3D*labs*®

PERMEDIA2® *Software Release Note for Windows 95 Display Driver*

Release 4.10.01.2103-0231 Issue 12

Email: info@3dlabs.com
Web: http://www.3dlabs.com

3D*labs Inc.*
181 Metro Drive, Suite 520
San Jose, CA 95110
United States

Tel: (408) 436 3455
Fax: (408) 436 3458

3D*labs Ltd.*
Meadlake Place
Thorpe Lea Road, Egham
Surrey, TW20 8HE
United Kingdom

Tel: +44 (0) 1784 470555
Fax: +44 (0) 1784 470699

Change History

Document	Issue	Date	Change
	10	1 July 97	Updated style
	11	25 July 97	Updated for Release 167
	12	19 Nov 97	Updated for Win95 OpenGL 1.1.22 Release

Contents

Tables

1. Introduction

This document describes the Windows 95 Display Driver for PERMEDIA 2 reference boards.

The driver includes support for Direct3D, DirectDraw, OpenGL and Heidi.

1.1 Release Identity

Once the driver has been installed and the machine rebooted, the display driver release number can be determined by inspecting the Driver page of the Device Manager property sheets for the installed board.

The glint.drv binary should show version 4.10.01.2103-0231.

The 3dogldrv.dll binary should show version 4.10.01.2103-0315.

Note: You need to right-click and choose 'properties' on the glint.drv file to display the version.

1.2 Prerequisites

- Windows 95 (Final Release), OEMSR2 or Memphis
- Pentium processor or later.
- PERMEDIA 2 reference board.

1.3 Important Information

This release has been QA'd only for PERMEDIA 2 boards. It may also run on GLINT 500TX, GLINT MX and PERMEDIA NT boards, but these are not supported.

It is compatible with DirectX 2, 3 and 5. Subsequent releases of this driver can also be obtained from the 3Dlabs web site:

http://www.3dlabs.com

Technical Support Contacts:

Email: support@3Dlabs.com

Tel: 408 436 3455, +44 (0) 1784 470 555

Fax: 408 436 3458, +44 (0) 1784 470 699

2. Installation

Note: If you are upgrading from the 4.03.00.2101-0071 Release Candidate 2 then you will be warned that three files have a different language. This was a bug in RC2 and you should reply NO when asked if you want to keep the old file in all three cases, so that Windows 95 DOES install the files from the final release.

2.1 Fitting a new card with Windows 95 already installed

Please refer to the original instructions provided with your PC for information on how to replace the graphics board. Remember to use anti-static precautions at all times to protect the hardware from static electricity. If in any doubt about the installation consult a qualified PC engineer.

Once your original graphics board has been removed, replace it with the 3Dlabs card. Ensure that it is firmly seated and secured to the machine using the appropriate fastenings.

Note: The PERMEDIA 2 boards are VGA enabled, and there should be no other VGA in the machine once the card has been fitted. If your PC motherboard has built-in VGA it should be disabled. Again refer to the original equipment manufacturer's manuals for details on how to do this.

Fit the PC casing, switch on and boot Windows 95.

After the login procedure, your new card should be detected and Windows will ask for the correct drivers. Insert the floppy disk containing the 3Dlabs drivers and install them. Upon completion, Windows will reboot using the new card. PERMEDIA 2 boards have more than one PCI device on the board, which means there may be some extra reboots as Windows 95 detects the other devices present. You should start in at least a 16Bit mode for Direct3D acceleration.

2.2 Installing Windows 95 with a 3Dlabs board in the PC

Install Windows 95 in the usual way. When installation has completed, Windows 95 should be booted on the VGA. It will not have recognised the 3Dlabs graphics card on installation.

Boot Windows 95 into safe mode. This can be done by pressing “F8” just after the machine has rebooted, when the message “Starting Windows 95…” appears.

Open the device manager. To do this right-click on the “My computer” icon and choose “Properties”. Once the device manager is open, select the “Device Manager” tab at the top of the property sheet.

Look at the list of devices on your machine. If there are any listed under "3D Accelerators", such as "Delta Geometry Processor" or "GLINT 500 TX", remove them. Next look at the "Display Adapters" section. All the devices listed in this section should be removed. Finally look in the "Other" section, and remove all the devices in this section.

Close the device manager and reboot the machine.

After the login procedure, your new card should be detected and Windows will ask for the correct drivers. Insert the floppy disk containing the 3Dlabs drivers and install them. Upon completion, Windows will reboot using the new card. PERMEDIA, PERMEDIA NT, PERMEDIA 2, GLINT 500TX and GLINT MX boards have more than one PCI device on the board, which means there may be some extra reboots as Windows 95 detects the other devices present. You should start in at least a 16Bit mode for Direct3D acceleration.

2.3 Updating existing drivers

Whilst running Windows95, open the control panel and choose "Display".

Navigate to the "Settings" tab at the top of the panel, and choose "Change Display Type" or "Advanced Properties". From here you can change the driver. Select "Change" and "Have Disk". Windows will now prompt for the installation floppy. Follow the usual procedure to install the drivers.

Windows will now reboot your machine using the new drivers.

2.4 Updating your monitor type

The 3Dlabs drivers will try to use the optimal refresh rate that the monitor can handle. For this reason it is important to correctly set up the monitor type to match your display. If the monitor is over-specified in the Windows95 control panel settings, it is possible that the drivers will attempt to use a refresh rate that is beyond the capabilities of your monitor. To correct this, please ensure that you have correctly specified your monitor type. The monitor settings can be found in the "Display" settings in the control panel.

The above procedures should work correctly for the 3Dlabs reference boards. If you experience problems with installing the drivers, and you have previously installed 3Dlabs drivers on the machine, there may be a conflict. Follow the steps below to resolve it:

Boot Windows 95 into safe mode. This can be done by pressing "F8" just after the machine has rebooted, when the message "Starting Windows 95…" appears.

Open the device manager. To do this right-click on the “My computer” icon and choose “Properties”. Once the device manager is open, select the “Device Manager” tab at the top of the property sheet.

Look at the list of devices on your machine. If there are any listed under “3D Accelerators”, such as “Delta Geometry Processor” or “GLINT 500TX”, remove them. Next look at the “Display Adapters” section. If any 3Dlabs devices, e.g. “PERMEDIA” or “GLINT Compatible VGA”, are listed remove them.

Close the device manager.

Open explorer and navigate to the “Windows” directory. In here you will find a hidden directory called “Inf”. Some releases of Windows 95 use a different structure to others inside this directory, so some of the following files may not be present on your release. Find any file names in this directory that begin with “oem”. Examine each of the OEM files, if you have them, in turn by double clicking to load them into Notepad. If any of the files contain references to 3Dlabs, PERMEDIA, or GLINT, delete them. You also need to delete the file called 3dlabs.inf from this directory. Look in the “Other” directory, if you have one, and delete any files whose name starts with 3Dlabs.

Windows is now clean of all references to 3Dlabs hardware. Reboot the machine normally and auto-detection of the hardware will begin again. Remember to supply the correct driver disk when it is requested.

3. Resolutions and Color Depths

The following modes are supported by this display driver release. This driver does not accelerate Direct3D in 256 color modes, although it does accelerate DirectDraw in these modes. For hardware acceleration, please ensure you are running in a High Color or True Color display mode. If a Direct3D sample application is started in 256 color mode, it may attempt to change color depth to High Color to work with the card.

The resolutions supported for Windows display are shown below.

Note: DirectDraw driver will support additional resolutions to the Windows desktop (i.e 512 x 384, 320 x 240, 320 x 200 on the PERMEDIA 2). The memory requirements will differ depending upon the applications you run. If the Direct3D/DirectDraw applications complain that there is not enough memory, try reducing screen resolution.

For most modes, all refresh rates between 60Hz and 100Hz are supported by the driver, although the actual maximum resolution and refresh rate that you can achieve will depend on your monitor setting. The following tables show what resolutions are available and the Maximum refresh rate achievable in each mode.

Resolution	256	High Color (32768 colors)	True Color
640x400	100Hz	100Hz	100Hz
640x480	100Hz	100Hz	100Hz
800x600	100Hz	100Hz	100Hz
1024x768	100Hz	100Hz	100Hz
1152x870	100Hz	100Hz	87Hz
1280x1024	100Hz	100Hz	69Hz
1600x1200	100Hz	85Hz	Not Supported

Table 3.1 **PERMEDIA 2 Refresh Rates**

4. DirectX Support

The driver contains support for hardware optimized DirectDraw and Direct3D functionality. The driver is compatible with DirectX 2, 3 and 5. Please obtain a release of this software if you require DirectDraw / Direct3D acceleration (you are advised to choose latest release – DirectX5). This release of the DirectX drivers supports PERMEDIA 2 and MX silicon.

Global changes in this release include:

Performance enhancements – Significant gains in performance can be seen in this release – these are particularly apparent on 3Dwinbench but benefits are seen in almost all Direct3D applications.

Registry location change – All DirectX registry keys now reside in: \\HKEY_LOCAL_MACHINE\software\3Dlabs\DirectX

P2 Support – This release supports the new PERMEDIA 2 chip.

New PERMEDIA features in this release include:

Linear Allocator – Previous releases of the drivers reported 2 heaps for PERMEDIA. This led to confusion about the amount of free texture/z-buffer memory. This release removed this restriction and reports one simple heap that can be used for all surface types. On DirectX 5 "GetAvailableVidMem" is also supported to assist in the correct reporting of available memory.

MipMapping – PERMEDIA 2 now supports a Per-Polygon Mipmapping scheme.

AGP support – PERMEDIA 2 can texture from AGP system memory on this release of the drivers. This requires a recent build of Memphis, or OEMSR2 with the Intel VGARTD VXD. Also DirectX 5 must be installed on the system. Additionally if a registry key is set PERMEDIA 2 can be forced to texture from AGP memory for ALL texture maps that are allocated – see the additional information section. By default AGP memory will be used when conventional video memory runs out. This is transparent to legacy applications.

8 Bit Index Textures – PERMEDIA 2 supports 8 Bit index textures, and the 8Bit Paletted texture is now reported from the Direct3D driver . The texture LUT is stored in the SGRAM leading to very fast palette updates.

Use of DMA – Interrupt driven DMA is now the preferred (and default) way of interacting with Permedia 2 under DirectX on Pentium Pro class machines and higher. Pentium class machines will still use PIO in the same way as the preceeding releases. The default setup is for the driver to allocate two 64K buffers, individual buffers are allocated to individual D3D contexts and are each split into 8 sub-buffers. Sub-buffers are passed to the graphics processor by an interrupt handler.

Triangle Flags – Permedia 2's renderers no longer rely on the triangle flags to maximize performance, instead the driver takes responsibility for determining the optimal set of vertices on the chip to update. This has the significant benefit of fixing many of the applications which incorrectly handled the flags with no loss of speed or efficiency.

Miscellaneous improvements – Many of the changes to the driver rely on facilities available only in DirectX 5, for this reason it is strongly recommended that users install DirectX 5 in preference to earlier versions. Because of Microsoft's policy of full backward compatibility between DirectX versions it is recommended that users install the most recent version of DirectX at all times..

Bugs fixed in this release:

Various ChromaKeying and AlphaBlend bugs – Both GLINT and PERMEDIA did not correctly ChromaKey in some circumstances, and particularly 4Bit paletted textures were never ChromaKeyed.

Textures filling bug – PERMEDIA did not correctly color fill texture map surfaces in the last release.

MDK D3D fix – The game MDK with the D3D patch did not run on the last release.

Triangles missing in 3DWinbench – 3DWinbench showed up a bug in the Direct3D component that drives the HAL, which was causing triangles to be ommitted or mis-drawn. As there was nothing that could be done in the driver to work around this the triangle flag support has been removed; instead the driver will aggressively mesh the triangles its-self.

Texture Patching bug in DirectX 5 – Due to texture download changes in DirectX 5, the PERMEDIA patching scheme was broken. This has been worked around in the driver.

Lines drawn through Delta incorrectly placed – Drawing lines in a Delta based renderer was broken in the last release due to incorrect setup. This is fixed in this release.

ROCKEM3D Fix – When using DMA on either PERMEDIA or GLINT a synchronization problem was visible where rendered text would sometimes appear *behind* polygons. This was due to Rockem and some other applications not correctly obeying the NO2DDURING3DSCENE flag. This has been worked around in the driver and the flag is no longer set. This also fixes some versions of Hellbender which check for this flag and do not run if the flag is present.

PCPlayer Benchmark drawn in blue – Some applications were confused by the combination of RGB and MONO rendering modes in the driver and incorrectly chose MONO rendering. To eliminate the confusion the MONO flags have been removed from the driver. This does not effect rendering performance.

Partial Texture Updates – Applications (such as Populous 3, from Bullfrog) that wish to download only a sub-rectangle of a texture map can now do so. Previous implementations incorrectly arranged for the download to always update only the top left of a texture, irrespective of the requested download coordinates.

4.1 Additional Useful Information

Some registry keys are not exposed through the control panels but are useful to know. All DirectX Registry keys live in \\HKEY_LOCAL_MACHINE\SOFTWARE\3Dlabs\DirectX.

The following values can be set. They are all stored in the registry as 'String' Values.

Direct3DHAL.AppStats = 0 or 1 – Enables debug output of driver statistics such as state changes/frame, etc. This is intended for use only by developers.
Direct3DHAL.DisableAlphaTextures = 0 or 1 – Disables reporting of Alpha based texture formats.
Direct3DHAL.NoVBlankWait = 0 or 1 – Disables waiting for the Video Blanking interval before a page flip. Useful to measure the real speed of rendering when the application is not quantized to VBLANK.
Direct3DHAL.NoFlip = 0 or 1 – Causes the driver to ignore page flipping requests. Useful to enable scrolling down in video memory with the DbgTrace tool while page flipping.
Direct3DHAL.FilterVideo = 0 or 1 – Enables video filtering during video playback.

5. Known Anomalies and Restrictions

The following errors have been noted during testing of this software release:

Starting The Lucas Arts game "Jedi Knight" from a 256 colour desktop will lock the machine. The workaround is to switch to a higher colour resolution before launching the game.

G-Police published by Psygnosis will not run on a 4MB PCI PERMEDIA 2 board. No work around is available yet, but we are working on resolving the problem.

Laplink does not function correctly using the remote control option, due to application bugs. Old versions of the application will not run remotely at all, where newer versions run with many black icons and bitmaps.

6. OpenGL

The driver provides support for OpenGL version 1.1. This graphics API library is a part of the OEM SR2 Windows95 release. (There is a self-extracting executable opengl95.exe available from Microsoft and included with the 3Dlabs drivers which provides all the required libraries for those using the original retail release of Windows95). This enables hardware acceleration of OpenGL applications and requires installation of DirectX 3 (or later) on which it depends for pixel buffer allocation (back buffer, z-buffer, texture storage etc).

The last two digits of the OpenGL version number returned by a call to glGetString(GL_VERSION) (i.e 1.1xx) are updated with every major release of the ICD. The version for this release is 1.1.22.

All OpenGL version 1.1 API calls are now implemented in release 1.1.22 and later.

6.1 OpenGL games

The following web page URL provides a useful FAQ for users of iD Software's Quake, Quake 2 and Hexen titles.

http://www.3dlabs.com/enduser/quake.html

7. OpenGL Known Anomalies and Restrictions

The DirectX surfaces upon which OpenGL applications depend will be lost during a display resolution mode change, disabling rendering but allowing the application to be safely shutdown.

The standard maze screen saver does not get hardware accelerated. This is due to a bug in the Microsoft screen saver library. A customized accelerated 3Dlabs version that also supports linear filtering in the settings option has been provided with this release.

Support for texture memory caching is available for the Permedia family only in this release (NOT for GLINT chips).

Support for the Overlay planes extension is not provided in this release.

Support for hardware OpenGL acceleration on a 256 colour desktop is no longer supported in this release.

8. OpenGL Texturing & Extensions

8.1 Efficient use of multiple textures

OpenGL applications that wish to render primitives with multiple texture maps will achieve much higher performance by avoiding the invoking of the different textures in immediate mode. There are two alternative options for efficient switching between multiple textures.

The first and much preferred option is to use the OpenGL texture object functionality. This is available with 3Dlabs Installable Client Driver as part of the standard API for OpenGL version 1.1 on Windows 95. Texture objects are fully editable and may have their images and parameters altered at any time (unlike the use of textures in display lists). Details on texture object functionality is available in the OpenGL 1.1 specification (Appendix E) and also in Appendix D of the 3Dlabs OpenGL Extensions document. The performance gain using this approach will benefit performance for PERMEDIA 2.

The second option is to define each texture (or array of mip map resolutions) within a display list - *with the limitation that only one texture is allowed per display list*. Switching between different textures is then achieved by referencing the appropriate display list. Since display lists are not editable in OpenGL, the OpenGL implementation is able to cache texture data defined within a display list. In effect the display list identifier acts as a texture handle. This caching cannot be performed when a texture is invoked in immediate mode since the application in this case is at liberty to have changed the texture data since any previous reference.

From release 1.1.20 onwards this non-standard behaviour can be disabled with the registry variable OpenGL.DisableDisplayListTextureSwitch - this will prevent conflicts with texture object routines in display lists (e.g. glBindTexture etc) and also prevent glTexImage calls from downloading texel data in compile only display lists (hence overwriting any existing immediate mode texture).

8.2 Texture Memory Cache Management

For the PERMEDIA 2, texture data is stored in the off screen memory on the graphics card. The memory available for textures is therefore constrained by the off screen memory available. It is also constrained by the amount of memory already consumed for the depth buffer, stencil buffer, and the display, which varies according to the current display resolution in use. I.e. there is more memory available for textures when the display resolution (and therefore the size of the display, depth buffer, stencil buffer memory requirements) is lowered.

On 3Dlabs OpenGL releases prior to version 1.1.22, if the condition is reached where there is insufficient off screen memory to load a new texture then the OpenGL texture download will not succeed and will set the error code GL_OUT_OF_MEMORY. Textured primitives that expected to use this texture will not be rendered correctly.

To improve on this behavior a scheme for swapping textures to/from system host memory has been implemented for the PERMEDIA chip family. By setting aside a portion of texture memory on the graphics card for use as a texture cache and tracking when a texture switch takes place, textures can be reloaded to the cache as needed from a copy kept in host memory when the texture was first downloaded. If the requested texture is already present in the cache, then no reload is performed.

Ideally for the greatest flexibility and most efficient use of available texture memory, all textures should be cacheable. However for a software texture cache manager there is a small performance overhead to be paid for this tracking plus any delay in reloading a swapped out texture (as a texture could be swapped out at any time by another OpenGL process). By allowing the user to specify the size of the texture cache through the use of the registry variable OpenGL.MaxTextureSize , an approximate balance between non-swappable and swappable textures can be made (and hence performance). Thus an application should load any real-time critical textures first as the texture manager will only place textures in cache and/or host memory if a space of sufficient size is not available in the non-swappable area of texture memory.

In order to guarantee all texture requests no matter how large, any texture whose size in texels is greater than the cache size will be silently filtered down to fit in the cache (while preserving aspect ratio).

As noted under the description for OpenGL.MaxTextureSize, the texture cache manager can be disabled by setting this registry variable to zero so that all textures are non-swappable as in previous releases.

8.3 Texture Filter Modes

The default texture minification filtering for OpenGL involves mip-map filtering. This gives good textured rendering quality but at the cost of low performance. Much higher performance can be obtained by changing the default texture filtering such that the minification and magnification filtering modes are the SAME. Setting them to GL_LINEAR gives good quality bilinear filtering and improved performance. Setting both modes to GL_NEAREST will give nearest neighbour filtering and the fastest possible performance.

Only linear and nearest neighbour texture filtering modes are supported on PERMEDIA 2(mip-map filtering is not supported).

8.4 BGRA Extension

This extension provides an additional pixel color format for compatibility with the blue, green, red component ordering of Microsoft Windows DIB's (device independent bitmaps). Refer to Appendix C in the 3Dlabs OpenGL Extensions document for further details.

8.5 Palette Texture Extension

The PERMEDIA 2 provides direct support for palette textures, where each texel represents an index into an on-chip RGBA (8-bits per component) lookup-table. An OpenGL palette texture extension has been defined by Microsoft which is supported by 3Dlabs OpenGL ICD from release 1.0.11. The supported texel depths depend on the chip. They are as follows:

PERMEDIA only supports 4 bit texel depths.

PERMEDIA2 only supports 4 bit and 8 bit texel depths.

Besides improving texture performance and reducing the memory requirements for storing textures in the local buffer, by repeatedly updating the texture LUT, animation effects such as real-time color cycling are also possible. Refer to Appendices B and C for further details.

If many textures share the same look-up table (LUT), performance gains can be obtained with paletted texture objects by forcing the textures to share the same palette (particularly for 8-bit palette textures on the MX and P2). The default behaviour when texture switching through calls to glBindTexture is to send down the LUT on every switch. This can be disabled by the EXT_shared_texture_palette extension described in Appendix H of the OpenGL Extensions document.

8.6 3Dlabs Driver Extension

In addition to the extensions mentioned above, the 3Dlabs_DriverState extension has been added and is detailed at Appendix F in the OpenGL Extensions document. This extension is simply a mechanism for adding extra state to the Client Driver and provides extra control to the currently selected context.

8.7 Swap Hint Extension

The Microsoft defined extension GL_WIN_swap_hint is now supported. This extension allows the area of a window swapped by the SwapBuffers call to be restricted. This can give performance benefits when only a small area of the display is being updated at a time. For more information please contact Microsoft directly or search in the latest Win32 help files for glAddSwapHintRectWIN.

9. 3D Studio MAX

This release supports hardware acceleration of both 3D Studio MAX 1 and 2. 3D Studio MAX 1 is accelerated through the Autodesk propriety API, Heidi. 3D Studio MAX 2 is accelerated through OpenGL.

9.1 3D Studio MAX 1.x

The 3Dlabs Heidi driver has the same system requirements as OpenGL. These are that the 1.1 versions of Microsoft's OpenGL binaries be installed and that Direct X3 (or later) is installed. The 1.1 OpenGL binaries are provided as standard on the OEM SR2 release of Windows 95 and are also provided in the self-extracting archive opengl95.exe. This archive can be downloaded from the Microsoft website and is also provided with 3Dlabs drivers.

If PERMEDIA hardware is already in use for 3D Studio Max 1 then simply copy the wglint.hdi file to your 3dsmax/drivers directory to upgrade to this driver. If 3D Studio is not currently using PERMEDIA hardware then first copy the wglint.hdi file to your 3dsmax/drivers directory and then start up 3D Studio Max. Under the File menu choose Preferences and in the resulting dialog click on the Viewports tab. Click on the button labelled Choose Driver and then select GLINT Hardware. You will then need to restart Max to enable GLINT hardware acceleration.

On PERMEDIA based boards best performance will be obtained when running with 32k colors. On GLINT based boards best performance will be provided when running in TrueColor display modes.

Please note that only version 1.2 of 3D Studio MAX 1 is recommended by Kinetix for use under Windows 95. For information on how to upgrade to version 1.2 please contact Kinetix.

9.2 3D Studio MAX 2

When you first run 3D Studio MAX 2 you will be given the option of which graphics driver to use. You should select the OpenGL driver to get hardware acceleration with this driver release. 3Dlabs do not provide a Heidi driver compatible with 3D Studio Max 2. If you wish to change which driver 3D Studio Max 2 uses at a later date then this can be done by selecting Preferences from the File menu. Click on the Viewports tab and then press the Choose Driver button. On the resulting dialog box select OpenGL.

When running with OpenGL you can confirm that you getting hardware acceleration by viewing the About from the Help menu. The driver box in the top right hand corner of the resulting dialog should read 'OpenGL (3Dlabs v1.1.xx)'. If it reads ‘OpenGL (Microsoft Corporation v.1.1.0)’ then you are running through Microsoft’s software OpenGL and should consult the OpenGL section of this document for more information on enabling OpenGL acceleration.

9.3 Known Anomalies

When running 3D Studio MAX 2 with OpenGL acceleration under Win95 you may sometimes experience a crash on exit from the application. This problem also affects Microsoft’s software OpenGL implementation and is under investigation.

When running 3D Studio MAX 2 with OpenGL acceleration under Win95 you may sometimes notice that textures do not appear in some of the viewports. This problem also affects the Microsoft software OpenGL implementation. It will be fixed in a future release of 3D Studio MAX 2