024      32bpp   60
0x13A   1600        1200      8bpp    60
0x13C   1920        1440      8bpp    60
0x14B   1600        1200      16bpp   60
0x14D   1920        1440      16bpp   60
0x15A   1600        1200      32bpp   60


#############################################################################
Video BIOS Tests (bios_crc) #################################################

Video BIOS checksums match
Calculated checksum is       0xF7
Video BIOS stored checksum is 0xF7


#############################################################################
Display Data Channel Information (ddc_info) #################################

DDC1 is not supported.
DDC2 is supported at I2C address A0.
DDC2 is not supported at I2C address A2.
DDC2 is not supported at I2C address A6.
Multiple video ports are not supported.
Screen is not blanked during reading of EDID table.
Estimated time to transfer one 128 byte EDID block: 1 second(s)

EDID Structure Information:
      Version:          1.0
      Checksum:         5Ah

Manufacturer Information:
      EISA Code:        NEC
      Model Code:       53B6h (21430 dec)
      Serial Number:    1
      Date (Wk/Yr):     18/1995

Basic Display Parameters & Features:
      Signal Type:      Analog
      Signal Level:     Reference White Above Blank:   0.700
                        Level of Sync Tip Below Blank: 0.300
                        Voltage:                        1.000 V p-p
                        Blank-to-black setup or pedestal is not expected.
                        Separate syncs is supported.
                        Composite sync is supported on Hsync line.
                        Sync on green video is supported.
                        Serration of the Vsync pulse is not required when
                        composite sync or sync-on-green video is used.
      Max Image Size:   40 cm (W) x 30 cm (H)
      Gamma:            2.80
      DPMS Flags:       Standby is supported
                        Suspend is supported
                        Active Off is supported
      Display Type:     R/G/B color
                        Primary color space is not the default color space.
                        Preferred timing mode is not indicated
                        in first detailed timing block.
                        GTF standards based timings are not supported.

Color Characteristics:
       Chroma Information
           |  X   |  Y   |
      -----+------+------+
      Red  |0.6250|0.3398|
      Green|0.3066|0.5947|
      Blue |0.1504|0.0654|
      White|0.2813|0.3105|
      -----+------+------+

Established Timings Supported:
      Resolution  @ Refresh Rates (Hz)
      --------------------------------
       640 x  480 @ 60 67 72 75
       720 x  400 @ 70 88
       800 x  600 @ 56 60 72 75
       832 x  624 @ 75
      1024 x  768 @ 60 70 75 87
      1152 x  870 @ 75
      1280 x 1024 @ 75

Detailed Timings:
      Descriptor 1: Detailed Timing Descriptor
            Pixel Clock:                          28.32 MHz
            Horizontal Active Area:               720 pixels
            Horizontal Blanking Area:             180 pixels
            Horizontal Sync Offset:               18 pixels
            Horizontal Sync Pulse Width:          108 pixels
            Horizontal Border:                    9 pixels
            Vertical Active Area:                 400 lines
            Vertical Blanking Area:               49 lines
            Vertical Sync Offset:                 12 lines
            Vertical Sync Pulse Width:            2 lines
            Vertical Border:                      7 lines
            Image Size:                           390 x 293 mm
            Flags:                                Non-Interlaced
                                                  Normal display (no stereo)
                                                  Digital separate.
```

```
                                                Vsync polarity: +
                                                Hsync polarity: -

      Descriptor 2: Detailed Timing Descriptor
            Pixel Clock:                        80.00 MHz
            Horizontal Active Area:             1024 pixels
            Horizontal Blanking Area:           288 pixels
            Horizontal Sync Offset:             24 pixels
            Horizontal Sync Pulse Width:        96 pixels
            Horizontal Border:                  0 pixels
            Vertical Active Area:               768 lines
            Vertical Blanking Area:             34 lines
            Vertical Sync Offset:               0 lines
            Vertical Sync Pulse Width:          2 lines
            Vertical Border:                    0 lines
            Image Size:                         390 x 293 mm
            Flags:                              Non-Interlaced
                                                Normal display (no stereo)
                                                Digital separate.
                                                Vsync polarity: -
                                                Hsync polarity: -

      Descriptor 3: Detailed Timing Descriptor
            Pixel Clock:                        187.00 MHz
            Horizontal Active Area:             1600 pixels
            Horizontal Blanking Area:           504 pixels
            Horizontal Sync Offset:             64 pixels
            Horizontal Sync Pulse Width:        192 pixels
            Horizontal Border:                  0 pixels
            Vertical Active Area:               1200 lines
            Vertical Blanking Area:             67 lines
            Vertical Sync Offset:               2 lines
            Vertical Sync Pulse Width:          7 lines
            Vertical Border:                    0 lines
            Image Size:                         390 x 293 mm
            Flags:                              Non-Interlaced
                                                Normal display (no stereo)
                                                Digital separate.
                                                Vsync polarity: -
                                                Hsync polarity: -

      Descriptor 4: Detailed Timing Descriptor
            Pixel Clock:                        49.50 MHz
            Horizontal Active Area:             800 pixels
            Horizontal Blanking Area:           256 pixels
            Horizontal Sync Offset:             16 pixels
            Horizontal Sync Pulse Width:        80 pixels
            Horizontal Border:                  0 pixels
            Vertical Active Area:               600 lines
            Vertical Blanking Area:             25 lines
            Vertical Sync Offset:               1 lines
            Vertical Sync Pulse Width:          3 lines
            Vertical Border:                    0 lines
            Image Size:                         390 x 293 mm
            Flags:                              Non-Interlaced
                                                Normal display (no stereo)
                                                Digital separate.
                                                Vsync polarity: +
                                                Hsync polarity: +

Flat Panel Information:
      Supplemental VBE Extension Information:
            Signature:                          VBE/FPÿ
            Version:                            01.00
      Basic Display Parameters & Features:
            Panel Type:                         TFT (Active Matrix)
            Type:                               Color
            Flat Panel Contruction:             Single Panel
            Flat Panel Size:                    1600 x 1200 pixels
            RGB Bits Per Primary:               8:8:8
            Reserved Bits Per Primary:          0


##############################################################################
Graphics Memory Data Line (memory_data_line) #################################

Performed graphics memory data line test on address 00bffffb


##############################################################################
Graphics Memory Address Line (memory_address) ################################

Performed graphics memory address line test on range 00000000-00bfffff


##############################################################################
Graphics Memory Stuck Bit (memory_stuck_bit) #################################

Performed graphics memory stuck bit test on range 00000000-00bfffff


##############################################################################
Graphics Memory Burst (memory_burst) #########################################

Performed graphics memory 1 byte block burst test on range 00000000-00bfffff
Performed graphics memory 2 byte block burst test on range 00000000-00bfffff
Performed graphics memory 4 byte block burst test on range 00000000-00bfffff
Performed graphics memory 8 byte block burst test on range 00000000-00bfffff
Performed graphics memory 16 byte block burst test on range 00000000-00bfffff
Performed graphics memory 32 byte block burst test on range 00000000-00bfffff
Performed graphics memory 64 byte block burst test on range 00000000-00bfffff


##############################################################################
Register Read/Write Test (registers) #########################################

Register Read/Write test passed


##############################################################################
Graphics Pipeline (command_exec) #############################################
```

```
Primary Ring 0 command execution passed
Primary Ring 1 command execution passed
Primary Ring 2 command execution passed
Primary Ring 0 batch command execution passed
Primary Ring 1 batch command execution passed
Primary Ring 2 batch command execution passed


#############################################################################
Hardware Cursor Overlay (render_cursor) #####################################

Non-subjective test for mode 0x111 passed.
Non-subjective test for mode 0x114 passed.
Non-subjective test for mode 0x117 passed.


#############################################################################
2d BLT Rendering (render_blt) ###############################################

2d BLT non-subjective test for mode 0x111 passed.
2d BLT non-subjective test for mode 0x114 passed.
2d BLT non-subjective test for mode 0x117 passed.


#############################################################################
3d Rendering (render_3d) ####################################################

3d triangle rendering non-subjective test for mode 0x111 passed.
3d triangle rendering non-subjective test for mode 0x114 passed.
3d triangle rendering non-subjective test for mode 0x117 passed.


#############################################################################
Final report ################################################################


All the selected tests passed with no failures.
```

12. Appendix E: Sample Metrics File

Now we will examine the metrics log file created. The metrics log file outputs data for only those tests that were run, in the order that they were run. The following is the metrics file generated by the configuration file specified in Appendix A.

```
Test Metrics                                                     Mon Apr  5 09:41:49 2004

No. Subsection test                                                Elapsed Time (sec)
--- ---------------                                                ------------------
 1  PCI Device Information (pci_info)                                          0.1098
 2  CPU Information (cpu_info)                                                 0.0000
 3  Memory Information (memory_info)                                           0.0000
 4  Video BIOS Information (bios_info)                                         0.0000
 5  Video BIOS Tests (bios_crc)                                                0.0000
 6  Display Data Channel Information (ddc_info)                                0.1098
 7  Graphics Memory Data Line (memory_data_line)                               0.0549
 8  Graphics Memory Address Line (memory_address)                              0.0549
 9  Graphics Memory Stuck Bit (memory_stuck_bit)                               4.9410
10  Graphics Memory Burst (memory_burst)                                      27.9990
11  Register Read/Write Test (registers)                                       0.1098
12  Graphics Pipeline (command_exec)                                           0.0549
13  Hardware Cursor Overlay (render_cursor)                                   24.8697
14  2d BLT Rendering (render_blt)                                             13.7250
15  3d Rendering (render_3d)                                                  11.1447
    Test overhead time:                                                        0.0000
                                                                   ------------------
Total elapsed test time:                                                      83.1735
```

All times listed are typical. The elapsed times will vary depending on your individual hardware and software configurations. Timing granularity is 0.001 seconds.