NAME

intel - Intel integrated graphics chipsets

SYNOPSIS

Section "Device"
 Identifier "*devname***"**
 Driver "intel"
 ...
EndSection

DESCRIPTION

intel is an Xorg driver for Intel integrated graphics chipsets. The driver supports depths 8, 15, 16 and 24. All visual types are supported in depth 8. For the i810/i815 other depths support the TrueColor and DirectColor visuals. For the i830M and later, only the TrueColor visual is supported for depths greater than 8. The driver supports hardware accelerated 3D via the Direct Rendering Infrastructure (DRI), but only in depth 16 for the i810/i815 and depths 16 and 24 for the 830M and later.

SUPPORTED HARDWARE

intel supports the i810, i810-DC100, i810e, i815, i830M, 845G, 852GM, 855GM, 865G, 915G, 915GM, 945G, 945GM, 965G, 965Q, 946GZ, 965GM, 945GME, G33, Q33, Q35, G35, GM45, G45, Q45, G43, G41 chipsets, Pineview-M in Atom N400 series, Pineview-D in Atom D400/D500 series, Intel(R) HD Graphics, Intel(R) Iris(TM) Graphics, Intel(R) Iris(TM) Pro Graphics.

CONFIGURATION DETAILS

Please refer to xorg.conf(5) for general configuration details. This section only covers configuration details specific to this driver.

The Intel 8xx and 9xx families of integrated graphics chipsets have a unified memory architecture meaning that system memory is used as video RAM. For the i810 and i815 family of chipsets, operating system support for allocating system memory is required in order to use this driver. For the 830M and later, this is required in order for the driver to use more video RAM than has been pre-allocated at boot time by the BIOS. This is usually achieved with an "agpgart" or "agp" kernel driver. Linux, FreeBSD, OpenBSD, NetBSD, and Solaris have such kernel drivers available.

By default, the i810/i815 will use 8 MB of system memory for graphics if AGP allocable memory is < 128 MB, 16 MB if < 192 MB or 24 MB if higher. Use the **VideoRam** option to change the default value.

For the 830M and later, the driver will automatically size its memory allocation according to the features it will support. Therefore, the **VideoRam** option, which in the past had been necessary to allow more than some small amount of memory to be allocated, is now ignored.

The following driver **Options** are supported

Option "ColorKey" "*integer***"**

This sets the default pixel value for the YUV video overlay key.

Default: undefined.

Option "DRI" "*string***"**

Disable or enable DRI support. A driver name to use can be provided instead of simple boolean value, which will be passed to the GL implementation for it to load the appropriate backend. Alternatively the maximum level of DRI to enable (e.g. "1", "2" or "3") can be specified.

Default: All levels of DRI are enabled for configurations where it is supported.

The following driver **Options** are supported for the i810 and i815 chipsets:

Option "CacheLines" "*integer***"**

This allows the user to change the amount of graphics memory used for 2D acceleration and video when XAA acceleration is enabled. Decreasing this amount leaves more for 3D textures. Increasing it can improve 2D performance at the expense of 3D performance.

Default: depends on the resolution, depth, and available video memory. The driver attempts to allocate space for at 3 screenfuls of pixmaps plus an HD-sized XV video. The default used for a specific configuration can be found by examining the Xorg log file.

Option "DDC" "*boolean***"**

Disable or enable DDC support.

Default: enabled.

Option "Dac6Bit" "*boolean***"**

Enable or disable 6-bits per RGB for 8-bit modes.

Default: 8-bits per RGB for 8-bit modes.

Option "XvMCSurfaces" "*integer***"**

This option enables XvMC. The integer parameter specifies the number of surfaces to use. Valid values are 6 and 7.

Default: XvMC is disabled.

VideoRam *integer*

This option specifies the amount of system memory to use for graphics, in KB.

The default is 8192 if AGP allocable memory is < 128 MB, 16384 if < 192 MB, 24576 if higher. DRI require at least a value of 16384. Higher values may give better 3D performance, at expense of available system memory.

Option "Accel" "*boolean***"**

Enable or disable acceleration.

Default: acceleration is enabled.

The following driver **Options** are supported for the 830M and later chipsets:

Option "Accel" "*boolean***"**

Enable or disable acceleration.

Default: acceleration is enabled.

Option "Present" "*boolean***"**

Enable use of hardware counters and flow control for the Present extension.

Default: Enabled

Option "AccelMethod" "*string***"**

Select acceleration method. There are a couple of backends available for accelerating the DDX. "UXA" (Unified Acceleration Architecture) is the mature backend that was introduced to support the GEM driver model. It is in the process of being superseded by "SNA" (Sandybridge's New Acceleration). Until that process is complete, the ability to choose which backend to use remains for backwards compatibility. In addition, there are a pair of sub-options to limit the acceleration for debugging use. Specify "off" or "none" to disable all acceleration, or "blt" to disable render acceleration and only use the BLT engine.

Default: use UXA (render acceleration)

Option "TearFree" "*boolean*"

Disable or enable TearFree updates. This option forces X to perform all rendering to a backbuffer prior to updating the actual display. It requires an extra memory allocation the same size as a framebuffer, the occasional extra copy, and requires Damage tracking. Thus enabling TearFree requires more memory and is slower (reduced throughput) and introduces a small amount of output latency, but it should not impact input latency. However, the update to the screen is then performed synchronously with the vertical refresh of the display so that the entire update is completed before the display starts its refresh. That is only one frame is ever visible, preventing an unsightly tear between two visible and differing frames. Note that this replicates what the compositing manager should be doing, however TearFree will redirect the compositor updates (and those of fullscreen games) directly on to the scanout thus incurring no additional overhead in the composited case. Also note that not all compositing managers prevent tearing, and if the outputs are rotated, there will still be tearing without TearFree enabled.

Default: TearFree is disabled.

Option "ReprobeOutputs" "*boolean*"

Disable or enable rediscovery of connected displays during server startup. As the kernel driver loads it scans for connected displays and configures a console spanning those outputs. When the X server starts, we then take the list of connected displays and framebuffer layout and use that for the initial configuration. Sometimes, not all displays are correctly detected by the kernel and so it is useful in a few circumstances for X to force the kernel to reprobe all displays when it starts. To make the X server recheck the status of connected displays, set the "ReprobeOutputs" option to true. Please do file a bug for any circumstances which require this workaround.

Default: reprobing is disabled for a faster startup.

Option "VideoKey" "*integer*"

This is the same as the **"ColorKey"** option described above. It is provided for compatibility with most other drivers.

Option "XvPreferOverlay" "*boolean*"

Make hardware overlay be the first XV adaptor. The overlay behaves incorrectly in the presence of compositing, but some prefer it due to it syncing to vblank in the absence of compositing. While most XV-using applications have options to select which XV adaptor to use, this option can be used to place the overlay first for applications which don't have options for selecting adaptors.

Default: Textured video adaptor is preferred.

Option "Backlight" "*string*"

Override the probed backlight control interface. Sometimes the automatically selected backlight interface may not correspond to the correct, or simply most useful, interface available on the system. This allows you to override that choice by specifying the entry under /sys/class/backlight to use.

Default: Automatic selection.

Option "CustomEDID" "*string***"**

Override the probed EDID on particular outputs. Sometimes the manufacturer supplied EDID is corrupt or lacking a few usable modes and supplying a corrected EDID may be easier than specifying every modeline. This option allows to pass the path to load an EDID from per output. The format is a comma separated string of output:path pairs, e.g. DP1:/path/to/dp1.edid,DP2:/path/to/dp2.edid

Default: No override, use manufacturer supplied EDIDs.

Option "FallbackDebug" "*boolean***"**

Enable printing of debugging information on acceleration fallbacks to the server log.

Default: Disabled

Option "DebugFlushBatches" "*boolean***"**

Flush the batch buffer after every single operation.

Default: Disabled

Option "DebugFlushCaches" "*boolean***"**

Include an MI_FLUSH at the end of every batch buffer to force data to be flushed out of cache and into memory before the completion of the batch.

Default: Disabled

Option "DebugWait" "*boolean***"**

Wait for the completion of every batch buffer before continuing, i.e. perform synchronous rendering.

Default: Disabled

Option "HWRotation" "*boolean***"**

Override the use of native hardware rotation and force the use of software, but GPU accelerated where possible, rotation. On some platforms the hardware can scanout directly into a rotated output bypassing the intermediate rendering and extra allocations required for software implemented rotation (i.e. native rotation uses less resources, is quicker and uses less power). This allows you to disable the native rotation in case of errors.

Default: Enabled (use hardware rotation)

Option "VSync" "*boolean*"

This option controls the use of commands to synchronise rendering with the vertical refresh of the display. Some rendering commands have the option to be performed in a "tear-free" fashion by stalling the GPU to wait for the display to be outside of the region to be updated. This slows down all rendering, and historically has been the source of many GPU hangs.

Default: enabled.

Option "PageFlip" "*boolean*"

This option controls the use of commands to flip the scanout address on a VBlank. This is used by glXSwapBuffers to efficiently perform the back-to-front exchange at the end of a frame without incurring the penalty of a copy, or stalling the render pipeline (the flip is performed asynchronrously to the render command stream by the display engine). However, it has historically been the source of many GPU hangs.

Default: enabled.

Option "SwapbuffersWait" "*boolean*"

This option controls the behavior of glXSwapBuffers and glXCopySubBufferMESA calls by GL applications. If enabled, the calls will avoid tearing by making sure the display scanline is outside of the area to be copied before the copy occurs. If disabled, no scanline synchronization is performed, meaning tearing will likely occur.

Default: enabled.

Option "TripleBuffer" "*boolean*"

This option enables the use of a third buffer for page-flipping. The third buffer allows applications to run at vrefresh rates even if they occasionally fail to swapbuffers on time. The effect of such missed swaps is the output jitters between 60fps and 30fps, and in the worst case appears frame-locked to 30fps. The disadvantage of triple buffering is that there is an extra frame of latency, due to the pre-rendered frame sitting in the swap queue, between input and any display update.

Default: enabled.

Option "Tiling" "*boolean*"

This option controls whether memory buffers for Pixmaps are allocated in tiled mode. In most cases (especially for complex rendering), tiling dramatically improves performance.

Default: enabled.

Option "LinearFramebuffer" "*boolean*"

This option controls whether the memory for the scanout (also known as the front or frame buffer) is allocated in linear memory. A tiled framebuffer is required for power conservation features, but for certain system configurations you may wish to override this and force a linear layout.

Default: disabled

Option "RelaxedFencing" "*boolean*"

This option controls whether we attempt to allocate the minimal amount of memory required for the buffers. The reduction in working set has a substantial improvement on system performance. However, this has been demonstrate to be buggy on older hardware (845-865 and 915-945, but ok on PineView and later) so on those chipsets defaults to off.

Default: Enabled for G33 (includes PineView), and later, class machines.

Option "XvMC" "*boolean*"

Enable XvMC driver. Current support MPEG2 MC on 915/945 and G33 series. User should provide absolute path to libIntelXvMC.so in XvMCConfig file.

Default: Disabled.

Option "Throttle" "*boolean*"

This option controls whether the driver periodically waits for pending drawing operations to complete. Throttling ensures that the GPU does not lag too far behind the CPU and thus noticeable delays in user responsible at the cost of throughput performance.

Default: enabled.

Option "HotPlug" "*boolean*"

This option controls whether the driver automatically notifies applications when monitors are connected or disconnected.

Default: enabled.

Option "Virtualheads" "*integer*"

This option controls specifies the number of fake outputs to create in addition to the normal outputs detected on your hardware. These outputs cannot be assigned to the regular displays attached to the GPU, but do otherwise act as any other xrandr output and share a portion of the regular framebuffer. One use case for these extra heads is for extending your desktop onto a discrete GPU using the Bumblebee project. However, the recommendation here is to use PRIME instead to create a single Xserver that can addresses and coordinate between multiple GPUs.

Default: 0

Option "ZaphodHeads" "*string*"

Specify the randr output(s) to use with zaphod mode for a particular driver instance. If you set this option you must use it with all instances of the driver. By default, each head is assigned only one CRTC (which limits using multiple outputs with that head to cloned mode). CRTC can be manually assigned to individual heads by preceding the output names with a comma delimited list of pipe numbers followed by a colon. Note that different pipes may be limited in their functionality and some outputs may only work with different pipes.
For example:

Option "ZaphodHeads" "LVDS1,VGA1"

will assign xrandr outputs LVDS1 and VGA1 to this instance of the driver.

Option "ZaphodHeads" "0,2:HDMI1,DP2"

will assign xrandr outputs HDMI1 and DP2 and CRTCs 0 and 2 to this instance of the driver.

OUTPUT CONFIGURATION

On 830M and better chipsets, the driver supports runtime configuration of detected outputs. You can use the **xrandr** tool to control outputs on the command line as follows:

xrandr --output *output* **--set** *property value*

Note that you may need to quote property and value arguments that contain spaces. Each output listed below may have one or more properties associated with it (like a binary EDID block if one is found). Some outputs have unique properties which are described below. See the "MULTIHEAD CONFIGURATIONS" section below for additional information.

VGA

VGA output port (typically exposed via an HD15 connector).

LVDS

Low Voltage Differential Signalling output (typically a laptop LCD panel). Available properties:

BACKLIGHT - current backlight level (adjustable)

By adjusting the BACKLIGHT property, the brightness on the LVDS output can be adjusted. In some cases, this property may be unavailable (for example if your platform uses an external microcontroller to control the backlight).

scaling mode - control LCD panel scaling mode

When the currently selected display mode differs from the native panel resolution, various scaling options are available. These include

Center

Simply center the image on-screen without scaling. This is the only scaling mode that guarantees a one-to-one correspondence between native and displayed pixels, but some portions of the panel may be unused (so-called "letterboxing").

Full aspect

Scale the image as much as possible while preserving aspect ratio. Pixels may not be displayed one-to-one (there may be some blurriness). Some portions of the panel may be unused if the aspect ratio of the selected mode does not match that of the panel.

Full

Scale the image to the panel size without regard to aspect ratio. This is the only mode which guarantees that every pixel of the panel will be used. But the displayed image may be distorted by stretching either horizontally or vertically, and pixels may not be displayed one-to-one (there may be some blurriness).

The precise names of these options may differ depending on the kernel video driver, (but the functionality should be similar). See the output of **xrandr --prop** for a list of currently available scaling modes.

TV

Integrated TV output. Available properties include:

BOTTOM, RIGHT, TOP, LEFT - margins

Adjusting these properties allows you to control the placement of your TV output buffer on the screen. The options with the same name can also be set in xorg.conf with integer value.

BRIGHTNESS - TV brightness, range 0-255

Adjust TV brightness, default value is 128.

CONTRAST - TV contrast, range 0-255

Adjust TV contrast, default value is 1.0 in chipset specific format.

SATURATION - TV saturation, range 0-255

Adjust TV saturation, default value is 1.0 in chipset specific format.

HUE - TV hue, range 0-255

Adjust TV hue, default value is 0.

TV_FORMAT - output standard

This property allows you to control the output standard used on your TV output port. You can select between NTSC-M, NTSC-443, NTSC-J, PAL-M, PAL-N, and PAL.

TV_Connector - connector type

This config option should be added to xorg.conf TV monitor's section, it allows you to force the TV output connector type, which bypass load detect and TV will always be taken as connected. You can select between S-Video, Composite and Component.

TMDS-1

First DVI SDVO output

TMDS-2

Second DVI SDVO output

TMDS-1 , TMDS-2 , HDMI-1 , HDMI-2

DVI/HDMI outputs. Available common properties include:

BROADCAST_RGB - method used to set RGB color range

Adjusting this property allows you to set RGB color range on each channel in order to match HDTV requirement(default 0 for full range). Setting 1 means RGB color range is 16-235, 0 means

RGB color range is 0-255 on each channel. (Full range is 0-255, not 16-235)

SDVO and DVO TV outputs are not supported by the driver at this time.

See xorg.conf(5) for information on associating Monitor sections with these outputs for configuration. Associating Monitor sections with each output can be helpful if you need to ignore a specific output, for example, or statically configure an extended desktop monitor layout.

MULTIHEAD CONFIGURATIONS

The number of independent outputs is dictated by the number of CRTCs (in X parlance) a given chip supports. Most recent Intel chips have two CRTCs, meaning that two separate framebuffers can be displayed simultaneously, in an extended desktop configuration. If a chip supports more outputs than it has CRTCs (say local flat panel, VGA and TV in the case of many outputs), two of the outputs will have to be "cloned", meaning that they display the same framebuffer contents (or one displays a subset of another's framebuffer if the modes aren't equal).

You can use the "xrandr" tool, or various desktop utilities, to change your output configuration at runtime. To statically configure your outputs, you can use the "Monitor-<type>" options along with additional monitor sections in your xorg.conf to create your screen topology. The example below puts the VGA output to the right of the builtin laptop screen, both running at 1024x768.

```
Section "Monitor"
  Identifier "Laptop FooBar Internal Display"
  Option "Position" "0 0"
EndSection

Section "Monitor"
  Identifier "Some Random CRT"
  Option "Position" "1024 0"
  Option "RightOf" "Laptop FoodBar Internal Display"
EndSection

Section "Device"
  Driver "intel"
  Option "monitor-LVDS" "Laptop FooBar Internal Display"
  Option "monitor-VGA" "Some Random CRT"
EndSection
```

TEXTURED VIDEO ATTRIBUTES

The driver supports the following X11 Xv attributes for Textured Video. You can use the "xvattr" tool to query/set those attributes at runtime.

XV_SYNC_TO_VBLANK

XV_SYNC_TO_VBLANK is used to control whether textured adapter synchronizes the screen update to the vblank to eliminate tearing. It is a Boolean attribute with values of 0 (never sync) or 1 (always sync). An historic value of -1 (sync for large windows only) will now be interpreted as 1, (since the current approach for sync is not costly even with small video windows).

XV_BRIGHTNESS

XV_CONTRAST

REPORTING BUGS

The xf86-video-intel driver is part of the X.Org and Freedesktop.org umbrella projects. Details on bug reporting can be found at https://01.org/linuxgraphics/documentation/how-report-bugs. Mailing lists are also commonly used to report experiences and ask questions about configuration and other topics. See lists.freedesktop.org for more information (the xorg@lists.freedesktop.org mailing list is the most appropriate place to ask X.Org and driver related questions).

SEE ALSO

Xorg(1), xorg.conf(5), Xserver(1), X(7)

AUTHORS

Authors include: Keith Whitwell, and also Jonathan Bian, Matthew J Sottek, Jeff Hartmann, Mark Vojkovich, Alan Hourihane, H. J. Lu. 830M and 845G support reworked for XFree86 4.3 by David Dawes and Keith Whitwell. 852GM, 855GM, and 865G support added by David Dawes and Keith Whitwell. 915G, 915GM, 945G, 945GM, 965G, 965Q and 946GZ support added by Alan Hourihane and Keith Whitwell. Lid status support added by Alan Hourihane. Textured video support for 915G and later chips, RandR 1.2 and hardware modesetting added by Eric Anholt and Keith Packard. EXA and Render acceleration added by Wang Zhenyu. TV out support added by Zou Nan Hai and Keith Packard. 965GM, G33, Q33, and Q35 support added by Wang Zhenyu.