AMD Geode™ LX Processor Windows® XP Display Driver Customization Guide

June 2006

Publication ID: 40283B

Contacts

www.amd.com

Trademarks

Contents

1 Overview

This document details the methods by which end users can customize the functionality and features provided by the AMD Geode™ LX processor Windows® XP display driver. All customizations are accessible through the system registry or display driver escapes, so end users should rarely need to create special driver binaries. This is particularly advantageous when new driver binaries are released by AMD with bug fixes or feature additions, as users can simply apply the new binaries to their systems without the need to merge potentially complex changes into an updated code base.

The customization options exposed by the driver are targeted at the frequent need to generate a Windows XP display on a non-standard output, such as a widescreen panel or TV. In most cases, the core driver functionality does not need to change.

1.1 Customization Options

The Geode LX processor XP display driver allows the following levels of customization:

- Invisible customizations, such as whether to favor size or speed when allocating from video memory.
- Complete panel customization. TFT timings, clock rates, and power sequences can be specified for any custom TFT or LVDS panel.
- Complete advanced mode customization. The end user can specify all aspects of one or more display modes, including the output color space and format, pixel depth, etc.
- Driver escapes to enable advanced functionality, such as reading and writing registry settings and toggling between CRT/TFT output modes.

1.2 Customization Sequence

The typical sequence by which end users customize the Geode LX processor XP display driver for their systems is simple:

1) For custom panels or advanced customizations, create a customization text file that details the display details.
2) Compile the text file into a binary file and a custom driver .inf using lxcustom.exe or lxpanel.exe.
3) Install the driver using the custom .inf and binary file.
4) Interact with the system as normal.

2 Registry Settings

A modest amount of functionality can be customized through the use of driver registry keys. These keys are normally programmed when the driver is installed using the driver .inf. However, they can also be modified post-installation through a driver escape. This section details the available registry keys and their impact on normal system operation.

2.1 Display Stride

The registry key FBLinear is used to influence the alignment of graphics data for the primary display. This key has the following meaning.

Key	Description
FBLinear	Alignment restrictions for the display: 0: The stride for the display mode is forced to a power of 2. A power of 2 stride allows line-by-line display compression at the price of increased memory usage for the display. The minimum amount of video memory to support a display mode can be calculated as the display stride multiplied by the display height. For example, a user wants to set 1280x1024x32 bpp. 32 bpp implies 4 bytes per pixel, so the minimum line size is 1280*4 = 5120 bytes. The next power of 2 above this value is 8192 bytes. The minimum amount of video memory to support this resolution is thus 8192*1024 = 8 MB. If the amount of video memory required for a mode is larger than the available amount of video memory, the mode is rejected and is excluded from the mode list presented to the operating system. 1: The stride for the display is forced to a power of 2, unless a power of 2 stride exceeds the amount of available video memory. In these cases, the driver sets a linear pitch where the stride is equal to the display line size. For example, in the above section the minimum amount of video memory needed for 1280x1024x32 bpp with a power of 2 stride was calculated as 8 MB. If the size of video memory is 6 MB, instead of being rejected the mode is set with a linear stride, such that the required mode size is only 5120*1024 = 5 MB. 2: The stride for the display is forced to be linear, such that the display stride is always equal to the line size. For all modes except 1024x768 (for which both the linear and power of 2 strides are identical) line-by-line compression is disabled. Compression can still be enabled. However, the mechanism by which compressed lines are dirtied no longer occurs for individual lines, but for the frame as a whole.

2.2 Compression

The registry key *Compression* is used to choose the criteria by which display compression is configured and enabled for each display mode. Display compression serves to reduce display bandwidth usage for simple screens. This key has the following meaning.

Key	Description
Compression	Compression Criteria: 0: Compression is always disabled. No video memory is allocated to contain the compression buffer. 1: Compression is only enabled if the extra memory needed for the compression buffer does not cause a mode to exceed the amount of available video memory. The calculation to determine the size of a display mode with compression enabled occurs after the calculation for the FBLinear registry key. The resulting mode size and display stride are used to determine the location of the compression buffer, and by extension the minimum mode size. The compression buffer is placed in the unused space at the end of each line if the unused space is greater than or equal to 544 bytes. Otherwise, it is placed after the data for the display mode. To return to the example of 1280x1024x32 bpp, if the mode is set with a power of 2 stride, there will be 8192-5120 = 3 Kbytes of unused space at the end of each line. The compression buffer would thus be placed at the end of each line and no extra space would be required for the display mode. However, if the stride were set to a linear pitch, there would be no unused space at the end of the line and the compression buffer would be placed after the display data. In this case, the compression buffer requires 544 bytes for each line of the display. 1280x1024x32 bpp would require an additional 544*1024= 544 Kbytes. 2: Compression is always enabled. If enabling compression causes the size of the display mode to exceed the amount of available video memory, the mode will be rejected and will not be presented to the operating system.

2.3 Miscellaneous Settings

The remaining registry keys specify the display format and have the following meaning:

Key	Description
DDCEnabled	This key should be set to 1 to enable Display Data Channel communication with the monitor. If this key is non-zero, the DDC_GPIOData and DDC_GPIOClock registry keys should also be set to appropriate values.
BIOSDetectOutput	Setting this key to 1 indicates that the BIOS is responsible for the detection and initialization of the output device. The driver queries the BIOS for the current output device (CRT, TFT, etc.) and any output device settings, such as simultaneous CRT output. When the output is TFT, the driver will not change the panel timings in the Display Filter.
PanelEnable	This registry setting is used to enable or disable TFT output. Setting this key to 1 enables TFT display, while setting it to 0 forces CRT output. When BIOSDetectOutput is 1, this key is initialized by the driver each time the system boots and can only be changed if a panel has been detected. When BIOSDetectOutput is 0, this key is programmable by software and the user and should only be changed when the PanelInSystem registry setting is also 1.
SimultaneousCRT	This key indicates that a TFT display should also be displayed through the CRT DACs. This setting is only valid if a TFT is currently enabled using the PanelEnable key.
PanelWidth	If BIOSDetectOutput is 0, this key is used to manually specify the width of an attached TFT panel. If BIOSDetectOutput is 1, this key is read-only and specifies the width of the attached TFT panel, if detected by the BIOS.

Key	Description
PanelHeight	If BIOSDetectOutput is 0, this key is used to manually specify the height of an attached TFT panel. If BIOSDetectOutput is 1, this key is read-only and specifies the height of the attached TFT panel, if detected by the BIOS.
XVGAPanel	Setting this key to 1 indicates that the attached TFT panel is a 2 pixel-per-clock XVGA TFT panel. It is only valid if the BIOSDetectOutput key is 0 and the PanelEnable key is 1.
DDC_GPIOData	This key specifies the AMD Geode™ CS5535/CS5536 companion device GPIO pin to which the DDC data line is wired for CRT communication. This key is only needed if the DDCEnabled key is non-zero.
DDC_GPIOClock	This key specifies the CS5535/CS5536 companion device GPIO pin to which the DDC clock line is wired for CRT communication. This key is only needed if the DDCEnabled key is non-zero.
EnablePanelScale	Setting this key to 1 enables display modes that are smaller or larger than the current TFT panel size to be scaled to fit the panel display. If this key is 0, display modes are centered or panned. Only display modes less than or equal to 1024 pixels in width can be scaled.
EnableNon60HzPanel	When a TFT panel is attached, the display driver normally filters out all non-60 Hz modes, as the driver TFT timings are all 60 Hz timings. This mode disables that mode filter such that all refresh rates are presented to the operating system. This is useful when enabling CRT-only output, as users can then generate higher refresh rates on the CRT.
RotationAngle	Specifies an angle of rotation in degrees clockwise. 0, 90, 180 and 270 degree rotations are supported. The display driver adjusts the modes that are presented to the operating system based on this value. When a portrait mode configuration is selected, modes are presented as *height_x_width* instead of *width*_x_*height*.
PanelInSystem	Specifies that the driver installation was performed for a system attached to a TFT. The driver uses this flag when resuming from standby or after a reset. If this flag is set to 1 and PanelEnable is currently 0, the driver will look for an attached CRT monitor using DDC. If a CRT is not found, the driver reverts to TFT output, forcing the PanelEnable setting to 1.

3 Driver Escapes

The AMD Geode™ LX processor Windows® XP display driver includes several backdoor functions to expose functionality not offered through the Windows API. These functions can be accessed with the GDI routine, ExtEscape, as demonstrated in the following code sequence:

```
unsigned long array[3];
array[0] = 0xABADD00D;
array[2] = 1;

HDC dc = GetDC (NULL);
ExtEscape (dc, 0x404, 12, (LPCSTR)&array[0], 12, (LPSTR)&array[0]);
```

3.1 Get/Set Registry Settings

The first two driver escapes are used to update and retrieve the current display driver registry settings. These escapes are used by the display applet that ships with the standard XP display driver. A single structure holds all registry information. The structure is defined as follows:

```
typedef struct tagLXRegistrySettings7
{
  unsigned long secretWord;           // Identifier to indicate valid buffer
  unsigned long unused;               // Unused DWORD
  unsigned long BIOSDetectOutput;     // Use the BIOS to detect the output device.
  unsigned long Compression;          // Compression Settings (enabled, auto, etc.)
  unsigned long FBLinear;             // Linear frame buffer setting
  unsigned long DDCEnabled;           // Enable DDC communication to a monitor
  unsigned long DDC_GPIOData;         // GPIO for DDC data line.
  unsigned long DDC_GPIOClock;        // GPIO for DDC clock line.
  unsigned long PanelEnable;          // Dynamic state of panel enable.
  unsigned long PanelWidth;           // Manually specified panel width.
  unsigned long PanelHeight;          // Manually specified panel height.
  unsigned long XVGAPanel;            // XVGA Panel for supported panel sizes.
  unsigned long SimultaneousCRT;      // Dynamic flag enabling simultaneous CRT display.
  unsigned long ManualTimingEnable;   // Custom mode timings have been specified.
  unsigned long EnablePanelScale;     // Allow modes to be stretched to fit the display.
  unsigned long EnableNon60HzPanel;   // Allow non-60Hz CRT modes when displaying on a TFT.
  unsigned long CustomPanelTimings;   // Custom panel timings have been specified.
  unsigned long RotationAngle;        // Angle of desktop rotation.
  unsigned long PanelRefresh;         // Detected refresh rate of the panel
  unsigned long PanelInSystem;        // Panel detected with BIOSDetectOutput or panel
                                      // specifically specified at install time.
} LX_REGISTRY_SETTINGS;
```

Before calling ExtEscape, the *secret_word* field should be set to ABADDOODh. The other structure fields correspond to the driver registry settings explained in Section 2.0 "Registry Settings" on page 5. Registry settings, when updated, take effect the next time the display mode is changed.

Table 3-1. Get/Set Registry Escapes

Escape Number	Input Size	Input Description	Output Size	Output Description
1024	80	Pointer to a LX_REGISTRY_SETTINGS structure. Only the secret_word field is relevant.	80	Pointer to a LX_REGISTRY_SETTINGS structure to receive the current registry settings.
1025	80	Pointer to a LX_REGISTRY_SETTINGS structure containing new registry settings to program.	80	Pointer to a LX_REGISTRY_SETTINGS structure. The structure will not be overwritten, but must be included as a safeguard.

3.2 Alpha Window

The Geode LX processor hardware includes basic alpha blending support between the video overlay and the primary graphics surface. However, this functionality is not easily or cleanly exposed through the DirectDraw API. The driver thus provides an escape to toggle alpha blending functionality for the video overlay.

Alpha blending data is provided in the following structure:

```
typedef struct tagLXBlend
{
    unsigned long secret_word;
    unsigned long spare;
    int enable;
    int per_pixel;
    unsigned long fixed_alpha;


} LX_VIDEO_BLEND;
```

secret_word

Should be set to ABADDOODh before calling ExtEscape. This value helps to prevent spurious or accidental driver escapes.

spare

Currently ignored.

enable

Set to 1 to enable alpha blending. Set to 0 to revert to color or chroma keying as specified through the DirectDraw API.

per_pixel

Set to 1 to enable per-pixel alpha blending. This value should only be 1 when inside a 24- or 32-bpp mode. Enabling per-pixel alpha blending automatically overrides the current mode settings and enables the graphics alpha channel. Setting this value to 1 inside a 16-bpp mode has no effect.

fixed_alpha

When per-pixel alpha blending is disabled, this value selects a constant alpha value to apply to the entire video overlay. This value can be in the range of 0x00 to 0xFF, with 0x00 indicating full video and 0xFF indicating full graphics.

Table 3-2. Alpha Window Registry Escapes

Escape Number	Input Size	Input Description	Output Size	Output Description
1027	20	Pointer to a LX_VIDEO_BLEND structure containing the alpha settings to program.	N/A	Ignored

3.3 Gamma

The Geode LX processor supports gamma correction for both the desktop graphics and video overlay window. A variety of driver escapes are provided to read and write the current gamma settings. These driver escapes are used by the display applet that ships with the driver.

3.3.1 Pixel Calculation

Gamma, brightness and contrast controls influence the pixel colors according to the following equations:

$$color_{out} = \left(contrast \cdot in^{\frac{1}{gamma}} + brightness \right) \cdot 255$$

$$in = \frac{color_{in}}{255}$$

$$0.5 \le gamma \le 3.5$$

$$-0.25 \le brightness \le 0.25$$

$$0.5 \le contrast \le 1.5$$

In the above equations, $color_{in}$ and $color_{out}$ refer to the individual red, green and blue 8-bit pixel components. Gamma correction is applied individually to each color before the colors are composited to form the final pixel value.

3.3.2 Gamma Escape Structure

Gamma information is communicated with the following structure:

```
typedef struct tagGamma
{
    unsigned long secret_word;
    unsigned long unused;
    unsigned long gamma_red;
    unsigned long gamma_green;
    unsigned long gamma_blue;
    unsigned long bright_red;
    unsigned long bright_green;
    unsigned long bright_blue;
    unsigned long contrast_red;
    unsigned long contrast_green;
    unsigned long contrast_blue;
} LX_GAMMA;
```

Table 3-3. LX_GAMMA Members

Member	Description
secret_word	Should be set to ABADDOODh before calling ExtEscape. This value helps to prevent spurious or accidental driver escapes.
unused	Unused DWORD.
gamma_red	Gamma value for the red color channel. This value must be in the range of 0 to 60. The actual gamma value used in the equation is calculated as 0.5 + (gamma_red * 0.05).
gamma_green	Gamma value for the green color channel. This value must be in the range of 0 to 60. The actual gamma value used in the equation is calculated as 0.5 + (gamma_green * 0.05).
gamma_blue	Gamma value for the blue color channel. This value must be in the range of 0 to 60. The actual gamma value used in the equation is calculated as 0.5 + (gamma_blue * 0.05).

Table 3-3. LX_GAMMA Members (Continued)

Member	Description
bright_red	Brightness value for the red color channel. This value must be in the range of 0 to 128. The actual brightness value used in the equation is calculated as (bright_red - 64) / 256.
bright_green	Brightness value for the green color channel. This value must be in the range of 0 to 128. The actual brightness value used in the equation is calculated as (bright_green - 64) / 256.
bright_blue	Brightness value for the blue color channel. This value must be in the range of 0 to 128. The actual brightness value used in the equation is calculated as (bright_blue - 64) / 256.
contrast_red	Contrast value for the red color channel. This value must be in the range of 0 to 200. The actual value used in the equation is calculated as 0.5 + (contrast_red * 0.005).
contrast_green	Contrast value for the green color channel. This value must be in the range of 0 to 200. The actual value used in the equation is calculated as 0.5 + (contrast_green * 0.005).
contrast_blue	Contrast value for the blue color channel. This value must be in the range of 0 to 200. The actual value used in the equation is calculated as 0.5 + (contrast_blue * 0.005).

3.3.3 Gamma Escapes

The following driver escapes are available to process gamma information.

Table 3-4. Gamma Escapes

Escape Number	Input Size	Input Description	Output Size	Output Description
1029	44	Pointer to a LX_GAMMA structure. Only the secret_word field is relevant.	44	Pointer to a LX_GAMMA structure to receive the gamma settings for the desktop.
1030	44	Pointer to a LX_GAMMA structure containing new gamma settings for the desktop. The settings are applied to the current display, but are not stored persistently in the registry.	44	Not used.
1031	44	Pointer to a LX_GAMMA structure. Only the secret_word field is relevant.	44	Pointer to a LX_GAMMA structure to receive the gamma settings for the video overlay.
1032	44	Pointer to a LX_GAMMA structure containing new gamma settings for the video overlay. The settings are applied to the current display, but are not stored persistently in the registry.	44	Not used.
1033	44	Pointer to a LX_GAMMA structure containing new gamma settings for the desktop. The settings are applied to the current display and are stored persistently in the registry.	44	Not used.
1034	44	Pointer to a LX_GAMMA structure containing new gamma settings for the video overlay. The settings are applied to the current display and are stored persistently in the registry.	44	Not used.

3.4 Previously Set Mode

When switching between CRT and TFT output formats, the Geode LX processor regenerates the list of available display modes. As such, the current display mode may not be a valid mode after the new output format is applied. A driver escape is provided to query the previously set mode for the current output format. This escape can be used to set the last valid mode.

3.4.1 Mode Structure

Mode information is communicated with the following structure:

```
typedef struct tagPreviousMode
{
    unsigned long secret_word;
    unsigned long unused;
    unsigned long mode_width;
    unsigned long mode_height;
    unsigned long mode_refresh;
    unsigned long mode_rotation;
} PREVIOUS_MODE;
```

Member	Description
secret_word	Should be set to ABADDOODh before calling ExtEscape. This value helps to prevent spurious or accidental driver escapes.
unused	Unused DWORD.
mode_width	Width of the previously set display mode.
mode_height	Height of the previously set display mode.
mode_refresh	Refresh rate of the previously set display mode.
mode_rotation	Rotation angle of the previously set display mode.

3.4.2 Escapes

The following driver escape is available to read the last set mode.

Table 3-5. Mode Escapes

Escape Number	Input Size	Input Description	Output Size	Output Description
1028	24	Pointer to a PREVIOUS_MODE structure. Only the secret_word field is relevant.	44	Pointer to a PREVIOUS_MODE structure to receive the previously set mode for the current output format.

4 Custom Panels

The most common customization applied to the Geode™ LX processor XP display driver is to adapt the driver to a custom TFT panel. The stock driver provides basic support for 60 Hz TFT panels at most standard resolutions. However, this support often falls short as users design systems with higher refresh-rate TFTs or widescreen panels.

The display driver now provides a simple interface to specify the details for any custom panel. A user need simply enter the timing and power sequence values for the TFT panel and the display driver automatically exports the native panel resolution to the operating system. This process can be accomplished with the standard driver binaries without access to the driver source code.

A user can also optionally expose hand-picked resolutions to be displayed in addition to the native panel mode. This is often used to allow scaling scenarios, where resolutions smaller or larger than the native panel size accomplish larger text or more desktop area, respectively.

4.1 Driver Behavior

When the driver is configured for a custom panel, the driver behavior alters slightly. Specifically, mode lists and output format toggling behave differently.

4.1.1 Mode Lists

The list of available modes are different between CRT and TFT output formats. When the driver is configured to enable display to the TFT, including simultaneous CRT/TFT display, the only modes presented to the user are the native panel mode and any optional resolutions specified in the customization file. All modes display using the panel timings, and are “fit” as necessary to the display using either scaling, centering or panning. In this operational mode, DDC communication with the monitor is disabled, such that the monitor type displays as “Default Monitor” in Device Manager.

When the driver is configured to display only to the CRT, the TFT is disabled. In this mode, all CRT modes and refresh rates are available. In this operational mode, DDC communication with the monitor is enabled, such that the monitor name should display as either “Plug and Play Monitor” or the actual model number of the attached CRT. Note that DDC communication is predicated on the DDCEnabled registry key being set to 1.

4.1.2 Output Format Toggling

As mentioned in Section 4.1.1, toggling the output format alters the list of available modes. As such, the display mode must generally be changed when the output format toggles between CRT and TFT. The display applet that ships with the driver uses the driver escapes described in Section 3.4 on page 13 to read and reprogram the last set mode for the new output format.

4.2 Procedure

The following procedure is used to customize the Geode LX processor XP display driver for a given TFT panel.

1) Create a text file that details the panel timings and power sequence. The syntax for the text file is specified in Section 4.3 on page 16. AMD provides a utility to assist in the auto-generation of this file. The utility, lx_gtf.exe, generates a TFT customization file based on typical TFT settings and CRT-compatible timings. It can be used to create an initial valid customization file that can later be tweaked by the end user. The timings for the file are calculated using the Generalized Timing Formula (GTF) published by VESA. Configurable parameters in the formula, such as margins, blanking duty cycle, etc., are hardcoded to the formula defaults.
2) Preprocess the text file using lxpanel.exe. A preprocessor step is used to avoid polluting the registry with large quantities of registry keys. It also serves to streamline the boot process and avoid errors from omitted or incorrect user settings.
3) The preprocessor generates an installation .inf as well as a binary file. These files should be used along with the standard driver binaries. No change to the binaries themselves is necessary.
4) Install the driver using the new .inf and reboot.

4.3 Panel File Syntax

This section details the syntax used in the panel customization text file.

4.3.1 Numerical Value Syntax

The ‘values’ referenced in the following sections refer to numerical values. In most instances, these values are specified as plain hexadecimal or decimal numbers, with a ‘0x’ prefix indicating hexadecimal. However, in some circumstances, numbers may be entered symbolically using predefined constants. This is necessary because some customization values are actually transparent links to API routines in the Cimarron HAL. It is thus simpler and safer to allow the user to enter values using the Cimarron API definitions.

The syntax for specifying a value using predefined constants is as follows:

(constant [| constant] ...)

The parentheses may be omitted if a single constant is used. For example,

(VG_MODEFLAG_NEG_HSYNC | VG_MODEFLAG_PANELOUT)

and

DF_OUTPUT_RGB

are valid uses of predefined constants. The available constants are specified inside their associated identifier descriptions.

4.3.2 Example Customization File

The explanation of the text file syntax is best accomplished by first showing an annotated example file. The sections inside this example file are explained and detailed in the following sections.

```
-------------------------------------------------------------------  ------------------
binary_file lx_qvga.out                                                Section 4.3.3 "File
inf_file lx_qvga.inf                                                   Header" on page 18
disk_id "AMD QVGA Custom Panel Example"
show_crt_modes 0
simultaneous_CRT 0
-------------------------------------------------------------------  ------------------
                                                                       Section 4.3.4 "Basic Mode
                                                                       Info" on page 19
flags                (VG_MODEFLAG_NEG_HSYNC |
                      VG_MODEFLAG_NEG_VSYNC |
                      VG_MODEFLAG_QVGA)
panel_width           320
panel_height          240
panel_tim1            0
panel_tim2            0
panel_dither_ctl      0
panel_pad_sel_low     0
panel_pad_sel_high    0
hactive               0x140
hblankstart           0x148
hsyncstart            0x162
hsyncend              0x180
hblankend             0x188
htotal                0x190
vactive               0x0F0
vblankstart           0x0F4
vsyncstart            0x0F9
vsyncend              0x0FD
vblankend             0x0FF
vtotal                0x104
frequency             0x1F3333

-------------------------------------------------------------------  ------------------
extra_width           1024                                             Section 4.3.5 "Extra Reso-
extra_height          768                                              lutions" on page 20
extra_width           800
extra_height          600
```

4.3.3 File Header

The first five lines of the text file serve to customize the driver installation. The syntax for each line in this section is:

identifier value

The identifier and value can be separated by tabs, spaces and carriage returns. The identifier/value pairs shown in Table 4-1 must be entered in order.

Table 4-1. Text File Header

Indentifier	Value Meaning	Example Value
binary_file	Filename for the binary output file. The filename should be a unique filename that does not conflict with any of the other driver binary filenames. It must not contain any spaces.	lx_adv7301.out
inf_file	Filename for the driver installation .inf. The filename should be a unique filename that does not conflict with any of the other driver binary filenames. It must not contain any spaces.	lx_adv7301.inf
disk_id	Installation string. This string is displayed when installing the driver. It is also displayed whenever the user browses to the display adapter in device manager. This value may contain spaces, but must be inside double quotes.	“AMD WinXP ADV7301 Example Installation”
show_crt_modes	If this value is non-zero, the display driver exposes all supported CRT modes to the operating system. The modes will be adapted to the panel timings when the TFT is active. The CRT timings are used, unmodified, when CRT-only output is selected. CRT-only output can be enabled/disabled with a driver escape. In general, users should set this value to 0 and use the extra_width and extra_height entries specified in Section 4.3.5 on page 20 as they provide greater control and flexibility.	0
simultaneous_CRT	Selects whether or not the CRT should be initially active at first boot. The TFT is always active when the driver first loads.	0

4.3.4 Basic Mode Info

This section details the display timings for the custom panel. The syntax for all lines in this section is:

identifier value

The identifiers and their expected values are listed in Table 4-2 on page 19. The identifiers must be entered in the order shown in the table.

Table 4-2. Display Timings

Identifier	Value Meaning	
flags	Mode Flags. Can be one of the following.	
	VG_MODEFLAG_NEG_HSYNC	This flag indicates that the hsync signal for a display mode is normally high and set low for the sync interval.
	VG_MODEFLAG_NEG_VSYNC	This flag indicates that the vsync signal for a display mode is normally high and set low for the sync interval.
	VG_MODEFLAG_LINEARPITCH	A mode is set by default using a stride that is a power of 2 to allow efficient compression. This flag indicates that the stride should equal the display line size to maximize memory efficiency.
	VG_MODEFLAG_QVGA	Indicates that the dot PLL is divided by 4 to accommodate a QVGA panel.
	VG_MODEFLAG_EXCLUDEPLL	Indicates that the PLL should not be changed during the mode set.
	VG_MODEFLAG_NOPANELTIMINGS	Indicates that the existing panel timings should not be changed when setting a TFT display mode.
	VG_MODEFLAG_XVGA_TFT	Indicates that the output TFT is a 2 pixel per clock XVGA panel.
	VG_MODEFLAG_CUSTOM_PANEL	Indicates that the user wants to program custom panel timings. If this flag is not set, default timings are programmed based on the other flags.
	VG_MODEFLAG_LOW_BAND	This flag configures the display controller arbitration settings for a low bandwidth mode. Only one of the bandwidth flags should be set for a given mode.
	VG_MODEFLAG_AVG_BAND	This flag configures the display controller arbitration settings for an average bandwidth mode.
	VG_MODEFLAG_HIGH_BAND	This flag configures the display controller arbitration settings for a high bandwidth mode.
	VG_MODEFLAG_INVERT_SHFCLK	Invert the shfclk out of LX processor to a TFT panel.
	VG_MODEFLAG_MANUAL_FREQUENCY	The *frequency* structure member contains a direct MSR setting and not a 16.16 fraction. This flag allows the user to directly specify a PLL setting without relying on Cimarron's internal mode tables.
	VG_MODEFLAG_PLL_BYPASS	The dot PLL is driven directly from the dot clock reference clock.
panel_width	Indicates the width of the panel.	
panel_height	Indicates the height of the panel.	
panel_tim1	Manual setting for the DF_PANEL_TIM1 register. This setting is only valid if the VG_MODEFLAG_CUSTOM_PANEL flag is also set and the VG_MODEFLAG_NOPANELTIMINGS is clear.	

Table 4-2. Display Timings

Identifier	Value Meaning
panel_tim2	Manual setting for the DF_PANEL_TIM2 register. This setting is only valid if the VG_MODEFLAG_CUSTOM_PANEL flag is also set and the VG_MODEFLAG_NOPANELTIMINGS is clear.
panel_dither_ctl	Manual setting for the DF_DITHER_CTL register. This setting is only valid if the VG_MODEFLAG_CUSTOM_PANEL flag is also set and the VG_MODEFLAG_NOPANELTIMINGS is clear.
panel_pad_sel_low	Manual setting for the DF_PAD_SEL MSR[31:0]. This setting is only valid if the VG_MODEFLAG_CUSTOM_PANEL flag is also set and the VG_MODEFLAG_NOPANELTIMINGS is clear
panel_pad_sel_high	Manual setting for the DF_PAD_SEL MSR[63:32]. This setting is only valid if the VG_MODEFLAG_CUSTOM_PANEL flag is also set and the VG_MODEFLAG_NOPANELTIMINGS is clear
hactive	Indicates the horizontal active timing value.
hblankstart	Indicates the horizontal blank start timing value.
hsyncstart	Indicates the horizontal sync start timing value.
hsyncend	Indicates the horizontal sync end timing value.
hblankend	Indicates the horizontal blank end timing value.
htotal	Indicates the horizontal total timing value.
vactive	Indicates the vertical active timing value.
vblankstart	Indicates the vertical blank start timing value.
vsyncstart	Indicates the vertical sync start timing value.
vsyncend	Indicates the vertical sync end timing value.
vblankend	Indicates the vertical blank end timing value.
vtotal	Indicates the vertical total timing value.
frequency	Indicates the pixel clock frequency of the display mode in 16.16 fixed point format. When the VG_MODEFLAG_MANUAL_FREQUENCY flag is set, this value represents the upper 32 bits of the GLCP_DOTPLL MSR, allowing the M, N and P values to be directly specified by the user. If the flag is not set, the Cimarron API picks the closest match that is found in its internal mode tables.

4.3.5 Extra Resolutions

This section specifies any additional resolutions that the user would like to include in addition to the native panel resolution. This section is optional and of arbitrary length. Additional resolutions are specified using the syntax:

extra_width number
extra_height number

4.4 Examples

This section details examples of extracting the display timings from TFT specifications.

4.4.1 Common Parameters

The Flags and Panel Registers file options wil be consistent for most panels. When generating settings for the first time, users should typically follow the rules for these sections.

4.4.1.1 Flags

The mode flags provide a great deal of flexibility that is not necessary for most users. In general, users will set the *flags* value to:

(VG_MODEFLAG_NEG_HSYNC | VG_MODEFLAG_NEG_VSYNC | VG_MODEFLAG_AVG_BAND)

Negative sync pulses are standard for most TFT panels and the third flag programs the display controller for average bandwidth needs.

4.4.1.2 Panel Registers

Panel_tim1, panel_tim2, panel_dither_ctl, panel_pad_sel_low, and panel_pad_sel_high specify values for panel registers in the Geode LX processor hardware. These settings cannot be extracted from the TFT specification and can be confusing to most users. In general, these values should be set to 0, as the display driver can and will program appropriate default values.

4.4.2 1440x900 Example

This section illustrates an example of how a user might customize the driver for a 1440x900 widescreen panel.

The most important and often difficult step in customizing the driver for a TFT is extracting the panel display timings from the TFT specification. Most TFT specifications do not list hactive, hblankstart, etc. Instead, those values must often be calculated based on terms like “front porch” and “TFT clock”. Table 4-3 is an example of a table that might be found in a TFT specification. Often, this table is the only description for the TFT display timings in the entire specification.

Table 4-3. Example TFT Specification Table

Signal	Item	Symbol	Min	Typ	Max	Unit	Note[1]
LVDS Clock	Frequency	F_C		44.5	56	MHz	
	Period	T_C	17.9	22.5		ns	
	High Time	T_{CH}		4/7		T_C	
	Low Time	T_{CL}		3/7		T_C	
LVDS Data	Setup Time	T_{LVS}	600			ps	
	Hold Time	T_{LVH}	600			ps	
Vertical Active Display Term	Frame Rate	F_R	56	60	75	Hz	$T_V = T_{VD} + T_{VB}$
	Total	T_V	905	926	1050	T_H	
	Display	T_{VD}	900	900	900	T_H	
	Blank	T_{VB}	$T_V - T_{VD}$	26	$T_V - T_{VD}$	T_H	
Horizontal Active Display Term	Total	T_H	750	800	960	T_C	$T_H = T_{HD} + T_{HB}$
	Display	T_{HD}	720	720	720	T_C	
	Blank	T_{HB}	$T_H - T_{HD}$	80	$T_H - T_{HD}$	T_C	

1. Because this module is operated by display enable only mode, hsync and vsync input signals should be set to low logic level or ground. Otherwise, this module would operate abnormally.

4.4.2.1 Horizontal Timings

This section details how to calculate the horizontal timing values: hactive, hblankstart, hsyncstart, hsyncend, hblankend, and htotal. Hactive should invariably be set to the width of the TFT panel. In the example in Table 4-3 on page 21, the TFT resolution is 1440x900, implying that hactive should be set to 1440. The remaining horizontal timings can be calculated from the “Horizontal Active Display Term” section in Table 4-3. Notice that the active display in the table is set to 720, which implies that the TFT uses two pixels per clock. This implies that we must multiply all horizontal values in the table by 2 before we perform any calculations.

Hblankstart refers to the beginning of the blanking interval. This value should almost invariably be set equal to hactive. This implies that the blanking interval starts immediately after the active interval.

Hsyncstart and hysncend refer to the beginning and end of the sync pulse. The sync pulse is used on CRTs when aligning the beam to the beginning of the next line. In general, the time between the beginning of the blanking interval and the beginning of the sync pulse is referred to as the “front porch”. The time between the end of the sync pulse and the end of the blanking interval is referred to as the “back porch”. For the TFT in this example, the sync pulses are listed as irrelevant. To prevent incompatibility with CRTs, the sync pulse is programmed to a non-zero pulse width inside the blanking interval.

Hblankend and htotal should almost invariably be set to the total number of clocks for the line. In the above example, the total for the line is listed as 800. From earlier observations we know to multiply that number by 2, and so the horizontal total for this TFT is 1600.

4.4.2.2 Vertical Timings

This section details how to calculate the vertical timing values: vactive, vblankstart, vsyncstart, vsyncend, vblankend, and vtotal. Vactive should invariably be set to the height of the TFT, or 900 in this example. The remaining timings can be calculated from the “Vertical Active Display Term” in Table 4-3 on page 21. As with the horizontal timings, vblankstart should be set to equal vactive, and vblankend should be set to equal vtotal.

The vertical total for this TFT is listed as 926. As the sync pulses are irrelevant, a logical sync position is assigned inside the blanking interval.

4.4.2.3 Frequency

The pixel clock for the TFT refers to the rate at which the display timings are clocked. In general, the refresh rate for a display mode can be calculated as frequency / (htotal * vtotal).

In Table 4-3 on page 21, the frequency is listed as 44.5 MHz. Knowing that this frequency is based on two pixels per clock and that the horizontal values have already been expanded, the frequency value must be multiplied by 2, yielding 89 MHz. This value can be specified in 16.16 format as 0x590000. This will request 89 MHz as the pixel clock frequency, although the driver will program the closest match from its internal mode tables.

4.4.2.4 Additional Resolutions

Since this is a widescreen panel, CRT resolutions will be at the wrong aspect ratio. Including some additional 16:9 modes, such as 1024x600, 800x480, or 1280x768 may be needed.

4.4.2.5 Sanity Check

Checking the calculated timings is recommended to validate that pix_clk / (htotal * vtotal) is close to the desired refresh rate. In our example, 89 MHz / (1600*926) = 60.07. As the typical refresh rate for TFTs is 60 Hz, this would suggest our timings are acceptable.

4.4.2.6 Putting It All Together

The important information has now been calculated and the final customization file can be created as follows:

```
binary_file example_tft.out
inf_file lx_example_tft.inf
disk_id "AMD 1440x900 Example"
show_crt_modes 0
simultaneous_CRT 1
```

flags	(VG_MODEFLAG_NEG_HSYNC | VG_MODEFLAG_NEG_VSYNC | VG_MODEFLAG_AVG_BAND)	Section 4.4.1.1 "Flags" on page 21
panel_width	1440	Section 4.4.1.2 "Panel Registers" on page 21
panel_height	900	
panel_tim1	0	
panel_tim2	0	
panel_dither_ctl	0	
panel_pad_sel_low	0	
panel_pad_sel_high	0	
hactive	1440	Section 4.4.2.1 "Horizontal Timings" on page 22
hblankstart	1440	
hsyncstart	1472	
hsyncend	1568	
hblankend	1600	
htotal	1600	
vactive	900	Section 4.4.2.2 "Vertical Timings" on page 22
vblankstart	900	
vsyncstart	908	
vsyncend	911	
vblankend	926	
vtotal	926	
frequency	0x590000	Section 4.4.2.3 "Frequency" on page 22
extra_width	800	Section 4.4.2.4 "Additional Resolutions" on page 22
extra_height	480	
extra_width	1024	
extra_height	600	
extra_width	1280	
extra_height	768	

4.4.3 1280x768 Example

This section provides an example with a 1280x768 widescreen panel.

Table 4-4. Example TFT Specification Table

Item	Symbol	Min	Typ	Max	Unit	Notes
DCLK	Frequency / f_{CLK}		65	70	MHz	
	Period / t_{CLK}	14.3	15.4		ns	
hsync	Period / t_{HP}	1312	1320		t_{CLK}	
	Width-Active / t_{WH}	8	8			
vsync	Frequency / f_{VSY}	59	60	61	Hz	
	Period / t_{VP}	772	812		t_{HP}	
	Width-Active / t_{WV}	1	6			
Data Enable	Horizontal back porch / t_{HBP}	8	16		t_{CLK}	for DCLK
	Horizontal front porch / t_{HFP}	16	16			
	Horizontal active data / t_{HA}	1280	1280	1280		
	Vertical back porch / t_{VBP}	2	35		t_{HP}	
	Vertical front porch / t_{VFP}	1	3			
	Vetical active data / t_{VA}	768	768	768		

4.4.3.1 Horizontal Timings

The rules for this TFT are similar to the rules in the example in Section 4.4.2 on page 21. Hactive and hblankstart should be set to the number of active pixels in the panel, 1280. Hsyncstart is calculated by adding the front porch value to the number of active pixels. In this example, the horizontal front porch is listed as 16 clocks, which equals 16 pixels. Hsyncstart is thus 1296. Hsyncend is calculated in one of two ways. It can be either calculated by subtracting the back porch value from the total number of horizontal pixel clocks, or it can be calculated by adding the sync width to hsyncstart. n this example, the sync width is 8, which yields a hsyncend value of 1304. To double check, the total number of pixels per line is listed as 1320 and the back porch is listed as 16. 1320-1316 = 1304. Finally, hblankend and htotal should be set to the total number of pixel clocks per line. In Table 4-4 on page 24, the number of pixel clocks per line is referred to as the “hsync period”. So, hblankend and htotal should be 1320.

4.4.3.2 Vertical Timings

First, vactive and vblankstart should be set to the number of lines in the TFT, or 768. Vsyncstart is calculated by adding the vertical front porch value to vblankstart. This yields a value of 771. Vsyncend can be calculated similar to hsyncend, either by adding the vertical sync width to vsyncstart or by subtracting the vertical back porch from the total number of lines. In Table 4-4 on page 24, both methods yield a value of 777. Vtotal and vblankend are set to the total number of lines. In Table 4-4, the total number of lines is referred to as the “vsync period”. So, these values should be 812.

4.4.3.3 Frequency

DCLK is listed as 65 MHz, which would be written in 16.16 format as 0x410000.

4.4.3.4 Additional Resolutions

Since this is a widescreen panel, CRT resolutions will be at the wrong aspect ratio. Including some additional 16:9 modes, such as 1024x600 or 800x480 may be needed.

4.4.3.5 Sanity Check

65 MHz / (1320*812) = 60.6 Hz, which is reasonably close to the targeted 60 Hz.

4.4.3.6 Putting It All Together

The important information has now been calculated and the final customization file can be created as follows:

```
binary_file example_tft2.out
inf_file lx_example_tft2.inf
disk_id "AMD 1280x768 Example"
show_crt_modes 0
simultaneous_CRT 1
```

flags	(VG_MODEFLAG_NEG_HSYNC | VG_MODEFLAG_NEG_VSYNC | VG_MODEFLAG_AVG_BAND)	Section 4.4.1.1 "Flags" on page 21
panel_width	1280	Section 4.4.1.2 "Panel Registers" on page 21
panel_height	768	
panel_tim1	0	
panel_tim2	0	
panel_dither_ctl	0	
panel_pad_sel_low	0	
panel_pad_sel_high	0	
hactive	1280	Section 4.4.3.1 "Horizontal Timings" on page 24
hblankstart	1280	
hsyncstart	1296	
hsyncend	1304	
hblankend	1320	
htotal	1320	
vactive	768	Section 4.4.3.2 "Vertical Timings" on page 24
vblankstart	768	
vsyncstart	771	
vsyncend	777	
vblankend	812	
vtotal	812	
frequency	0x410000	Section 4.4.3.3 "Frequency" on page 24
extra_width	800	Section 4.4.3.4 "Additional Resolutions" on page 24
extra_height	480	
extra_width	1024	
extra_height	600	

5 Advanced Mode Customization

In advanced scenarios, a user may need complete control over all aspects of the display modes that are presented to the user and the operating system. To accomplish this, the Geode™ LX processor XP display driver supports the creation and use of a mode customization file. This file contains all details for one or more display modes, as well as two optional sequences of SMBus writes to program external devices, such as TV encoders. The following aspects of each display mode can be configured. Some of these settings are optional.

- Bits Per Pixel (bpp)
- Output Path (TFT, VOP, DRGB, CRT, etc.)
- Output Color Space (RGB, YUV, HDTV YUV, etc.)
- Border/Overscan color
- Source dimensions (width and height)
- Complete mode timings, including interlaced TV timings
- Panel timings and power sequence values
- Pixel clock frequency
- Flicker filter strength
- VOP bus size and format
- Graphics scaler filter coefficients
- SMBus writes before and after the mode is set

5.1 Procedure

The following procedure is used to customize the Geode LX processor XP display driver for advanced modes.

1) Create a text file that details all aspects of the custom modes. The syntax for the text file is specified in Section 5.2 on page 28.
2) Preprocess the text file using lxcustom.exe. A preprocessor step is used to avoid polluting the registry with large quantities of registry keys. It also serves to streamline the boot process and avoid errors from omitted or incorrect user settings.
3) The preprocessor generates an installation .inf as well as a binary file. These files should be used along with the standard driver binaries. No change to the binaries themselves is necessary.
4) Install the driver using the new .inf and reboot.

5.2 Customization File Syntax

This section details the syntax used in the customization text file.

5.2.1 Numerical Value Syntax

The ‘values’ referenced in the following sections refer to numerical values. In most instances, these values are specified as plain hexadecimal or decimal numbers, with a ‘0x’ prefix indicating hexadecimal. However, in some circumstances, numbers may be entered symbolically using predefined constants. This is necessary because some customization values are actually transparent links to API routines in the Cimarron HAL. It is thus simpler and safer to allow the user to enter values using the Cimarron API definitions.

The syntax for specifying a value using predefined constants is:

(constant [| constant] ...)

The parentheses may be omitted if a single constant is used. For example,

(VG_MODEFLAG_NEG_HSYNC | VG_MODEFLAG_PANELOUT)

and

DF_OUTPUT_RGB

are valid uses of predefined constants. The available constants are specified inside their associated identifier descriptions.

5.2.2 Example Customization File

The explanation of the text file syntax is best accomplished by first showing an annotated example file. The sections inside this example file are explained and detailed in the following sections.

```
-------------------------------------------------------------------------- ----------------
                                                                             "File Header" on page
                                                                                   32
binary_file lx_adv7301.out
inf_file lx_adv7301.inf
disk_id "AMD Win XP ADV7301 Example Installation"
manual_display_mode_count 1
-------------------------------------------------------------------------- ----------------
                                                                             "Basic Mode Informa-
                                                                             tion" on page 32
manual_mode_0 bpp_mask                    0x6
manual_mode_0 output_path                 0
manual_mode_0 output_space                DF_OUTPUT_SDTV
manual_mode_0 flicker_strength               VG_FLICKER_FILTER_1_4
manual_mode_0 line_double_video_on_interlaced  0
manual_mode_0 border_color                0
manual_mode_0 premode_smbus_count         8
manual_mode_0 postmode_smbus_count        8
manual_mode_0 custom_coefficients         1
manual_mode_0 configure_vop               1
```

```
------------------------------------------------------------------------ ----------------
                                                                          "Mode Timings" on
                                                                              page 34
manual_mode_0\manual_timings flags   (VG_MODEFLAG_INTERLACED |
                                      VG_MODEFLAG_TVOUT      |
                                      VG_MODEFLAG_HALFCLOCK  |
                                      VG_MODEFLAG_INT_FLICKER)
manual_mode_0\manual_timings src_width              720
manual_mode_0\manual_timings src_height             480
manual_mode_0\manual_timings mode_width             0
manual_mode_0\manual_timings mode_height            0
manual_mode_0\manual_timings panel_width            0
manual_mode_0\manual_timings panel_height           0
manual_mode_0\manual_timings panel_tim1             0
manual_mode_0\manual_timings panel_tim2             0
manual_mode_0\manual_timings panel_dither_ctl       0
manual_mode_0\manual_timings panel_pad_sel_low      0
manual_mode_0\manual_timings panel_pad_sel_high     0
manual_mode_0\manual_timings hactive                0x2D0
manual_mode_0\manual_timings hblankstart            0x2D0
manual_mode_0\manual_timings hsyncstart             0x2E1
manual_mode_0\manual_timings hsyncend               0x320
manual_mode_0\manual_timings hblankend              0x35A
manual_mode_0\manual_timings htotal                 0x35A
manual_mode_0\manual_timings vactive                0xF0
manual_mode_0\manual_timings vblankstart            0xF0
manual_mode_0\manual_timings vsyncstart             0xF4
manual_mode_0\manual_timings vsyncend               0xF8
manual_mode_0\manual_timings vblankend              0x107
manual_mode_0\manual_timings vtotal                 0x107
manual_mode_0\manual_timings vactive_even           0xF0
manual_mode_0\manual_timings vblankstart_even       0xF0
manual_mode_0\manual_timings vsyncstart_even        0xF4
manual_mode_0\manual_timings vsyncend_even          0xF8
manual_mode_0\manual_timings vblankend_even         0x106
manual_mode_0\manual_timings vtotal_even            0x106
manual_mode_0\manual_timings frequency              0x1B0000

------------------------------------------------------------------------ ----------------
                                                                          "Basic VOP Settings"
                                                                              on page 39
manual_mode_0\vop_settings flags            0
manual_mode_0\vop_settings mode             VOP_MODE_601
manual_mode_0\vop_settings conversion_mode  VOP_422MODE_COSITED
manual_mode_0\vop_settings vsync_out     VOP_MB_SYNCSEL_DISABLED

------------------------------------------------------------------------ ----------------
                                                                          "601 VOP Settings"
                                                                              on page 40
manual_mode_0\vop_settings\vop601 flags             0
manual_mode_0\vop_settings\vop601 vsync_shift       0
manual_mode_0\vop_settings\vop601 vsync_shift_count 0
manual_mode_0\vop_settings\vop601 output_mode       0
```

```
------------------------------------------------------------------------ ----------------
                                                                          "Horizontal Coeffi-
                                                                          cients" on page 40
manual_mode_0\custom_hcoeff hcoeff_0_0              34
manual_mode_0\custom_hcoeff hcoeff_0_1              223
manual_mode_0\custom_hcoeff hcoeff_0_2              221
manual_mode_0\custom_hcoeff hcoeff_0_3              34
manual_mode_0\custom_hcoeff hcoeff_0_4              0
manual_mode_0\custom_hcoeff hcoeff_1_0              34
manual_mode_0\custom_hcoeff hcoeff_1_1              219
manual_mode_0\custom_hcoeff hcoeff_1_2              224
manual_mode_0\custom_hcoeff hcoeff_1_3              35
manual_mode_0\custom_hcoeff hcoeff_1_4              0
.
.
.
manual_mode_0\custom_hcoeff hcoeff_80_0             12
manual_mode_0\custom_hcoeff hcoeff_80_1             154
manual_mode_0\custom_hcoeff hcoeff_80_2             267
manual_mode_0\custom_hcoeff hcoeff_80_3             77
manual_mode_0\custom_hcoeff hcoeff_80_4             2
.
.
.
manual_mode_0\custom_hcoeff hcoeff_255_0            0
manual_mode_0\custom_hcoeff hcoeff_255_1            35
manual_mode_0\custom_hcoeff hcoeff_255_2            224
manual_mode_0\custom_hcoeff hcoeff_255_3            219
manual_mode_0\custom_hcoeff hcoeff_255_4            34

------------------------------------------------------------------------ ----------------
                                                                          "Vertical Coefficients"
                                                                          on page 41
manual_mode_0\custom_vcoeff vcoeff_0_0              257
manual_mode_0\custom_vcoeff vcoeff_0_1              255
manual_mode_0\custom_vcoeff vcoeff_0_2              0
manual_mode_0\custom_vcoeff vcoeff_1_0              253
manual_mode_0\custom_vcoeff vcoeff_1_1              259
manual_mode_0\custom_vcoeff vcoeff_1_2              0
.
.
.
manual_mode_0\custom_vcoeff vcoeff_80_0             86
manual_mode_0\custom_vcoeff vcoeff_80_1             416
manual_mode_0\custom_vcoeff vcoeff_80_2             10
.
.
.
manual_mode_0\custom_vcoeff vcoeff_255_0            1
manual_mode_0\custom_vcoeff vcoeff_255_1            259
manual_mode_0\custom_vcoeff vcoeff_255_2            252
```

```
------------------------------------------------------------------------ ----------------
                                                                          "Pre-mode SMBus
                                                                          Writes" on page 41
manual_mode_0\premode_smbus base_0                  0x54
manual_mode_0\premode_smbus addr_0                  0x01
manual_mode_0\premode_smbus data_0                  0x80
manual_mode_0\premode_smbus base_1                  0x54
manual_mode_0\premode_smbus addr_1                  0x40
manual_mode_0\premode_smbus data_1                  0x30
manual_mode_0\premode_smbus base_2                  0x54
manual_mode_0\premode_smbus addr_2                  0x42
manual_mode_0\premode_smbus data_2                  0x49
manual_mode_0\premode_smbus base_3                  0x54
manual_mode_0\premode_smbus addr_3                  0x44
manual_mode_0\premode_smbus data_3                  0x00
manual_mode_0\premode_smbus base_4                  0x54
manual_mode_0\premode_smbus addr_4                  0x4C
manual_mode_0\premode_smbus data_4                  0x16
manual_mode_0\premode_smbus base_5                  0x54
manual_mode_0\premode_smbus addr_5                  0x4D
manual_mode_0\premode_smbus data_5                  0x7C
manual_mode_0\premode_smbus base_6                  0x54
manual_mode_0\premode_smbus addr_6                  0x4E
manual_mode_0\premode_smbus data_6                  0xF0
manual_mode_0\premode_smbus base_7                  0x54
manual_mode_0\premode_smbus addr_7                  0x4F
manual_mode_0\premode_smbus data_7                  0x21

------------------------------------------------------------------------ ----------------
                                                                          "Post-mode SMBus
                                                                          Writes" on page 41
manual_mode_0\postmode_smbus base_0                 0x54
manual_mode_0\postmode_smbus addr_0                 0x01
manual_mode_0\postmode_smbus data_0                 0x80
manual_mode_0\postmode_smbus base_1                 0x54
manual_mode_0\postmode_smbus addr_1                 0x40
manual_mode_0\postmode_smbus data_1                 0x30
manual_mode_0\postmode_smbus base_2                 0x54
manual_mode_0\postmode_smbus addr_2                 0x42
manual_mode_0\postmode_smbus data_2                 0x49
manual_mode_0\postmode_smbus base_3                 0x54
manual_mode_0\postmode_smbus addr_3                 0x44
manual_mode_0\postmode_smbus data_3                 0x00
manual_mode_0\postmode_smbus base_4                 0x54
manual_mode_0\postmode_smbus addr_4                 0x4C
manual_mode_0\postmode_smbus data_4                 0x16
manual_mode_0\postmode_smbus base_5                 0x54
manual_mode_0\postmode_smbus addr_5                 0x4D
manual_mode_0\postmode_smbus data_5                 0x7C
manual_mode_0\postmode_smbus base_6                 0x54
manual_mode_0\postmode_smbus addr_6                 0x4E
manual_mode_0\postmode_smbus data_6                 0xF0
manual_mode_0\postmode_smbus base_7                 0x54
manual_mode_0\postmode_smbus addr_7                 0x4F
manual_mode_0\postmode_smbus data_7                 0x21
```

5.2.3 File Header

The first four lines of the text file serve to customize the driver installation as well as to specify the number of custom modes to follow. The file header is the only section that does not need to be repeated for each display mode.

The syntax for each line in this section is:

identifier value

The identifier and value can be separated by tabs, spaces and carriage returns. The four identifier/value pairs, shown in Table 5-1, must be entered in order.

Table 5-1. Text File Header

Indentifier	Value Meaning	Example Value
binary_file	Filename for the binary output file. The filename should be a unique filename that does not conflict with any other driver binary filenames. It must not contain any spaces.	lx_adv7301.out
inf_file	Filename for the driver installation .inf. The filename should be a unique filename that does not conflict with any other driver binary filenames. It must not contain any spaces.	lx_adv7301.inf
disk_id	Installation string. This string is displayed when installing the driver. It is also displayed whenever the user browses to the display adapter in device manager. This value may contain spaces, but must be inside double quotes.	“AMD WinXP ADV7301 Example Installation”
manual_display_mode_count	Mode count. This value indicates the number of custom modes to follow in the text file.	1

5.2.4 Basic Mode Information

This section details basic options for each display mode. Each display mode in the text file must include this section.

The syntax for all lines in this section is:

manual_mode_xx identifier value

‘xx’ in the above syntax refers to the current mode number, starting from 0. The identifiers and their expected values are listed in Table 5-2 on page 33. The identifiers must be entered in the order shown in the table.

Table 5-2. Display Modes

Indentifiers	Value Meaning
bpp_mask	This mask indicates which pixel depths should be included for this mode. **Value** — **Description** 0x01 — Include 8 bpp 0x02 — Include 16 bpp 0x04 — Include 32 bpp (no alpha) 0x08 — Include 32 bpp (with per-pixel alpha) A mask should not include both 32-bpp modes, as there is no way to distinguish between the two settings when presenting the modes to the operating system. Note that per-pixel alpha blending can be enabled dynamically with a driver escape.
output_path	This value specifies the output path for display data. This value corresponds to the parameter passed to the Cimarron API routine *df_set_output_path.* For more information see the Cimarron API specification. **Value** — **Description** 0x00 — Output path is implicitly specified in the mode timings DF_DISPLAY_CRT — CRT Output DF_DISPLAY_FP — TFT Output DF_DISPLAY_CRT_FP — Simultaneous CRT/TFT DF_DISPLAY_VOP — VOP DF_DISPLAY_DRGB — DRGB DF_DISPLAY_CRT_DRGB — Simultaneous CRT/DRGB
output_space	This value specifies the output color space. This value corresponds to the parameter passed to the Cimarron API routine *df_set_output_color_space.* For more information, see the Cimarron API specification. **Value** — **Description** 0x00 — Color Space is implicitly specified in the mode timings (Currently RGB only). DF_OUTPUT_RGB — RGB DF_OUTPUT_ARGB — ARGB DF_OUTPUT_SDTV — SDTV YUV DF_OUTPUT_HDTV — HDTV YUV
flicker_strength	This value is only applicable when the mode timings select a flicker-filtered interlaced mode. However, for simplicity, it must always be specified. **Value** — **Description** VG_FLICKER_FILTER_NONE — No flicker filter. VG_FLICKER_FILTER_1_16 — 1/16, 7/8, 1/16 (light) VG_FLICKER_FILTER_1_8 — 1/8, 3/4, 1/8 (medium) VG_FLICKER_FILTER_1_4 — 1/4, 1/2, 1/4 (strong) VG_FLICKER_FILTER_5_16 — 5/16, 3/8, 5/16 (smudge)
line_double_video_on_interlaced	This value is only applicable when interlaced output is selected. The standard desktop rendering in Windows® XP is targeted for a progressive display. Desktop images may thus flicker when viewed on an interlaced display, such as a TV. The flicker filter is used to prevent flicker for normal desktop images. However, video playback bypasses the flicker filter and so may produce flicker regardless of flicker filter settings. When set to 1, this option enables video 'line doubling', which reduces the visible flicker on video images at the expense of quality.
border_color	This value sets the 24-bit RGB color for the border or overscan. This color is used when centering a mode inside a panel or when adding overscan for TV output.
premode_smbus_count	This value indicates the number of base/address/data SMBus triplets that must be written before a modeset.

Table 5-2. Display Modes

Indentifiers	Value Meaning
postmode_smbus_count	This value indicates the number of base/address/data SMBus triplets that must be written after a modeset.
custom_coefficients	This value is only used when graphics scaling is enabled. However, for simplicity, it must always be specified. When set to 1, it indicates that the user is specifying custom coefficients for the graphics scaler filter. When set to 0, the default coefficients are used.
configure_vop	This value is used to indicate that the user would like to program the VOP configuration for the mode.

5.2.5 Mode Timings

This section specifies the timing details for each display mode. It must be included for each mode.

The syntax for all lines in this section is:

manual_mode_xx\manual_timings identifier value

'xx' in the above syntax refers to the current mode number, starting from 0. The identifiers and their expected values are listed in Table 5-4 on page 38. Each of these identifiers correspond to entries in the VG_DISPLAY_MODE structure passed to the *vg_set_custom_mode* routine in the Cimarron API.

5.2.5.1 VG_DISPLAY_MODE

The VG_DISPLAY_MODE structure definition from the Cimarron API is included here for convenience.

```
typedef struct tagVGSimParams {
        unsigned long internal_flags;
        unsigned long flags;
        unsigned long src_width;
        unsigned long src_height;
        unsigned long mode_width;
        unsigned long mode_height;
        unsigned long panel_width;
        unsigned long panel_height;
        unsigned long panel_tim1;
        unsigned long panel_tim2;
        unsigned long panel_dither_ctl;
        unsigned long panel_pad_sel_low;
        unsigned long panel_pad_sel_high;
        unsigned long hactive;
        unsigned long hblankstart;
        unsigned long hsyncstart;
        unsigned long hsyncend;
        unsigned long hblankend;
        unsigned long htotal;
        unsigned long vactive;
        unsigned long vblankstart;
        unsigned long vsyncstart;
        unsigned long vsyncend;
        unsigned long vblankend;
        unsigned long vtotal;
        unsigned long vactive_even;
        unsigned long vblankstart_even;
        unsigned long vsyncstart_even;
        unsigned long vsyncend_even;
        unsigned long vblankend_even;
        unsigned long vtotal_even;
        unsigned long frequency;
} VG_DISPLAY_MODE;
```

Table 5-3. VG_DISPLAY_MODE Members

Member	Description
internal_flags	Flags used in Cimarron's internal mode tables. These flags can be read and interpreted by the user when searching through the table. They are ignored when calling vg_set_custom_mode.
flags	A set of flags that indicate more detailed information about the display mode. The supported flags are:

Flag	Description
VG_MODEFLAG_NEG_HSYNC	This flag indicates that the hsync signal for a display mode is normally high and set low for the sync interval.
VG_MODEFLAG_NEG_VSYNC	This flag indicates that the vsync signal for a display mode is normally high and set low for the sync interval.
VG_MODEFLAG_INTERLACED	This flag indicates that a display mode is interlaced.
VG_MODEFLAG_PANELOUT	Indicates that the display mode is being set on a panel.
VG_MODEFLAG_CENTERED	When the VG_MODEFLAG_PANEL flag is set and the dimensions of the display mode are smaller than the dimensions of the panel, this flag enables panel centering. Otherwise, the display mode will be displayed in the upper left corner of the panel.
VG_MODEFLAG_LINEARPITCH	A mode is set by default using a stride that is a power of 2 to allow efficient compression. This flag indicates that the stride should equal the display line size to maximize memory efficiency.
VG_MODEFLAG_TVOUT	This flag indicates that the mode will be output to the VOP logic.
VG_MODEFLAG_HALFCLOCK	Indicates that the dot clock is divided by 2. This is usually to accomodate 8-bit VOP.
VG_MODEFLAG_QVGA	Indicates that the dot PLL is divided by 4 to accomodate a QVGA panel.
VG_MODEFLAG_EXCLUDEPLL	Indicates that the PLL should not be changed during the mode set.
VG_MODEFLAG_NOPANELTIMINGS	Indicates that the existing panel timings should not be changed when setting a TFT display mode. This setting is only valid when the VG_MODEFLAG_PANEL flag is also set.
VG_MODEFLAG_XVGA_TFT	Indicates that the output TFT is a 2 pixel per clock XVGA panel. This setting is only valid when the VG_MODEFLAG_PANEL flag is also set.
VG_MODEFLAG_CUSTOM_PANEL	Indicates that the user wants to program custom panel timings. If this flag is not set, default timings are programmed based on the other flags.
VG_MODEFLAG_CRT_AND_FP	This flag is used to set simultaneous CRT and TFT output. It is only valid if the VG_MODEFLAG_PANEL is also set.
VG_MODEFLAG_LOW_BAND	This flag configures the display controller arbitration settings for a low bandwidth mode. When bandwidth is not tight, such as in small display modes or with very fast bus speeds, this setting forces the display controller to be less intrusive on the bus, possibly boosting performance. Only one of the bandwidth flags should be set for a given mode.
VG_MODEFLAG_AVG_BAND	This flag configures the display controller arbitration settings for an average bandwidth mode. The display controller is configured such that it does not starve, but also that it does not stay in high priority for long periods of time. Only one of the bandwidth flags should be set for a given mode.
VG_MODEFLAG_HIGH_BAND	This flag configures the display controller arbitration settings for a high bandwidth mode. This is useful when bandwidth is tight due to lower bus speeds or higher system traffic. The display controller stays in high priority most of the time, guaranteeing lower latencies for memory accesses. However, this setting may degrade CPU performance. Only one of the bandwidth flags should be set for a given mode.

Table 5-3. VG_DISPLAY_MODE Members

Member	Description	
flags (continued)	**Flag**	**Description**
	VG_MODEFLAG_LEGACY_BAND	This flag configures the display controller arbitration settings to behave exactly as they did in the AMD Geode™ GX processor. A single set of FIFO watermarks is used to arbitrate between high and low priority on the GeodeLink™ bus. Only one of the bandwidth flags should be set for a given mode.
	VG_MODEFLAG_INT_ADDRESS	When interlacing is enabled via VG_MODEFLAG_INTERLACED, this flag configures the mode for interlaced addressing. When interlaced addressing is enabled, scaling and flicker filtering are disabled and the display pitch is set to skip every other line. This is the default interlacing method for all HDTV modes in Cimarron's internal tables. Only one interlacing flag should be specified for any given mode.
	VG_MODEFLAG_INT_LINEDOUBLE	When interlacing is enabled, this flag configures the mode for line doubling. A line-doubled mode is configured such that the odd and even fields fetch the same data. The height of the mode is set according to the equation: height = max(odd_height, even_height) Scaling can be enabled in this mode provided that the source width is less than or equal to 1024 pixels. Only one interlacing flag should be set for any given mode.
	VG_MODEFLAG_INT_FLICKER	When interlacing is enabled, this flag indicates that the mode should be properly interlaced and flicker filtered. This is the default interlacing method for all SDTV modes in Cimarron's internal tables. The flicker filter settings can be configured by the routine *vg_configure_flicker_filter*. Flicker filtered modes can be scaled, but the source and destination widths must be less than or equal to 1024 pixels. Only one interlacing flag should be set for any given mode.
	VG_MODEFLAG_INVERT_SHFCLK	Invert the shfclk out of the LX processor to a TFT panel.
	VG_MODEFLAG_MANUAL_FREQUENCY	The *frequency* structure member contains a direct MSR setting and not a 16.16 fraction. This flag allows the user to directly specify a PLL setting without relying on Cimarron's internal mode tables.
	VG_MODEFLAG_PLL_BYPASS	The dot PLL is driven directly from the dot clock reference clock.
	VG_MODEFLAG_VIP_TO_DOT_CLOCK	The dot PLL is driven directly from the VIP input clock.
src_width	Indicates the source width of the display mode when setting a mode. If this parameter differs from the active width of the mode, horizontal scaling and filtering is performed.	
src_height	Indicates the source height of the display mode when setting a mode. If this parameter differs from the active height of the mode, vertical scaling and filtering is performed.	
mode_width	Indicates the width of the display mode when setting a mode on a panel. This parameter is generally used to enable a virtual desktop, wherein the dimensions of the mode exceed the resolution of the panel.	
mode_height	Indicates the height of the display mode when setting a mode on a panel. This parameter is generally used to enable a virtual desktop, wherein the dimensions of the mode exceed the resolution of the panel.	
panel_width	Indicates the width of the panel when setting a panel mode. This is used to dynamically calculate centered timings and to configure panning for virtual desktops.	
panel_height	Indicates the height of the panel when setting a panel mode. This is used to dynamically calculate centered timings and to configure panning for virtual desktops.	
panel_tim1	Manual setting for the DF_PANEL_TIM1 register. This setting is only valid if the VG_MODEFLAG_CUSTOM_PANEL flag is also set and the VG_MODEFLAG_NOPANELTIMINGS is clear.	

Table 5-3. VG_DISPLAY_MODE Members

Member	Description
panel_tim2	Manual setting for the DF_PANEL_TIM2 register. This setting is only valid if the VG_MODEFLAG_CUSTOM_PANEL flag is also set and the VG_MODEFLAG_NOPANELTIMINGS is clear.
anel_dither_ctl	Manual setting for the DF_DITHER_CTL register. This setting is only valid if the VG_MODEFLAG_CUSTOM_PANEL flag is also set and the VG_MODEFLAG_NOPANELTIMINGS is clear.
panel_pad_sel_low	Manual setting for the DF_PAD_SEL MSR[31:0]. This setting is only valid if the VG_MODEFLAG_CUSTOM_PANEL flag is also set and the VG_MODEFLAG_NOPANELTIMINGS is clear.
panel_pad_sel_high	Manual setting for the DF_PAD_SEL MSR[63:32]. This setting is only valid if the VG_MODEFLAG_CUSTOM_PANEL flag is also set and the VG_MODEFLAG_NOPANELTIMINGS is clear.
hactive	Indicates the horizontal active timing value.
hblankstart	Indicates the horizontal blank start timing value.
hsyncstart	Indicates the horizontal sync start timing value.
hsyncend	Indicates the horizontal sync end timing value.
hblankend	Indicates the horizontal blank end timing value.
htotal	Indicates the horizontal total timing value.
vactive	Indicates the vertical active timing value.
vblankstart	Indicates the vertical blank start timing value.
vsyncstart	Indicates the vertical sync start timing value.
vsyncend	Indicates the vertical sync end timing value.
vblankend	Indicates the vertical blank end timing value.
vtotal	Indicates the vertical total timing value.
vactive_even	Indicates the vertical active timing value for the even field when in an interlaced mode.
vblankstart_even	Indicates the vertical blank start timing value for the even field when in an interlaced mode.
vsyncstart_even	Indicates the vertical sync start timing value for the even field when in an interlaced mode.
vsyncend_even	Indicates the vertical sync end timing value for the even field when in an interlaced mode.
vblankend_even	Indicates the vertical blank end timing value for the even field when in an interlaced mode.
vtotal_even	Indicates the vertical total timing value for the even field when in an interlaced mode.
frequency	Indicates the pixel clock frequency of the display mode in 16.16 fixed point format. When the VG_MODEFLAG_MANUAL_FREQUENCY flag is set, this value represents the upper 32 bits of the GLCP_DOTPLL MSR, allowing the M, N and P values to be directly specified by the user. If the flag is not set, the Cimarron API picks the closest match that is found in its internal mode tables.

5.2.5.2 Timing Identifiers

The identifiers in Table 5-4 must be included, in order, for each display mode. The syntax for each line is specified in Section 5.2.5 "Mode Timings" on page 34.

Table 5-4. Timing Identifiers

Indentifier	Value Meaning
flags	Corresponds to the member of same name in the VG_DISPLAY_MODE structure (Table 5-3 on page 35).
src_width	
src_height	
mode_width	
mode_height	
panel_width	
panel_height	
panel_tim1	
panel_tim2	
panel_dither_ctl	
panel_pad_sel_low	
panel_pad_sel_high	
hactive	
hblankstart	
hsyncstart	
hsyncend	
hblankend	
htotal	
vactive	
vblankstart	
vsyncstart	
vsyncend	
vblankend	
vtotal	
vactive_even	
vblankstart_even	
vsyncstart_even	
vsyncend_even	
vblankend_even	
vtotal_even	
frequency	

5.2.6 Basic VOP Settings

This section should only be included when the *configure_vop* value is non-zero from Section 5.2.4 "Basic Mode Information" on page 32. If *configure_vop* is 0, this section must be omitted or lxcustom.exe will flag a syntax error.

The syntax for all lines in this section is:

manual_mode_xx\vop_settings identifier value

'xx' in the above syntax refers to the current mode number, starting from 0. The identifiers correspond to entries in the VOP-CONFIGURATIONBUFFER structure from the Cimarron API. The identifiers and their expected values are listed in Table 5-5. The identifiers must be entered in the order shown in the table.

Table 5-5. VOP Settings

Indentifier	Value Meaning	
flags	VOP configuration flags. Can be one or more of the following:	
	Flag	**Description**
	VOP_FLAG_SINGLECHIPCOMPAT	Enables SCx2xx compatibility mode.
	VOP_FLAG_EXTENDEDSAV	Enables extended SAV/EAV codes.
	VOP_FLAG_VBI	Use the task bit to indicate VBI data.
	VOP_FLAG_TASK	Set Task Bit to 1in VIP 2.0 mode. Default is 0.
	VOP_FLAG_SWAP_UV	Swap the U and V data prior to output.
	VOP_FLAG_SWAP_VBI	Swap the VBI bytes prior to output.
mode	VOP operating mode. Must be one of the following:	
	Mode	**Description**
	VOP_MODE_DISABLED	VOP output is disabled.
	VOP_MODE_VIP11	VIP 1.1
	VOP_MODE_CCIR656	CCIR 656 output.
	VOP_MODE_VIP20_8BIT	16-bit VIP 2.0 output.
	VOP_MODE_VIP20_16BIT	8-bit VIP 2.0 output.
	VOP_MODE_601	601 output. The 601 VOP settings must also be specified. See Section 5.2.7 "601 VOP Settings" on page 40
conversion_mode	YUV 4:4:4 to 4:2:2 conversion algorithm. Must be one of the following:	
	Mode	**Description**
	VOP_422MODE_COSITED	Cosited downsample.
	VOP_422MODE_INTERSPERSED	U,V samples from respective co-samples.
	VOP_422MODE_ALTERNATING	U,V samples from alternating co-samples.
vsync_out	Selects the source of the vsync output from the chip. This field is only relevant if connecting the vsync output pin to an external encoder. Must be one of the following:	
	Mode	**Description**
	VOP_MB_SYNCSEL_DISABLED	Vsync out disabled.
	VOP_MB_SYNCSEL_VG	The vsync signal from the display controller.
	VOP_MB_SYNCSEL_VG_INV	The vsync signal from the display controller (inverted).
	VOP_MB_SYNCSEL_STATREG17	Manually via writes to bit 17 of the VIP status register (Writing a 1 causes a 0-1-0 transition).
	VOP_MB_SYNCSEL_STATREG17_INV	Manually via writes to bit 17 of the VIP status register (Writing a 1 causes a 1-0-1 transition).

5.2.7 601 VOP Settings

This section should only be included when the "Basic VOP Settings" on page 39 is included and the VOP mode is set to VOP_MODE_601. lxcustom.exe will flag a syntax error if this section is included when not required.

The syntax for all lines in this section is:

manual_mode_xx\vop_settings\vop601 identifier value

'xx' in the above syntax refers to the current mode number, starting from 0. The identifiers correspond to entries in the VOP_601DATA structure from the Cimarron API. The identifiers and their expected values are listed in Table 5-6. The identifiers must be entered in the order shown in the table.

Table 5-6. 601 VOP Settings

Indentifier	Value Meaning
flags	601 Configuration flags. Can be one or more of the following: **Flag** — **Description** VOP_601_INVERT_DISPE — Invert the polarity of display enable. VOP_601_INVERT_VSYNC — Invert the polarity of hsync. VOP_601_INVERT_HSYNC — Invert the polarity of vsync.
vsync_shift	The field indicator is accomplished in 601 by the position of the vsync pulse relative to the Hysnc pulse. The pulses can be aligned in one of the following ways: **Value** — **Description** VOP_VSYNC_EARLIER_BY4 — The vsync occurs 4 clocks earlier. VOP_VSYNC_EARLIER_BY2 — The vsync occurs 2 clocks earlier. VOP_VSYNC_NOSHIFT — The vsync is not shifted at all. VOP_VSYNC_LATER_BY_X — The vsync occurs X clocks later, where X is specified in *vsync_shift_count*.
vsync_shift_count	Contains the number of dot clocks by which to shift the vsync signal with respect to hsync when *vsync_shift* is set to VOP_VSYNC_LATER_BY_X.
output_mode	Indicates the output size and format. Should be one of the following: **Value** — **Description** VOP_601_YUV_8BIT — 8-bit YUV 4:2:2 VOP_601_YUV_16BIT — 16-bit YUV 4:2:2 VOP_601_RGB_8_8_8 — 24-bit RGB VOP_601_YUV_4_4_4 — YUV 4:4:4

5.2.8 Horizontal Coefficients

This section should only be included if the *custom_coefficients* value is non-zero in the "Basic Mode Information" on page 32. lxcustom.exe will flag a syntax error if this section is included improperly.

The values in this section are used to specify custom filter coefficients for the horizontal graphics scaler. The horizontal filter is a 5-tap filter with 256 phases. The values for each tap must be in the range of -511 to 511. The sum of all taps for each phase should equal 512. For example, the 5 tap values for phase 80 in the default coefficients are -12, 89, 475, -44 and 4.

The syntax for lines in this section is:

manual_mode_xx\custom_hcoeff hcoeff_yy_zz value

where 'xx' refers to the current display mode, 'yy' refers to the current phase and 'zz' refers to the current tap. Lines should be sorted first by phase and then by tap, as demonstrated in Section 5.2.2 "Example Customization File" on page 28.

5.2.9 Vertical Coefficients

This section should only be included if the *custom_coefficients* value is non-zero in the "Basic Mode Information" on page 32. lxcustom.exe will flag a syntax error if this section is included improperly.

The values in this section are used to specify custom filter coefficients for the vertical graphics scaler. The vertical filter is a 3-tap filter with 256 phases. The values for each tap must be in the range of -511 to 511. The sum of all taps for each phase should equal 512. For example, the 3 tap values for phase 80 in the default coefficients are 56, 469 and -13.

The syntax for lines in this section is:

manual_mode_xx\custom_vcoeff vcoeff_yy_zz value

where ‘xx’ refers to the current display mode number, ‘yy’ refers to the current phase and ‘zz’ refers to the current tap. Lines should be sorted first by phase and then by tap, as demonstrated in Section 5.2.2 “Example Customization File” on page 28.

5.2.10 Pre-mode SMBus Writes

This section is used to specify a sequence of SMBus writes to take place immediately before the display mode timings are programmed. The sequence of writes can be of arbitrary length and can write to different SMBus device addresses. Each display mode can have its own sequence of SMBus writes.

This section should only be included if the *premode_smbus_count* value is non-zero in the "Basic Mode Information" on page 32. lcustom.exe will flag a syntax error if this section is included improperly. Only byte-size SMBus writes are currently supported. Word and Dword accesses can be achieved by chaining two or more sequential byte addresses.

Each SMBus write requires three lines specifying the device base, the register address and the value to be written.

The syntax for these three lines is:

manual_mode_xx\postmode_smbus base_yy value

manual_mode_xx\postmode_smbus addr_yy value

manual_mode_xx\postmode_smbus data_yy value

where ‘xx’ refers to the current display mode and ‘yy’ refers to the current SMBus write. SMBus triplets should be specified in ascending order. They are written out in the exact order that they are listed.

5.2.11 Post-mode SMBus Writes

This section is used to specify a sequence of SMBus writes to take place immediately after the display mode timings are programmed. The sequence of writes can be of arbitrary length and can write to different SMBus device addresses. Each display mode can have its own sequence of SMBus writes.

This section should only be included if the *postmode_smbus_count* value is non-zero in the "Basic Mode Information" on page 32. lcustom.exe will flag a syntax error if this section is included improperly. Only byte-size SMBus writes are currently supported. Word and Dword accesses can be achieved by chaining two or more sequential byte addresses.

Each SMBus write requires three lines specifying the device base, the register address and the value to be written.

The syntax for these three lines is:

manual_mode_xx\postmode_smbus base_yy value

manual_mode_xx\postmode_smbus addr_yy value

manual_mode_xx\postmode_smbus data_yy value

where ‘xx’ refers to the current display mode and ‘yy’ refers to the current SMBus write. SMBus triplets should be specified in ascending order. They are written out in the exact order that they are listed.

A Support Documentation

A.1 Revision History

This is a report of the revision/creation process of the AMD Geode™ LX Processor Windows® XP Display Driver Customization Guide. Any revision (i.e., additions, deletions, parameter corrections, etc.) are recorded in the table(s) below.

Table A-1. Revision History

Revision # (PDF Date)	Revisions / Comments
A (March 2005)	Initial release.
B (June 2006)	Engineering edits. See Table A-2 "Edits to Current Revision" for details.

Table A-2. Edits to Current Revision

Section	Revision
Section 2.0 "Registry Settings"	• Edits to Section 2.3 on page 6: — Edited descriptions of PanelEnable, SimultaneousCRT, PanelWidth, and PanelHeight. — EnablePanelUpscale was changed to EnablePanelScale and the description was modified. — Added keys for RotationAngle and PanelInSystem.
Section 3.0 "Driver Escapes"	• Edits to Section 3.1 on page 9: — Updated structure definition code and added sentence to end of section - “Registry settings, when updated, take effect the next time the display mode is changed.” — Changed Input and Output sizes from 68 to 80 in Table 3-1 on page 10. • Added Section 3.3 "Gamma" on page 11. • Added Section 3.4 "Previously Set Mode" on page 13. • Deleted the previous Section 3.3 on TFT/CRT.
Section 4.0 "Custom Panels"	• Modified introductory text and added Section 4.1 "Driver Behavior" on page 15. • Added information on LX_GTF in Step 1 of Section 4.2 on page 16. • Section 4.3.2 on page 17: — In the 4th line of the first example file - show_crt_modes, changed 1 to 0. — Added last section to example file - Extra Resolutions. • Added to description of show_crt_modes in Table 4-1 on page 18: — In general, users should set this value to 0 and use the extra_width and extra_height entries specified in Section 4.3.5 "Extra Resolutions" as they provide greater control and flexibility. • Added Section 4.3.5 "Extra Resolutions" on page 20. • Added Section 4.4 "Examples" on page 21.

www.amd.com

One AMD Place • P.O. Box 3453 • Sunnyvale, CA 94088-3453 USA • Tel: 408-749-4000 or 800-538-8450 • TWX: 910-339-9280 • TELEX: 34-6306