Warp3D_User

Hans-Jörg Frieden

COLLABORATORS

	TITLE : Warp3D_User		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Hans-Jörg Frieden	August 24, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

Warp3D_User

1.1 Welcome

Welcome to Warp3D.

In case you are wondering what exactly this is, Warp3D is a hardware-independent driver system for 3D graphics boards. Warp3D allows the programmer to access 3D Hardware in a completely transparent and independent way, allowing support for any kind of hardware, present-day as well as future.

Warp3D is completely independent from both the hardware present in your system as well as the graphics driver used. This means that it does not matter if you have a CV64/3D running under Picasso96 - you will be able to use any Warp3D-compliant program without modifications. This is one of the major features of the system.

This guide is intended to help you understand the basic concepts of Warp3D, install the drivers on your system, and help you configure it to your personal taste. Although many things will depend on the program that uses Warp3D, you will be able to configure a few of the basic features of Warp3D to your personal preference.

1.2 Requirements

Requirements

Since Warp3D is a product primarily intended for 3D graphics games and applications, it is required that you have at least a 68040 or 68060 with a Floating Point Unit. Theoretically, Warp3D will also run on a 68030 with an 68881 or 882, but practically, this configuration is too slow. Warp3D also runs with a PowerPC processor.

Additionally, you will need a graphics card with a 3D chipset, like the CV64/3D or the Permedia2 equipped CV/BVisionPPC. As soon as a CPU-only driver is available, you will no longer need this in order to run Warp3D applications and games, although they will be an order of magnitude slower. There is currently no support for AGA, although this might be included in a later release.

Warp3D will run on any Kickstart 3.x equipped system. You will also need a supported graphics driver, like CyberGraphX (V3 or better) or Picasso96 (v1.40 or better). If you have a non-compatible system (like Merlin emulation or the old Picasso stuff), you might need to look out for a separate driver archive, or you might try to do it yourself.

1.3 Installation

Installation

The most obvious and easiest way to install Warp3D is to use the installer script provided with the archive. Just double click the icon and follow the on-screen prompts.

Should you for some reasons decide not to use the installer script (which is not the recommended way), here are a few things you should take care of.

Warp3D requires a fixed set of directories in your LIBS: directory. For the functioning of Warp3D, it is absolutely required that the the files are in the correct place:

LIBS:

Warp3D.library

Warp3DPPC.library

Warp3D/

GFXDrivers/

W3D_CyberGfx.library

W3D_CyberGfx_PPC.library

W3D_CyberGfx4.library

W3D_CyberGfx4_PPC.library

W3D_Picasso96.library

W3D_Picasso96_PPC.library

HWDrivers/

W3D_Virge.library

W3D_Virge_PPC.library

W3D_Permedia2.library

W3D_Permedia2_PPC.library

Usually, you just need one of the GFXDrivers and one of the HWDrivers. Due to the way Warp3D searches for its drivers, it is nessessary that the directory structure is maintained that way, and that the driver subdirectories are really created under the LIBS directory. If you fail to do this, Warp3D might not work.

(See why I said this is not the recommended way?)

If you are using a PPC together with Picasso96 and a WarpUp version 3 or earlier, you must add the P96Patch program somewhere into your startup-sequence or user-startup. This pre-loads the Picasso96API.library at startup, since due to the ramlib stack problems, you might get a crash on library initialization. This patch is no longer required with WarpUp V4.

Also note that for PPC usage, WarpUP must be installed on your system.

1.4 Basic Concepts

Basic Concepts

Warp3D is a system of drivers, implemented as shared libraries on both 68k architectures and on the PPC. There is one driver library for each graphics system (CyberGraphX, Picasso96) in the GFXDrivers, one for each Hardware

At each time, there are exactly three libraries open. One of the is the main API library, Warp3D.library. This library will, upon opening, scan the directory GFXDrivers and then HWDrivers for a suitable graphics system and hardware driver. If it has found a suitable gfx driver, it will then scan for a HW driver and open it too, or a CPU driver, if no hardware is installed or no suitable driver was present.

It is this structure that makes Warp3D so flexible. If new hardware becomes available, all it takes is to rewrite the HWDrivers library for that hardware. If a new graphics system emerges, like OS3.5 RTG, only the small GFXDrivers library needs to be rewritten.

1.5 Configuration

Although the programmer retains the most complete control over the Warp3D system, there are a few items that you, the user, can influence.

Configuration options are expressed with environment variables. Warp3D expects all variables to reside in a subdir called 'Warp3D' in the ENV: directory. Usually, those variables that affect the main library reside directly in the ENV:Warp3D directory. Those that affect a driver directly must be put into a directory that is named exactly like the driver, for example, the W3D_Virge.library variables reside in the ENV:Warp3D/Virge. Note that variables are only valid when the driver is really loaded. Variables for the ViRGE driver are ignored by every other driver.

Variables can either have a boolean, numerical or string contents. Boolean variables are considered to be true if they contain the string "1" or "on" (without quotes, case insensitive), false if set to "0" or "off", and use the default otherwise.

The main library reads the following variables when opened:

Warp3D/CPUDriver (string)

Specifies the preferred CPU Driver. If this variable is unset, Warp3D selects the first CPU Driver it finds, unless a hardware driver was found.

Warp3D/QueueSize (numeric)

Specifies the size of the internal queue buffer for indirect rendering. This defaults to 16384 bytes, but can be overridden with this variable. Sizes below 8192 are rejected.

The ViRGE driver reads the following variables when opened:

Warp3D/ViRGE/FastFilter (boolean)

If true, a bug in the ViRGE chip is exploited for a bit faster bilinear filtering. This will result in a bit faster display, but also in a reduction of display quality.

Defaults to false, and may be overridden by the programmer via Hinting.

Warp3D/ViRGE/Fog (string)

This variable determines the way the ViRGE driver handles fogging. Since the ViRGE does not support every fog mode available, some fogging stuff can be emulated. If this variable is unset, everything is rejected except interpolated fogging. If set to "linear" (without quotes), linear fog is also accepted as being supported. Setting this to "all" will allow any fogging mode to be accepted, although the result will not look correctly. Unset this variable if you are after precise appearance, otherwise set it to something different.

Defaults to unset (interpolated fog only).

Warp3D/ViRGE/FogSubdivide (boolean)

Normally, when a polygon lies only partly inside the fog boundaries, the hardware still interpolates fog over the entire polygon. This may look wrong, especially if the fog area is very tight. Setting this variable to true will subdivide the polygon into an inside and an outside part, where the outside part lies completely outside the fog and is hence drawn unfogged, and the inside part is drawn correctly fogged. Turning on this Feature will usually result in a certain performance overhead, but the visual appearance might be more than worth it.

Defaults to false, and may be overridden by the programmer via Hinting.

Warp3D/ViRGE/Pass24 (boolean)

This variable is for internal use only. Usually, 24 bit textures are converted to 16 bits by the driver, since 24 bit textures do not seem to work. This will force the pass-through of 24 bits texture data through the driver without conversion, but the result is most likely false colors.

Defaults to false

Warp3D/VIRGE/TexSubdivide (boolean)

Due to a limitation of the ViRGE chip, perspectively mapped textures cannot be repeated over 128 pixels. Setting this variable to true will, like the Fog Subdivision, partition polygons so that they lie completely on texture borders. The result will look exact, but this also introduces a speed penalty because of the vector math involved.

Defaults to false, and may be overridden by the programmer via Hinting.

The Permedia2 driver reads the following variables when opened:

Warp3D/Permedia2/Dither (boolean)

If this is set, Warp3D forces output of the Permedia2 to be dithered, regardless of the state set in the application.

Defaults to false

1.6 Frequently Asked Questions

Here are some common problems, and how they can be avoided:

Warp3D doesn't work at all:

It's difficult to give a solution on this topic, because of the variety of possibilities. Check to see if all libraries are installer (at least Warp3D.libray, and one library in each of the two Warp3D/#?drivers directories).

The workbench is trashed under CyberGraphX:

This problem may arise when not enough video ram is available. A possible solution is to reduce the resolution of the workbench. This problem may also occur when switching from a Warp3D application back to workbench. In this case, this is not an error of Warp3D, but the application is not doing correct locking.

The colors of the textures is not correct when using a Warp3D application:

This may have various reasons. One of the most obvious may be the screen mode selected. The ViRGE driver currently only supports 15 and 8 bit screen modes. 16 and 32 bit modes are not supported by the chip itself, and 24 bit is not supported by all driver software.

No 15 bit modes are shown in the screenmode requester:

Using Picasso96: setenv Picasso96/ShowModes all.

using CyberGraphX: setenv CyberGraphX/Hide15Bit 0, or remove the variable.

I can't find the Voodoo 3 driver:

That's because it's not included yet. We're still working on it, and will release it

later.

This list may (will) grow in the future. If you have any problems not covered here, contact

the authors .

1.7 Contact

The authors can be contacted by e-mail under the following addresses:

Hans-Joerg Frieden: hfrieden@uni-trier.de

Thomas Frieden: tfrieden@uni-trier.de

Sam Jordan: s.jordan@haage-partner.com

The most up-to-date version of Warp3D can always be found at the 3D-World web site at http://www.haage-partner.com/3dworld/ .

1.8 Acknowledgments

We want to express special gratitude towards KDH Datentechnik. The authors received a Voodoo 3 3000 from them, free of charge. Thanks a lot!

The authors also wish to thank all our beta testers for their valuable help and suggestions, and to everyone who gave hints and advice. Thanks also go to Haage&Partner for supporting us.