This file is a part of RivaTuner tweaking utility and it is not intended for external usage. Please use RivaTuner's built-in context help system to view this file instead of opening it outside the utility.

Tip: To open RivaTuner's context help right-click an option you want to know about then select 'Whats this?' from the popup menu.

OK

This button closes RivaTuner or sends it to tray if **Send to tray on close** option is enabled under **Settings** tab.

Custom tab order preview

This window allows you to preview tab order.

Tab order list

This is the list of tabs available in the currently opened property sheet. You can change default tab order and drag the most frequently used tabs to the top or even hide the tabs you don't use.

Target adapter

This dropdown list allows you to select the target display adapter on multiple monitor systems.

Device drivers list

This is a list of device drivers, which are currently loaded in memory. Once you select a driver in this list, information about its' sections is reflected in the **Device driver sections list**.
The following columns are available for each driver:

- **Driver** displays driver name
- **Base addr** displays driver's load (base) address.
- **Binary** displays fully qualified path to the driver's binary file.

Device driver sections list

This is a list of sections for the driver, which is currently selected in **Device drivers list**.
The following columns are available for each section:

• **Section** displays 1-based section index. Page fault (**PF**) postfix is added to the section index if its' memory is not fully readable and contains page fault areas.
• **Name** displays section name.
• **Type** displays section type and allows you to see if the section contains code or data.
• **Flags** displays full section characteristics flags. Some of the characteristics are decoded in the next columns.
• **RVA** displays section relative virtual address.
• **Size** displays section size in bytes.
• **Size** displays section size in bytes.
• **D** displays discardable section status.
• **NC** displays non-cacheable section status.
• **NP** displays non-pageable section status.
• **X** displays execution section permission.
• **R** displays read section permission.
• **W** displays write section permission.

Tips:

• You may select a section by ckicking to the left of its' name.
• All non-discarable non-pageable code sections are selected by default.

Ignore PF areas

By default RivaTuner doesn't dump the sections containing page fault (**PF**) areas. However, you may enable this option to override this behavior.

Tip: The sections containing page fault areas are marked with **PF** postfix after the section index in **Device driver sections list**.

Dump

Click this button to dump driver header and selected driver sections' memory to files.

Tips:

• By default RivaTuner doesn't dump the sections containing page fault (**PF**) areas. However, you may enable **Ignore PF areas** option to override this behavior.
• After dumping driver header memory is stored in **<driver_name>.00** file in the application folder. Selected sections are stored in **<driver_name>.<section_index>** files.
If a section contains page fault areas and **Ignore PF areas option** is enabled, page faults mask is also stored in **<driver_name>.<section_index>PF** file.

Device status bar

This status bar displays low-level hardware diagnostic info for the currently selected adapter. You can click the **Customize** button in order to activate **Hardware customization toolbar** which will allow you to tweak your graphics subsystem at low level.

Tips:

• **WARNING!** In low-level tweaking mode RivaTuner mostly doesn't use API /display driver and works with your graphics hardware directly. Improper low-level tweaking can cause OS to hang on certain hardware configurations.
It is strongly recommended to avoid low-level tweaking if you don't understand what are you doing.
• This status bar displays diagnostic info for supported display adapters only.

Enable low-level hardware overclocking

This setting allows you to enable low-level hardware overclocking. Unlike driver-level overclocking, it bypasses the driver and directly programs the GPU's registers. To ensure proper operation of the clock frequency controls RivaTuner must first detect default core and memory clock frequencies. Reboot is strongly recommended in order to detect them correctly. You'll be prompted to reboot your PC after enabling this option. However, power users have the option to ignore this recommendation and detect defaults without reboot. Don't use this mode if you aren't sure that RivaTuner shows the real defaults in the **Current core / memory clock frequency** fields.

Tips:

• RivaTuner automatically reverts clock frequencies to their default values after disabling low-level hardware overclocking.
• RivaTuner automatically deletes all **overclocking profiles** after disabling low-level overclocking.
• You will usually see fractional values in the **Current core / memory clock frequency** fields. Clock frequencies are usually derived from 13500 / 14318 / 27000 kHz oscillator, which is called reference / crystal frequency. Most integer clocks cannot be derived from the crystal frequency, therefore you will not be able to adjust a clock frequency in precise 1MHz increments. When you try to select any clock frequency, the closest value that can be generated is selected instead, for example 458.176MHz (32*14318 kHz) instead of 460MHz.
• You will not be able to enable low-level overclocking on GeForce FX GPU family. Access to this feature is purposely blocked to avoid interference with GeForce FX-specific dynamic driver-level clock frequency adjustment.

Low-level clock frequency adjustment sliders

These sliders allow you to adjust the core and memory clock frequencies.

Tips:

• Yellow exclamation sign over core or memory icons indicates that the respective clock is beyond its safe overclocking range. This range depends on hardware and it is highlighted on both sliders (from 90% till 125% of default clock frequency on NVIDIA-based boards, VGA BIOS dependent on ATI-based boards).
• The range of the clock frequencies you can set is calculated using the default clock frequency. By default, you may adjust it from 50% till 150% of the default clock frequency, but you can try to override both limits with **MinClockLimit** / **MaxClockLimit** registry entries.
• Clock frequencies can be adjusted with a constant step called clock frequency granularity. The granularity depends on RivaTuner's clock frequency generation mode and your clock frequency generator's possibilities. By default RivaTuner uses the safest granularity that requires minimum clock generator access (2.25MHz for ATI RADEON 8500 boards, 14.318 MHz for GeForce2 Ultra boards etc.). However, you may try to adjust the granularity by pressing the **More** button.

Current core clock frequency

This field displays the current core clock frequency.

Tips:

• You will usually see fractional values in the **Current core / memory clock frequency** fields. Clock frequencies are usually derived from 13500 / 14318 / 27000 kHz oscillator, which is called reference / crystal frequency. Most integer clocks cannot be derived from the crystal frequency, therefore you will not be able to adjust a clock frequency in precise 1MHz increments. When you try to select any clock frequency, the closest value that can be generated is selected instead, for example 458.176MHz (32*14318 kHz) instead of 460MHz.
• Sometimes RivaTuner may show abnormally low core clock frequency on certain mobile NVIDIA boards (e.g. GeForce2 Go, Quadro2 Go and so on).
Due to NVIDIA PowerMizer power saving technology these graphics processors run at 16MHz in Idle. It is strictly recommended to avoid enabling low-level overclocking when your GPU is in such state because RivaTuner will not be able to detect the real default clock frequencies correctly.

Current memory clock frequency

This field displays the current memory clock frequency.

Tips:

• You will usually see fractional values in the **Current core / memory clock frequency** fields. Clock frequencies are usually derived from 13500 / 14318 / 27000 kHz oscillator, which is called reference / crystal frequency. Most integer clocks cannot be derived from the crystal frequency, therefore you will not be able to adjust a clock frequency in precise 1MHz increments. When you try to select any clock frequency, the closest value that can be generated is selected instead, for example 458.176MHz (32*14318 kHz) instead of 460MHz.
• RivaTuner may show double memory clock frequency on certain GeForce2 MX and GetForce4 MX display adapters equipped with SDR memory. This problem is caused by hardware feature of clock frequency generator, which can be hardwired to halve resulting MPLL clock frequency.
MPLL doubling is used on the boards equipped with DDR memory, but some hardware vendors also use it on usual SDR boards to fool the customers. RivaTuner contains the list of such adapters in its database, which allows it to correct the memory clock frequency by halving it when necessary.
RivaTuner can also use a heuristic detection algorithm when it cannot find your display adapter in the database. This allows RivaTuner to report correct clock frequency on the most of such boards. However it is impossible to correct the clock frequency with heuristic algorithm when it is rather low (less or equal to 133MHz).

Maximum memory clock frequency

This is the maximum memory clock frequency you can set.

Tips:

• The range of the clock frequencies you can set is calculated using the default clock frequency. By default, you mat adjust it from 50% till 150% of the default clock frequency, but you can try to override the upper limit with the **MaxClockLimit** registry entry.

Apply overclocking at Windows startup

This option allows you to apply low-level overclocking settings at Windows startup.

Tips:

• In order to avoid clock settings interference don't try to use this option simultaneously with **Apply overclocking at Windows startup** under driver-level overclocking properties.
• You can press and hold **Ctrl** during Windows startup to reset startup overclocking settings.

Save

This button allows you to save current clock frequencies as startup overclocking settings.
This button is only active when the **Apply overclocking at Windows startup** option is enabled and current clock frequencies differ from current startup settings.

Save overclocking profile

This button allows you to save the current clock frequencies as **overclocking profile**, which will then be added to the adjacent **overclocking profiles list**. You can use profiles list later for quick setting these clock frequencies.

Tips:

• The clock frequencies must be tested and applied before you can save them to overclocking profile.
• You can associate an overclocking profile you have created with a launcher item via the **Launcher** tab.

Delete overclocking profile

This button allows you to remove an overclocking profile. It is only active when you select a profile in the adjacent list.

Overclocking profiles list

This list allows you to select **overclocking profile** you have created. Once selected, you may apply the clock frequencies stored in the profile without testing them.

Restore clock frequencies after suspended mode

Enabling this option allows saving clock frequencies when the system is entering standby mode and restoring them after resuming.

Tip: RivaTuner must reside in memory to save and restore clock frequencies.

More

This button allows to you to customize hardware dependent low-level overclocking properties.

Clock frequency generation accuracy

This option allows you to change clock frequency generation accuracy and select preferred clock frequency granularity. You can select one of the following modes:

• **Low (safest)** mode allows RivaTuner to change both memory / core PLL feedback dividers, but freezes memory PLL reference divider on DDR-based boards when it is set to the lowest possible value (1). Both memory and core PLL post dividers are also frozen in this mode. Granularity in this mode depends on default reference and post dividers and on some DDR-based boards it is limited by the crystal frequency (14318 / 27000 KHz depending on the crystal used). This mode provides the lowest clock generation accuracy but the highest stability.
• **Normal** mode allows RivaTuner to change all dividers, but the reference dividers are limited if the clock frequency is above 250000 kHz.
In this mode clock frequency granularity is equal to driver-level overclocking granularity. This mode may require framebuffer reinitialization on DDR-based display adapters and may cause temporary desktop image corruption or flashing.
• **High** is the same as **Normal**, but reference dividers aren't limited whatever the clock frequency. In this mode the clock frequencies can be adjusted with 0.5% accuracy.
• **Maximum** mode is similar to **High** mode, but clock frequency generation accuracy is not limited with 0.5%. The closest possible clock frequency will be selected. This mode provides the highest clock frequency generation accuracy but the lowest stability.

Custom clock frequency generation accuracy options

These options allow you to tune the clock frequency generation accuracy manually. It is strongly recommended to use the **Clock frequency generation accuracy** combo box instead of direct changing these options. Edit these options directly only when you perfectly understand what are you doing.
All the clock frequency generation accuracy options tune the PLL dividers calculation algorithm. You can change the following options in order to customize MNP calculation:

• **Freeze NVPLL/MPLL post divider** options prevent RivaTuner from changing the post dividers when setting a clock frequency. You may enable these options in order to improve stability. It will reduce the clock frequency generation accuracy when the clock frequency is below the maximum Fvco / 2.
• **Freeze MPLL reference divider** option prevent RivaTuner from tuning MPLL reference divider on DDR based boards when it is set to the lowest value (1).
Disabling this option allows RivaTuner to change MPLL reference divider regardless of its' current state. It may require framebuffer reinitialization on certain DDR based boards and may cause a temporary desktop image corruption.
Enabling these option allows to avoid framebuffer reinitialization when programming the MPLL, but results in the lowest clock generation granularity accuracy. The most of DDR based boards will be able to adjust a clock frequency with the crystal frequency granularity only when it is enabled.
• **Limit the maximum NVPLL/MPLL reference divider** options will force RivaTuner to limit the reference dividers with 6 when the clock frequency is above 250000KHz and limit it with 2 when the clock frequency is above 340000KHz.
• **Use core/memory clock early match criteria** options force RivaTuner to select the minimum post and reference dividers, which can derive the clock frequency with 0.5% accuracy.

Fake DDR workaround

RivaTuner may show double memory clock frequency on certain GeForce2 MX200 / MX400 and GeForce4 MX420 boards. This problem is caused by a hardware feature of the clock frequency generator, which can be hardwired to halve the MPLL clock frequency. MPLL doubling is used on the boards equipped with DDR memory, but some hardware vendors also use it on GeForce2 MX 200 / MX400 and GeForce4 MX420 with 4Mx16 SDR memory modules.
RivaTuner contains a list of such boards in the hardware database, which allows it to correct the memory clock frequency by halving it when it is necessary. RivaTuner can also use the heuristic detection algorithm when it cannot find a display adapter in the database.
It allows to show correct memory clock frequency on the most of such boards. Unfortunately it is impossible to detect this problem on the boards with slow (e.g. 133MHz) 4Mx16 SDR memory.
RivaTuner will attempt to detect problematic boards automatically when this option is set to **auto**.
Unfortunately it is impossible to detect this problem on the boards with slow (e.g. 133MHz) 4Mx16 SDR memory. In this case, you can use this option in order to fix this problem. Set it to **force on** to force RivaTuner to halve displayed memory clock frequency.
You may also set it to **force off** to force RivaTuner to show the real MPLL frequency.

Clock frequency generation accuracy

This option allows you to change clock frequency generation accuracy and select preferred clock frequency granularity. You can select one of the following modes:

• **Low (safest)** mode allows RivaTuner to change both memory / core PLL feedback dividers, but freezes reference divider. The granularity in this mode depends on default reference divider. This mode provides the lowest clock generation accuracy but the highest stability.
• **High** mode allows RivaTuner to calculate new reference divider each time when you change the clock frequencies. It automatically calculates and sets the minimum reference divider, that provides requested clock frequencies generation with 1MHz accuracy.
• **Custom** mode allows you to select desired reference divider manually. This mode provides the highest clock generation accuracy but may result in reducing system stability.

Tips:

• This option is only available on pre-R5xxx GPUs.

Reference divider status

This window displays the current reference divider status, which depends on the clock frequency generation accuracy mode you've selected.

Tips:

• This option is only available on pre-R5xxx GPUs.

Reference divider slider

This slider allows selecting reference divider manually when **Custom** clock frequency generation accuracy mode is selected.

Tip:

• This option is only available on pre-R5xxx GPUs.

Show effective memory clock frequency

This option allows you to select memory clock frequency visualization mode and display either real or effective clocks.

Reset memory controller after setting memory clock

Unlike the most third party overclocking tools changing just the clock frequency generator's dividers during memory clock frequency programming, RivaTuner can also optionally suspend memory clock frequency generator before programming it and fully restart it simultaneously with memory controller after programming is done. This allows increasing maximum achievable stable memory clock by up to 100MHz on some low-end display adapters (e.g. RADEON X550).

Tips:

• This option is only available on R3xxx GPUs.

Allow driver-level Overdrive 3D clock frequency control

This option allows RivaTuner to adjust 3D clocks on **Overdrive2** capable and **Overdrive3** capable display adapters with separate 2D / 3D clocking. When this option is enabled, RivaTuner adjusts 3D clocks on such cards via the Catalyst driver instead of accessing clock generator directly.

Tips:

• RivaTuner's Overdrive 3D clock frequency control will not work on the Catalyst 6.4 driver due to ATI bug. The problem is not RivaTuner specific and is also echoed in **ATI OverclockerX1K**, **HIS iTurbo** and **Sapphire Trixx** overclocking tools. The bug is fixed in the Catalyst 6.5 drivers suite.
• Due to specifics of ATI drivers, third-party overclocking settings done by tools like **RivaTuner**, **HIS iTurbo** and **Sapphire Trixx** are overridden by Catalyst Control Center's Overdrive settings.
You must disable manual overclocking in the Catalyst Control Center to get RivaTuner's Overdrive 3D clock frequency control working.

Allow VGA BIOS script parser usage for clock frequency programming

On ATI RADEON X1000 graphics card family RivaTuner can set clock frequencies via hardcoded GPU-specific hardware initialization codepaths like the most third party R5xxx overclocking tools, as well as via parsing VGA BIOS initialization scripts like ATI's own overclocking tools (**CCC** and **OverclockerX1K**). This option allows you to select preferred method of clock frequency setting.

Tips:

• This option is only available on R5xxx GPUs.
• This option has no effect on **Overdrive2** capable and **Overdrive3** capable display adapters with separate 2D / 3D clocking if **Allow driver-level Overdrive 3D clock control** is ticked. In this case clock frequency programming is done entirely at the Catalyst driver.
• Unlike ATI's own script parsers executing the scripts completely, RivaTuner's VGA BIOS parser is intelligent and can programmatically skip some portions of scripts causing undesired side effects like automatic memory timings tuning performed when memory clock frequency is changed via **ATI CCC** or **OverclockerX1K**. RivaTuner provides you both methods and allows you either to change memory clock frequency only or to allow the parser to tune memory timings synchronically with memory clock frequency change.

Allow VGA BIOS script parser to tune timings on memory clock change

On ATI RADEON X1000 graphics card family RivaTuner can set clock frequencies via hardcoded GPU-specific hardware initialization codepaths like the most third party R5xxx overclocking tools, as well as via parsing VGA BIOS initialization scripts like ATI's own overclocking tools (**CCC** and **OverclockerX1K**).
Unlike ATI's own script parsers executing the scripts completely, RivaTuner's VGA BIOS parser is intelligent and can programmatically skip some portions of scripts causing undesired side effects like automatic memory timings tuning performed when memory clock frequency is changed via **ATI CCC** or **OverclockerX1K**. RivaTuner provides you both ways.
This option allows you either to force the parser to change memory clock frequency only or to allow the parser to tune memory timings synchronically with memory clock frequency change.

Tips:

• This option is only available on R5xxx GPUs.
• This option has no effect on **Overdrive2** capable and **Overdrive3** capable display adapters with separate 2D / 3D clocking if **Allow driver-level Overdrive 3D clock control** is ticked. In this case clock frequency programming is done entirely at the Catalyst driver.
• This option has effect only when **Allow VGA BIOS script parser usage for clock frequency programming** is ticked.

Enable low-level fan control

This setting allows you to enable low-level fan control. Unlike driver-level fan control, it bypasses the driver and directly programs fan controller's registers. To ensure proper operation of fan control RivaTuner must first detect default settings of fan controller.
Reboot is strongly recommended in order to detect them correctly. You'll be prompted to reboot your PC after enabling this option. However, power users have the option to ignore this recommendation and detect defaults without reboot. Don't use this mode if you aren't sure that RivaTuner shows the real default state of your fan controller.

Tips:

• RivaTuner automatically reverts fan controller's settings to their default values after disabling low-level fan control.

Fan speed adjustment slider

These slider allow you to adjust fan speed manually.

Tips:

• Take a note that fan speed is not specified directly, value you set is a **fan duty cycle** passed to fan's **PWM** (**Pulse Width Modulation**) controller. **Fan duty cycle** is defined with **Ton/T** formula, were **T** is pulse width modulation frequency period and **Ton** is a part of period **T**, during which controller supplies voltage to the fan's input. Depending on their constructive specifics, different fans have different minimum fan duty cycles, causing fan to rotate. So be careful and take a note that some fans can work at low duty cycles below **10%**, but some can stop even at **50%**.
Due to this specific, by default the minimum value you can set is limited with **25%**. However, you may override it with **MinFanSpeedLimit** registry entry.

Fixed fan speed mode

Select this option to force fan to run at constant speed you have specified.

Automatic fan speed mode

Select this option to force fan controller to adjust fan speed dynamically.

Tips:

• Dynamic fan speed adjustment algorithm depends on the type of fan controller installed on your display adapter and on VGA BIOS, which usually programs initial parameters of this algorithm. The most of fan controllers (e.g. **LM63**, **F75363S**, **LM64**) use sets of programmable temperature ranges and the corresponding fan speeds for them. Some controllers (e.g. **ADT7473**) use more advanced speed control technique and changes speed more softly, trying to prevent temperature from reaching programmed maximum value using very soft fan speed tuning.

Apply fan settings at Windows startup

This option allows you to apply fan settings at Windows startup.

Tips:

• You can press and hold **Ctrl** during Windows startup to reset startup fan settings.

Save fan profile

This button allows you to save the current fan settings as **fan profile**, which will then be added to the adjacent **fan profiles list**. You can use profiles list later for quick applying these settings.

Tips:

• You can associate a **fan profile** you have created with a launcher item via the **Launcher** tab.

Delete fan profile

This button allows you to remove a **fan profile**. It is only active when you select a profile in the adjacent list.

Fan profiles list

This list allows you to select a **fan profile** you have created. Once selected, you may apply the settings stored in the profile.

Save

This button allows you to save current fan settings as startup fan settings. This button is only active when the **Apply fan settings at Windows startup** is enabled and current fan settings differ from current startup fan settings.

Restore fan settings after suspended mode

Enabling this option allows saving fan settings when the system is entering standby mode and restoring them after resuming.

Tip: RivaTuner must reside in memory to save and restore fan settings.

AGP sideband addressing support

This option allows NVStrap driver to override VGA BIOS defined AGP SBA capability and control SBA support without editing and flashing VGA BIOS.

Tips:

• NVStrap is a tiny driver, which loads before OS kernel and configures your hardware before the Detonator drivers.
It allows NVStrap to override VGA BIOS defined AGP SBA/FW capabilities as well as override the PCI DeviceID and convert a GeForce to a Quadro and vice versa.
• NVStrap is integrated in the operating system and it is loaded always, even when you load OS in safe mode. It is not enough to boot in safe mode in order to avoid the driver loading. The only way to uninstall it without loading OS is to remove the %WinDir%\System32\Drivers\NVStrap.sys file.

AGP fastwrites support

This option allows NVStrap driver to override VGA BIOS defined AGP FW capability and control FW support without editing and flashing VGA BIOS.

Tips:

• NVStrap is a tiny driver, which loads before OS kernel and configures your hardware before the Detonator drivers.
It allows NVStrap to override VGA BIOS defined AGP SBA/FW capabilities as well as override the PCI DeviceID and convert a GeForce to a Quadro and vice versa.
• NVStrap is integrated in the operating system and it is loaded always, even when you load OS in safe mode. It is not enough to boot in safe mode in order to avoid the driver loading. The only way to uninstall it without loading OS is to remove the %WinDir%\System32\Drivers\NVStrap.sys file.

Graphics adapter identification

This option allows NVStrap driver to override the PCI DeviceID and covert your GeForce to Quadro and vice versa without any soldering and patching the Detonator drivers.
It can unlock all the professional Quadro features and force the OpenGL ICD to use CAD optimized code on any pre-GeForce4 based board. It can also force the OpenGL ICD to use game optimized code on the real pre-NV25GL based Quadro boards.

Tips:

• NVStrap is a tiny driver, which loads before OS kernel and configures your hardware before the Detonator drivers.
It allows NVStrap to override VGA BIOS defined AGP SBA/FW capabilities as well as override the PCI DeviceID and convert a GeForce to a Quadro and vice versa.
• NVStrap is integrated in the operating system and it is loaded always, even when you load OS in safe mode. It is not enough to boot in safe mode in order to avoid the driver loading. The only way to uninstall it without loading OS is to remove the %WinDir%\System32\Drivers\NVStrap.sys file.
• The Detonators 30.82 and higher are protected against strapping based PCI DeviceID modification. Use **NVStrap antiprotection** patch scripts in order to use this function of the NVStrap with these drivers. Alternately, GeForce FX and newer GPUs owners may activate **Use ROM straps for PCI DeviceID programming** option, which makes NVStrap PCI DeviceID override invisible for NVIDIA driver and eliminates the need of installing additional antiprotection patch script.

Use ROM straps for PCI DeviceID programming

This option allows NVStrap driver to use alternate PCI DeviceID programming technique. This makes NVStrap PCI DeviceID override invisible for NVIDIA driver, eliminating the need of installing additional antiprotection patch script. It's strongly recommended to use this option when it is available.

Tips:

• NVStrap is a tiny driver, which loads before OS kernel and configures your hardware before the Detonator drivers.
It allows NVStrap to override VGA BIOS defined AGP SBA/FW capabilities as well as override the PCI DeviceID and convert a GeForce to a Quadro and vice versa.
• NVStrap is integrated in the operating system and it is loaded always, even when you load OS in safe mode. It is not enough to boot in safe mode in order to avoid the driver loading. The only way to uninstall it without loading OS is to remove the %WinDir%\System32\Drivers\NVStrap.sys file.
• This option is available on GeForce FX and newer graphics processors only.

Active pixel/vertex units configuration

This dropdown list allows altering graphics processor's configuration by disabling currently active or enabling hardware / software masked pixel and vertex units on NV4x graphics processors.
When you select **determined by VGA BIOS** mode, pixel and vertex units configuration is defined by **hardware units mask** and **software units mask**, if it exists in VGA BIOS.
Hardware units mask is hardwired with configuration resistors on the GPU package. Hardware units mask is set by GPU manufacturers and used to mark physically damaged or instable pixel and vertex units and prevent them from enabling by BIOS and third party tools like NVStrap driver.
Software units mask may also block some fully functional units at VGA BIOS level.

When you select **custom** mode, NVStrap is able to override state of each pixel and vertex unit, if it is not locked with hardware mask. NVStrap is also able to override hardware mask and enable locked units with **Allow enabling hardware masked units** option.

Allow enabling hardware masked units

By default programmability of each pixel / vertex unit is defined by **hardware units mask**, which is hardwired with configuration resistors on the GPU package. Hardware units mask is set by GPU manufacturers and used to mark physically damaged or instable pixel and vertex units and prevent them from enabling by BIOS and third party tools like NVStrap driver.
However, this option allows the NVStrap driver to override graphics processor's hardware units mask before actually programming desired units configuration. This allows the driver to enable any unit, even hardware masked one.

WARNING! Use this option with extreme caution at your own risk. Enabling hardware masked units, which haven't passed hardware quality tests, can cause absolutely unpredictable results including permanent hardware damages.

Unlock professional capabilities

This option allows you to activatie professional OpenGL capabilities of NV4x graphics processors. Use it in conjunction with **Quadro** mode selected in **Graphics adapter identification** to transform GeForce 6x00 to professional OpenGL accelerator clone.

Customize

This button is active when **custom** mode is selected in **Active pixel/vertex units configuration** dropdown list. Click it to activate **Custom graphics processor configuration** panel and override default state of programmable pixel and vertex units.

Pixel/vertex units mask status

This window displays default, current and target masks for active pixel and vertex units. **Default** mask indicates which units are activated by hardware and VGA BIOS. **Current** mask shows you which pixel and vertex units are currently active. **Target** mask defines a set of pixel and vertex units you are about to enable via the NVStrap driver after the next reboot.

Tips:

• Some pixel / vertex units can be locked with **hardware units mask**, which is hardwired with configuration resistors on the GPU package. Hardware units mask is set by GPU manufacturers and used to mark physically damaged or instable pixel and vertex units and prevent them from enabling by BIOS and third party tools like NVStrap driver.
Differences in **Current** and **Target** masks after reboot can be caused by your attempt to enable such hardware masked units without ticking **Allow enabling hardware masked units** option.

Pixel/vertex units mask bit editor

This window represents **Target** units mask in binary format.

Tips:

• You may click mask bit to select it and highlight its' description in **Pixel/vertex units mask bits list** window.
• You may double-click mask bits to toggle them. Once you toggle a bit in this window, changes in mask are automatically reflected in both **Pixel/vertex units mask status** and **Pixel/vertex units mask bits list** windows.

Pixel/vertex units mask bits list

This window represents a list of bits in **Target** units mask. There are four columns available for each bit:

- **Bit field / bit mask** displays decimal 0-based bit index and hexadecimal bit mask.
- **Description** displays bit description if available.
- **State** displays current activity state for the pixel / vertex unit controlled with this bit.
- **HW masked** displays hardware masking state for the pixel / vertex unit controlled with this bit. Hardware masked units cannot be activated until you tick **Allow enabling hardware masked units** option.

Tips:

- You may click mask bit in this list to select it and highlight it in **Pixel/vertex units mask bits editor**.
- You may click to the right of mask bit in this list to toggle it. Once you toggle a bit, changes in mask are automatically reflected in both **Pixel/vertex units mask status** and **Pixel/vertex units mask bit editor** windows.

Install

This button allows you to install / reinstall the NVStrap driver. After installing it will be integrated in operating system kernel and automatically loaded after the next reboot.

Tips:

• NVStrap is a tiny driver, which loads before OS kernel and configures your hardware before the Detonator drivers.
It allows NVStrap to override VGA BIOS defined AGP SBA/FW capabilities as well as override the PCI DeviceID and convert a GeForce to a Quadro and vice versa.
• NVStrap is integrated in the operating system and it is loaded always, even when you load OS in safe mode. It is not enough to boot in safe mode in order to avoid the driver loading. The only way to uninstall it without loading OS is to remove the %WinDir%\System32\Drivers\NVStrap.sys file.

Uninstall

This button allows you to uninstall the NVStrap driver. It is active only when the driver is installed.

Tips:

• NVStrap is a tiny driver, which loads before OS kernel and configures your hardware before the Detonator drivers.
It allows NVStrap to override VGA BIOS defined AGP SBA/FW capabilities as well as override the PCI DeviceID and convert a GeForce to a Quadro and vice versa.
• NVStrap is integrated in the operating system and it is loaded always, even when you load OS in safe mode. It is not enough to boot in safe mode in order to avoid the driver loading. The only way to uninstall it without loading OS is to remove the %WinDir%\System32\Drivers\NVStrap.sys file.

AGP transfer rate

This option allows you to set AGP transfer rate via low-level AGP command registers programming. The maximum transfer rate you can select, depends on your motherboard's and graphics adapter's capabilities. Usually display driver checks your hardware capabilities and selects the maximum rate supported by your graphics subsystem.
However, sometimes you may need to decrease transfer rate in order to improve stability on certain systems (e.g. Intel 440LX based platforms may work unstable in AGP 2x mode).

Tips:

• **WARNING!** Reconfiguring AGP on-the-fly, especially if AGP transactions are already initialized, is a risk of causing operating system lockup.
It is strongly recommended to use these settings with caution and reboot system before making any low-level AGP adjustments.

AGP sideband addressing

This setting allows you to control AGP sideband addressing via low-level AGP command registers programming. Usually display driver tests hardware capabilities and enables sideband addressing if it is supported by the motherboard and the graphics adapter. But in some cases you may need to control it manually.

Tips:

• **WARNING!** Reconfiguring AGP on-the-fly, especially if AGP transactions are already initialized, is a risk of causing operating system lockup.
It is strongly recommended to use these settings with caution and reboot system before making any low-level AGP adjustments.
• This setting is grayed if you display adapter operates in AGP 3.0 mode.

AGP fastwrites

This setting allows controlling AGP fastwrites via low-level AGP command registers programming. Usually display driver tests hardware capabilities and enables fastwrites if it is supported by your motherboard and the graphics adapter. But in some cases you may need to control it manually.

Tips:

• **WARNING!** Reconfiguring AGP on-the-fly, especially if AGP transactions are already initialized, is a risk of causing operating system lockup.
It is strongly recommended to use these settings with caution and reboot system before making any low-level AGP adjustments.

Apply AGP settings at Windows startup

This option allows you to apply low-level AGP settings at Windows startup.

Tips:

• You can press and hold **Ctrl** during Windows startup to reset startup AGP settings.

Save

This button allows to save current AGP settings as startup AGP settings. This button is only active when the **Apply AGP settings at Windows startup** is enabled and current AGP settings differ from current startup settings.

Enable refresh overrider

This option allows you to enable programmable refresh overrider module. When you enable refresh overrider RivaTuner hooks display mode changes by handling WM_DISPLAYCHANGE message and immediately sets new refresh rate when it is necessary.

Tips:

• RivaTuner uses **refresh override rules** to program refresh overrider. Each refresh override rule determines the range of display modes / refresh rates, which must be overridden and target refresh rate for those display modes. Refresh override rules are defined with special easy readable pseudo language (e.g. if (width >= 1024 and refresh <= 100) then refresh = max refresh).
You can create and edit refresh override rules with the **Refresh override rule editor**, which can be activated either by pressing the **Add new rule** button or from the context toolbar in the **Refresh override rules list**.
• Due to Windows 2000/XP kernel specific all fullscreen applications set the first refresh rate from the list of supported refresh rates when switching to fullscreen mode. This kernel specific causes well-known 60Hz refresh rate problem. You can use the refresh overrider module to work around this problem. As an alternate way of fixing this problem, you may use built-in **60Hz refresh rate fix wizard**, which fixes this problem at the display drivers level.
Unfortunately there is no way to work around 60Hz problem in every game with refresh overrider. RivaTuner's refresh overrider fixes this problem by trapping display mode change and immediately setting new refresh rate. It may cause some display mode switch sensitive applications to fail.
It is strongly recommended to use the **60Hz refresh rate fix wizard** instead of refresh overrider if both of them are supported by currently selected display adapter and its' drivers.
• It is strongly recommended to avoid using RivaTuner's refresh overrider simultaneously with any other resident refresh override utility.
• RivaTuner must reside in memory when using this option.

Refresh override rules list

This is a list of **refresh override rules** which are currently in use by the built-in refresh overrider module. RivaTuner uses refresh override rules to program refresh overrider. Each refresh override rule determines the range of display modes and refresh rates, which must be overridden and target refresh rate for those display modes. Refresh override rules are defined with special easy readable pseudo language (e.g. if (width >= 1024 and refresh <= 100) then refresh = max refresh).
You can create and edit refresh override rules with the **Refresh override rule editor**, which can be activated either by pressing the **Add new rule** button or from the context toolbar in the **Refresh override rules list**.

Tips:

• You can click right mouse button on any rule to show the **context toolbar**. Some useful actions can be performed via its' buttons.

Refresh override mode

This option allows you to select preferred method of trapping display mode switch, which will be used by RivaTuner's refresh overrider. There are four modes available:

• **Performance mode** allows RivaTuner to override a display mode in the WM_DISPLAYCHANGE message handler. In this mode RivaTuner's refresh overrider brings no additional CPU load, but this may result in incompatibility with certain applications.
• **Blend mode** allows RivaTuner to use a thread-wide hook to trap the moment when the operating system sends the WM_DISPLAYCHANGE message. This provides better compatibility at the cost of some additional CPU load.
• **Compatibility mode** is almost equal to **Blend mode**, but it uses a system-wide hook instead of the thread-wide hook. This mode provides the maximum compatibility, but it can cause the maximum performance impact.
• **Injection mode** allows RivaTuner to inject API hook directly in the application's display mode set routine. Unlike the rest modes, performing refresh override after actual display mode set, this mode totally prevents applications from setting undesired refresh rates.

Current refresh override rule

Displays pseudo code for the current **refresh override rule**.

Tips:

• RivaTuner uses **refresh override rules** to program refresh overrider. Each refresh override rule determines the range of display modes / refresh rates, which must be overridden and target refresh rate for those display modes. Refresh override rules are defined with special easy readable pseudo language (e.g. if (width >= 1024 and refresh <= 100) then refresh = max refresh).
You can create and edit refresh override rules with the **Refresh override rule editor**, which can be activated either by pressing the **Add new rule** button or from the context toolbar in the **Refresh override rules list**.

Add new rule

This button allows you to add new **refresh override rule**.

Tips:

• RivaTuner uses **refresh override rules** to program refresh overrider. Each refresh override rule determines the range of display modes / refresh rates, which must be overridden and target refresh rate for those display modes. Refresh override rules are defined with special easy readable pseudo language (e.g. if (width >= 1024 and refresh <= 100) then refresh = max refresh).
You can create and edit refresh override rules with the **Refresh override rule editor**, which can be activated either by pressing the **Add new rule** button or from the context toolbar in the **Refresh override rules list**.
• RivaTuner will not allow you create crosslinked refresh override rules.

Display mode dimension/color depth/refresh rate bounds

These edit fields allow you to change the current **refresh override rule** by specifying upper and lower bounds for display mode dimension, color depth and refresh rate match criteria. Each pair of edit fields define the range of valid display mode dimensions, color depths and refresh rates for the current rule. RivaTuner's refresh overrider will not handle this rule if at least one parameter doesn't fit into the corresponding range.
You can edit each field manually to change upper/lower range bound, leave it blank to unbound the range or press the **Browse for MIN match criteria**, **Browse for EQU match criteria**, **Browse for MAX match criteria** and **Reset match criteria** buttons.

Tips:

• All the changes you made in these edit fields are immediately reflected in the **Current refresh override rule status bar**.

Browse for MIN match criteria

This button allows you to select any display mode supported by your graphics subsystem and use Its' parameters as the lower bounds for the current match criteria.

Browse for EQU match criteria

This button allows you to select any display mode supported by your graphics subsystem and use Its' parameters as both lower and upper bounds for the current match criteria.

Browse for MAX match criteria

This button allows you to select any display mode supported by your graphics subsystem and use Its' parameters as the upper bounds for the current match criteria.

Reset match criteria

This button allows you to reset the current match criteria.

Set max supported refresh rate for matched display modes

This option allows to override refresh rate with the maximum supported for matched display modes.

Tips:

• **WARNING!** RivaTuner may detect the maximum supported refresh rates incorrectly if the monitor's driver is not installed properly.
It's strongly recommended to avoid using this mode if your monitor is configured as **Plug and Play monitor**, **Default monitor** or **Unknown monitor**.

Set … Hz refresh rate for matched display modes

This option allows to override refresh rate with user defined refresh rate for matched display modes. RivaTuner automatically clips user specified refresh rate and sets the closest refresh rate supported by your graphics subsystem.

Tips:

• **WARNING!** RivaTuner may detect the maximum supported refresh rates incorrectly if the monitor's driver is not installed properly.
It's strongly recommended to avoid specifying invalid refresh rates when your monitor is configured As **Plug and Play monitor**, **Default monitor** or **Unknown monitor**.

Plug and Play identifier

This edit field allows you to enter plug and play identifier for your monitor. The identifier consists of three letters identifying monitor manufacturer and four hexadecimal digits identifying monitor model. RivaTuner automatically reads plug and play identifier from EDID if your monitor is DDC capable and the display driver correctly stored EDID in Windows registry.
RivaTuner reads plug and play identifier from the currently installed monitor driver if your monitor is not DDC capable or the display driver failed to store EDID in Windows registry.

Tips:

• **Create** button is grayed if this field is not blank and the plug and play identifier you have entered is invalid.
• You can leave this field blank. This will allow the driver you are about to create to be installed on any monitor.

Model description

This edit field allows you to enter monitor model description. RivaTuner automatically reads monitor model description from EDID if your monitor is DDC capable, EDID contains monitor model description and display driver correctly stored EDID in Windows registry.
RivaTuner reads model description from the currently installed monitor driver if your monitor is not DDC capable or display driver failed to store EDID in Windows registry.

Tips:

• You can leave this field blank, in this case RivaTuner will use **Plug and Play identifier** as model description.

Maximum resolution

These edit fields allow you to enter the maximum resolution, supported by your monitor. RivaTuner automatically reads it from EDID if your monitor is DDC capable and the driver correctly stored EDID in Windows registry.
RivaTuner gets the maximum resolution from the currently installed monitor driver if your monitor is not DDC capable or the display driver failed to store EDID in Windows registry.

Tips:

• **Create** button is grayed if at least one of these fields is blank.

Horizontal / vertical scan frequencies

These edit fields allow you to enter the ranges of horizontal / vertical scan frequencies, supported by your monitor. RivaTuner automatically reads scan frequencies from EDID if your monitor is DDC capable and the driver correctly stored EDID in Windows registry.
RivaTuner reads scan frequencies ranges from the currently installed monitor driver if your monitor is not DDC capable or the display driver failed to store EDID in Windows registry.

Tips:

• **Create** button is grayed if at least one of these fields is blank.

DPMS support

Check this if your monitor is DPMS capable. RivaTuner automatically reads DPMS capabilities from EDID if your monitor is DDC capable and the driver correctly stored EDID in Windows registry. RivaTuner reads DPMS capabilities from the currently installed monitor driver if your monitor is not DDC capable or the display driver failed to store EDID in Windows registry.

Refresh overrides list

This is a list of refresh overrides for the monitor driver you are creating. You can override minimum refresh rate for each display mode supported by your graphics subsystem and fix infamous 60Hz problem under Windows 2000/XP at the monitor driver level.

Select all modes

This button allows you to select all the supported display modes. Once selected, you can press the following buttons: **Reset refresh override for all selected display modes**, **Enter minimum refresh rates for all selected display modes** and **Set maximum refresh rates for all selected display modes**.

Tip: You can toggle display mode selection by clicking to the left of its' name.

Unselect all modes

This button allows you to unselect all the supported display modes.

Tip: You can toggle display mode selection by clicking to the left of its' name.

Reset refresh override for all selected display modes

This button allows you to reset refresh override for all selected display modes.

Tips:

• This button is active if you have selected at least one display mode. You can toggle display mode selection by clicking to the left of its' name or use the **Select all modes** and **Unselect all modes** buttons.

Enter minimum refresh rates for all selected display modes

This button displays the dialog, which allows you to enter the minimum supported refresh rates for all selected display modes.

Tips:

• This button is active if you have selected at least one display mode. You can toggle display mode selection by clicking to the left of its' name or use the **Select all modes** and **Unselect all modes** buttons.
• RivaTuner automatically analyzes your graphics subsystem and selects the closest refresh rates for those display modes, which don't support requested refresh rate.

Set maximum refresh rates for all selected display modes

This button allows you to set the maximum supported refresh rates for all selected display modes.

Tips:

• This button is active if you have selected at least one display mode. You can toggle display mode selection by clicking to the left of its' name or use the **Select all modes** and **Unselect all modes** buttons.
• RivaTuner uses Win32 API in order to fill the list of supported display modes and refresh rates. It means, that it may detect the maximum supported refresh rate incorrectly if you have not properly installed the drivers for your monitor.

Get EDID info

This button allows you to fill all fields with information from the monitor's EDID. This button is grayed if your monitor is not DDC capable or the driver failed to store EDID in Windows registry.

Get driver info

This button allows you to fill all fields with information from the currently installed monitor driver.

Create

This button allows you to create the monitor driver using refresh overrides and data you've entered in the **Plug and Play identifier**, **Maximum resolution**, **Horizontal/vertical scan frequencies** and **DPMS support**. This button is grayed if at least one of these fields is not filled properly.

Tips:

• Once created a driver, you may use it if you've no driver from your monitor manufacturer. You can also use it to fix infamous 60Hz problem at the monitor driver level.

Report categories

This allows you to select the categories, which will be added to the graphics subsystem diagnostic report. The list of report categories depends on currently selected adapter.

Tips:

• You can press the **Capture** button after changing the list of report categories in order to update selected report.
• You can use the **Report categories** list to simplify navigation in the **Report preview** window. Double click any category in order to jump to this category in the preview window.

Report preview

This window displays selected graphics subsystem report preview. You can select either current or sample report with the **Report selection bookmark**.

Tips:

• Current report is always automatically captured by RivaTuner when you activate this dialog. You can also manually capture it with the **Capture current report** button.
• Sample report is just the previously created report, which can be compared with the current one. RivaTuner automatically saves the sample report when you close this dialog and restores it when you open it next time. You can either open any report as sample with the **Open sample report** or capture it with the **Capture sample report** button.
• You can use the **Highlight differences** / **Highlight missed lines** buttons to compare current and sample reports.
• You can use the **Report categories** list in order to customize the report categories.

Report selection bookmark

This allows you to switch between current and sample reports.

Tips:

• Current report is always automatically captured by RivaTuner when you activate this dialog. You can also manually capture it with the **Capture current report** button.
• Sample report is just the previously created report, which can be compared with the current one. RivaTuner automatically saves the sample report when you close this dialog and restores it when you open it next time. You can either open any report as sample with the **Open sample report** or capture it with the **Capture sample report** button.

Capture current report

This button allows you to capture current report. The report will contain only the categories which you have selected in the **Report categories** list.

Capture sample report

This button allows you to capture sample report. The report will contain only the categories which you have selected in the **Report categories** list.

Highlight differences in current report

When this button is pressed RivaTuner compares current and sample reports and highlights the differences in the current report. This feature is very useful when tracking changes in graphics processor's registers or in PCI configuration space.

Highlight differences in sample report

When this button is pressed RivaTuner compares current and sample reports and highlights the differences in the sample report. This feature is very useful when tracking changes in graphics processor's registers or in PCI configuration space.

Highlight missed lines in current report

When this button is pressed RivaTuner compares current and sample reports and highlights a line in current report if it doesn't exist in sample report. This feature is very useful when comparing two reports containing undetermined sets of lines (e.g. when comparing two OpenGL extensions lists).

Highlight missed lines in sample report

When this button is pressed RivaTuner compares current and sample reports and highlights a line in sample report if it doesn't exist in current report. This feature is very useful when comparing two reports containing undetermined sets of lines (e.g. when comparing two OpenGL extensions lists).

Save current report

This button allows you to save current report. You will be able to open it as sample report later.

Open sample report

This button allows you to open any report you have created before as sample report.

Open current report

This button allows you to open any report you have created before as current report.

Monitoring history graph

This window displays monitoring history graph for the corresponding monitoring data source.

Tips:

• You can move mouse cursor over this window to track results at the **monitoring marker** position.
• You can set up to 8 **custom markers** to track values in the corresponding positions. You can click any point on the graph to set custom marker at the corresponding position. To remove a marker you've created hold **Ctrl** and click it. You can also automatically set tracking markers at the graph's extremum points using **Mark minimum …** / **Mark maximum …** commands in the context menu.
• You may hold **Ctrl** when setting a **custom marker** to set it **asynchronically** on the focused graph only.
• You may hold **Shift** when pressing left mouse button on the graph, move mouse cursor then release left mouse button to select a **range**. Once a range is selected, its' statistics is automatically calculated and displayed on the graph. Some of the commands available in the data source's **context menu**, depend on the selection too.
• You can scale monitoring history graph along Y-axis using **vertical range scaling sliders**. You can also automatically scale the graph to fit in the window with **Scale to fit** command in the **context menu**.
• You can configure monitoring data sources via the **context menu** by pressing the **Setup** button.
• Some graphs (e.g. **Framerate**) require **RivaTunerStatisticsServer** to reside in memory for proper functionality. If the server is not loaded in memory, RivaTuner automatically displays **Run server** button on such graphs. The button is also displayed when **Show … in on-screen display** option is ticked in the graph's properties and the server is not loaded in memory. You may click this button to quick-load the server.
• You may press **F11** button to capture screenshot of hardware monitoring history. By default RivaTuner applies its' own watermark logo to the screenshot, however you can easily replace it with your own one by replacing contents of **.\Watermark** subfolder.

Monitoring sessions list

This is a list of hardware monitoring sessions containing in the file you are currently viewing. Click any session to display it on the graphs.

Tips: You may refresh currently selected session with F5 key.

Vertical range scaling sliders

These sliders allow you to scale **monitoring history graph** along Y-axis.

Tip: Yellow exclamation sign displayed over the slider indicates that the monitoring history graph lies beyond the respective scaling limit. Take a note, that it applies to selected **range** only, or to whole graph if there is no selection.

Enable background monitoring

Press this button to enable background monitoring. When background monitoring is enabled, data sources are polled always regardless of the **Hardware monitoring** window visibility.
Otherwise, RivaTuner starts polling data sources when you open this window and stops it when you close the window.

Tip: Be careful with enabling background monitoring, frequent background data polling may affect performance and drastically slow down your system.

Pause monitoring

Press this button to pause monitoring. This function is useful for monitoring history graph analysis.

Clear history

This button allows you to invalidate all monitoring history graphs.

Always on top

This button allows the **Hardware monitoring** window to appear always on top.

Toggle status bar visibility

This button toggles **Hardware monitoring status bar** visibility. This status bar displays some useful information, which allows you to estimate performance hit and disk space consumption caused by hardware monitoring module.

Log to file

Press this button to store hardware monitoring statistics in *.hml log file.

Tip: You may select path to the file and change some other logging related options by pressing the **Setup** button.

Open log

Press this button to open and view previously created *.hml log file.

Tip: RivaTuner automatically registers itself as *.hml file handler so you can also view *.hml log files directly from Windows explorer.

Hardware monitoring status bar

This window displays some useful information, which allows you to estimate performance hit and disk space consumption caused by hardware monitoring module.

Setup

This button allows you to customize monitoring data sources and some additional parameters.

Data polling interval

This edit field allows you to specify interval for monitoring sources data polling.

Tip: Be careful when specifying small (< 500ms) data polling intervals, especially in conjunction with background monitoring. Frequent data polling may affect performance and drastically slow down your system.

Log file path

This edit field allows you to enter path to the log file.

Tip: By default hardware monitoring log is located in **HardwareMonitoring.hml** file in RivaTuner's folder.

Browse

This button allows you to browse for the log file.

Tip: By default hardware monitoring log is located in **HardwareMonitoring.hml** file in RivaTuner's folder.

Recreate existing log files

Keep this option checked to force RivaTuner to recreate existing log files each time when you start new hardware monitoring logging session. Otherwise, RivaTuner will append existing log files.

Stop logging when log size exceeds this value

This option allows you to force RivaTuner to stop logging to a file as soon as log file size exceeds specified value (in megabytes).

Monitoring data sources list

This window is a list of monitoring data sources available on your system. You can change default order and drag the most frequently used sources to the top or uncheck the sources you don't need.

Tips:

• You may customize each monitoring data source by selecting it then clicking **Setup** button. Take a note that only actually existing sources can be customized, that's why **Apply** button must be clicked before customizing any newly added source. Alternately, you may customize source via the context menu of the corresponding graph.
• **Framerate** data source requires **RivaTunerStatisticsServer** to reside in memory. RivaTuner uses the server to collect application runtime related statistics.

Setup

This button allows customizing monitoring data source, which is currently selected in **monitoring data sources list**. Take a note that only actually existing sources can be customized, that's why **Apply** button must be clicked before customizing any newly added source.

Plugins

This button allows selecting active hardware monitoring plugin modules.

Data provider

Some monitoring data sources can receive data from multiple **data providers** (for example many NVIDIA graphics cards can read temperatures either from NVIDIA display driver or from low-level **monitoring plugins**, accessing sensor chip directly). In this case, you can select desired data provider in this window.

Data provider description

This window displays description for the currently selected **data provider**.

Maximum value limit

This edit box allows you to specify custom limit for the corresponding monitoring data source.

Grid spacing

This edit box allows you to specify custom grid spacing the corresponding monitoring data source.

Minimum value limit

This edit box allows you to specify custom limit for the corresponding monitoring data source.

Value offset

This edit box allows you to specify custom offset for the corresponding monitoring data source.

Tips:

• RivaTuner always stores native values when logging monitoring statistics to file. Offsets are added during viewing only.
• This feature can be used to synchronize RivaTuner's temperatures R360 core temperatures with Overdrive's because ATI control panel never shows the real sensor's temperatures on these VPUs and always adds constant 20C offset to it.

Fixed height

By default the graphs' heights are scaled automatically to best fit all the graphs in entire monitoring window, however sometimes you may need to view more detailed Y-axis for some graphs (e.g. if you have multiple thresholds defined for the temperature graph and the graph is too small to show all of them). In this case you may use this option to set **fixed graph height mode**.

Tips:

• You may switch a graph to **fixed graph height mode** visually by capturing and dragging the graph's **upper and lower sizing borders**.
• Once **fixed graph height mode** is selected, **Toggle automatic graph height mode** button is displayed on the graph. You may click it to switch it back to **automatic graph height mode**.

Show in tray icon tooltip

This option allows RivaTuner to display the last polled value for this data source in tray icon tooltip.

Tips:

• This option has effect only when **background monitoring** mode is enabled and RivaTuner is minimized to tray.
• Multiple data sources can be displayed in tray icon tooltip simultaneously.

Show in tray icon

This option allows RivaTuner to display the last polled value for this data source directly in tray icon.

Tips:

• By default RivaTuner uses **multiple tray icons mode**, allowing you to display data from multiple data sources in multiple tray icons simultaneously.
However, you may force **single tray icon mode** using **DisableMultipleTrayIconMode** registry entry.
• This option has effect only when **background monitoring** mode is enabled. If **single tray icon mode** is forced, RivaTuner must be also minimized to tray.
• When you set this option for a source in **single tray icon mode**, RivaTuner automatically resets it for the rest sources.

Tray text color

This is a color of tray text for the data source you are currently editing. You may click it to redefine the color.

Tips:

• This option is active only if **Show in tray icon** is enabled.

Show in on-screen display

You can enable this option to allow RivaTuner to send the last polled value for this data source to **RivaTunerStatisticsServer**'s on-screen display.

Tips:

• **RivaTunerStatisticsServer** must reside in memory in order to render on-screen display during 3D applications runtime.
• You can customize on-screen display appearance in **RivaTunerStatisticsServer**'s properties.
• Multiple data sources can be displayed in **RivaTunerStatisticsServer**'s on-screen display simultaneously.

Run server

This button is active only when **Show in on-screen display** is ticked and RivaTuner detects that **RivaTunerStatisticsServer** required for on-screen display rendering is not loaded in memory. You may click this button to quick-load the server.

Thresholds list

This window is a list of **thresholds** for the corresponding data source. The **thresholds** are user defined critical values of the monitored parameter, which are used to track the moment of reaching or crossing critical point. The thresholds can be configured to launch any external application on **upward** or **downward threshold crossing event**, which allows you to use them for different purposes (e.g. to shut system down when critical temperature is reached).

Tips:

• You may click **Add new threshold** button to add new thresholds.
• You may change threshold order by dragging a threshold to required position. Take a note that the thresholds receive data in the same order as they are displayed (the topmost is the first), so you may use this behavior if you have multiple thresholds associated with the same value.
• You may right-click an existing threshold to customize or remove it.
• You may use thresholds to implement temperature-based overclocking by specifying temperature limits and launching required RivaTuner's overclocking profiles via the command line when the threshold is crossed.

Add new threshold

Press this button to add new **threshold** for the corresponding data source.

More

This button allows you to customize additional data source specific settings.

Record event history via shared memory event queue API

RivaTuner provides so called **shared memory event queue** and open source API for accessing it from third party applications, allowing any application to store own **long event markers** identifying the application's runtime, **application specific instantaneous event markers** identifying an action performed by the application and **global instantaneous event markers** identifying system-wide events in the **Event history** panel. You may use this option to prevent such applications from using RivaTuner's **shared memory event queue**.

Record 3D application event history via statistics server

By default RivaTuner uses **RivaTunerStatisticsServer** for detecting 3D applications' launch and recording **long event markers** identifying 3D application's runtime. You may use this option to restrict 3D applications' event registration via **RivaTunerStatisticsServer**.

Run server

This button is active only when **Record 3D application event history via statistics server** is ticked and RivaTuner detects that **RivaTunerStatisticsServer** required for history recording is not loaded in memory. You may click this button to quick-load the server.

Exceptions list

This is a list of **event history exceptions**.

Tip: The exceptions allow you to force event registration even for 2D applications which are not supporting **shared memory event queue API** and normally not being registered in the **Event history** (e.g. CPU stress-testing applications). Also, the exceptions allow you to disable event registration only for some desired 3D applications if necessary.

Add exception

This button allows you to add new **event history exception**.

Tip: The exceptions allow you to force event registration even for 2D applications which are not supporting **shared memory event queue API** and normally not being registered in the **Event history** (e.g. CPU stress-testing applications). Also, the exceptions allow you to disable event registration only for some desired 3D applications if necessary.

Remove exception

This button allows you to remove currently selected **event history exception**.

Tip: The exceptions allow you to force event registration even for 2D applications which are not supporting **shared memory event queue API** and normally not being registered in the **Event history** (e.g. CPU stress-testing applications). Also, the exceptions allow you to disable event registration only for some desired 3D applications if necessary.

Voltage identifier interpretation mode

This option allows you to select desired VID interpretation mode and see target voltage instead of raw VID data.

Tips:

• RivaTuner always stores raw VID data when logging monitoring statistics to file. Interpretation is applied during viewing only.
• You may create your own VID interpretation modes by customizing **VIDInterpretation** sections in **RivaTuner.cfg** file.

Autoselect voltage identifier interpretation mode

Click this button to force RivaTuner to scan VID interpretators database and select an interpretator matched with voltage table stored in VGA BIOS.

Tips:

• This button is available only when the database contains proper interpretator for your VGA BIOS.
• You may create your own VID interpretation modes by customizing **VIDInterpretation** sections in **RivaTuner.cfg** file.

Voltage identifier interpretation preview

This window allows you to preview selected voltage interpretation mode and shows how raw VID data is mapped to the corresponding target voltages.

Threshold name

This edit box allows you to enter a name for the threshold. The name is displayed in **Thresholds list** and used for simplifying the threshold's functional purpose identification.

Tip: **OK** button is grayed until you specify non-empty threshold name.

Threshold color

This is a color of the threshold you're currently editing. You may click it to redefine the color.

Tip: If you use multiple thresholds on the same graph, you may use different threshold colors for easier threshold identification.

Threshold value

This edit box allows you to enter threshold value.

Tips: **OK** button is grayed until you specify valid threshold value.

Minimum threshold crossing period

This edit box allows specifying minimum period (in milliseconds) between registering two threshold crossing events. Take a note that RivaTuner uses independent periods for upward and downward threshold crossing events.

Treat missing … as …

This option allows RivaTuner to replace missing data from the source with desired value during threshold crossing condition verification. This can be useful for the sources, reporting no data under certain conditions (e.g. **Framerate** data source reports no data when there are no 3D applications running).

Path for upward threshold crossing event

This edit box allows entering a path to an application to be launched in case of upward threshold crossing event registration.

Command line for upward threshold crossing event

This edit box allows entering command line parameters to an application to be launched in case of upward threshold crossing event registration.

Launcher item for upward threshold crossing event

This dropdown list allows you to select item to be launched in case of upward threshold crossing event registration.

Browse for path for upward threshold crossing event

Press this button to browse for path to an application to be launched in case of upward threshold crossing event registration.

Path for downward threshold crossing event

This edit box allows entering a path to an application to be launched in case of downward threshold crossing event registration.

Command line for downward threshold crossing event

This edit box allows entering command line parameters to an application to be launched in case of downward threshold crossing event registration.

Launcher item for downward threshold crossing event

This dropdown list allows you to select item to be launched in case of downward threshold crossing event registration.

Browse for path for downward threshold crossing event

Press this button to browse for path to an application to be launched in case of downward threshold crossing event registration.

Apply this color scheme at Windows startup

Enabling this option will automatically apply the current color scheme at Windows startup.

Tip: In order to avoid color correction interference don't try to use this option simultaneously with driver-level color correction.

Low-level color adjustment sliders

These sliders allow you to adjust the brightness, contrast and gamma values for each or for all of the RGB channels. These adjustments apply to Direct3D/OpenGL games as well as to the desktop.

Tips:

• These sliders are locked when using **Custom color curve** color control mode.
• Unlike the driver-level color correction, low-level color correction doesn't use any driver specific color correction interface and can work with any **Get / SetDeviceGammaRamp** capable display adapter, or even directly accesses RAMDAC palette registers.
• The adjustments don't have an effect on the applications which perform own gamma correction (e.g. Quake III), until you associate the color scheme with it and use **Lock gamma** option.
• Windows 2000 / XP can fail to set some gamma ramps. Due to this limitation, you won't be able to set certain brightness/contrast/gamma combinations when controlling gamma via **GDI gamma ramp**.

Save color scheme

This button allows you to save the current color adjustments as a color scheme, which will then be added to the adjacent list. You can activate this color scheme later by selecting it in this list.

Tips:

• You can associate a color scheme you have created with a launcher item via the **Launcher** tab.

Color control mode

RivaTuner provides you two different color control modes:

• **Classic brightness / Contrast / Gamma mode** allows you to make usual gamma correction adjustments based upon 3 standard color correction parameters.
• **Custom color curve mode** gives you more freedom and allows drawing custom color curve for each RGB channel, so you can specify gamma ramps which cannot be derived from 3 classic color correction parameters (e.g. color temperature corrected gamma ramp).

Color control level

RivaTuner can control gamma ramp at two different levels. Gamma ramp can be accessed either via **GDI gamma ramp** functions or via **direct access to RAMDAC palette**. It is recommended to use **GDI gamma ramp** level in most of cases.
However, you can also use **direct access to RAMDAC palette** if GDI level gamma correction is not function properly or is not implemented in the driver.
You may also use **direct access to RAMDAC palette** to set gamma ramps, which cannot be set via GDI due to its' visibility validation algorithms (e.g. absolutely inverted gamma ramps).

Tips:

• **Direct access to RAMDAC palette** mode is available on all supported ATI and NVIDIA graphics processors only for displays connected to the primary RAMDAC.

Gamma ramp window

Displays current gamma ramp. It shows how each (R, G and B) or all channels are distributed when you move the **Brightness**, **Contrast** or **Gamma** slider to make your adjustments.
When using **Custom color curve** mode, this window also allows you to edit spline, defining color curve for the currently selected color channel(s).

Tips:

• Take a note that some gamma ramps cannot be accepted by GDI. The yellow exclamation sign in the top-left corner of this window indicates that the color curve you've specified produces such gamma ramp. You may click this sign to reset color curve to the last valid state.
• You may click any point on the color curve to add new spline node. Newly added spline node automatically become current spline node.
• You may click existing spline node to select it. Selected nodes can be deleted by pressing the **Remove spline node** button on the toolbar.
• You may capture and drag spline nodes to adjust color curve.

Lock spline node X coordinate

Use this button to lock spline node X coordinate when dragging it with mouse.

Lock spline node Y coordinate

Use this button to lock spline node Y coordinate when dragging it with mouse.

Snap spline nodes to grid

Use this button to snap spline nodes to grid when dragging them with mouse.

Remove spline node

Click this button to remove currently selected spline node.

Tip: Select spline nodes by clicking it with left mouse button.

Remove spline

Click this button to remove all spline nodes and reset it to default state.

Driver status bar

This status bar displays information about currently installed display drivers.

Tips:

• The **Customize** button appears on this status bar when you select supported display adapter as a tweak target and RivaTuner detects supported display drivers for this adapter. You can click this button to show the **Driver customization toolbar**, which allows customizing the driver's settings. When RivaTuner is minimized to tray you can activate the driver customization toolbar by holding the **Ctrl** button and clicking RivaTuner's tray icon or by holding left mouse button on the tray icon for at least 0.5s.
• The following drivers are currently supported for tweaking:

NVIDIA Detonator v2.08 or higher for Windows 9x
NVIDIA Detonator v5.08 or higher for Windows2000/XP.

Add new profile

Click this button to activate **Application profile editor** dialog, allowing you to add new executable file to the **Profiles list**.

Scan registry for existing profiles

Click this button to scan your registry for application specific settings and append the **Profiles list** with the applications these settings are targeted to.

Edit profile

Click this button to edit currently selected application profile.

Remove profile

Click this button to remove currently selected application from the profiles list. You may either simply remove the executable from the list leaving its' application specific settings in the registry or totally remove application specific settings from the registry forcing the driver to use global settings for it.

Profiles list

ForceWare 56.xx and newer drivers support flexible profiles technology, allowing you to specify personal Direct3D and OpenGL settings for each executable file. This list allows you to use this technology and switch between **Global driver settings** and **Appplication specific settings** modes.
When using **Global driver settings**, the options you change in Direct3D and OpenGL tweaking dialogs globally affect all applications.
When you use **Appplication specific settings** and select target application in this list, you can optionally make some Direct3D / OpenGL options specific only to the selected application.

Tips:

• You may append this list using **Add new profile** and **Scan registry for existing profiles** buttons.
• You may remove currently selected application from this list using **Remove profile** buttons.

Driver files

This is a list of files and associated version information which are in use by the currently installed display drivers. The list is empty if there are no supported display drivers detected for the current display adapter.

Tips:

• The following drivers are currently supported:

NVIDIA Detonator v2.08 or higher for Windows 9x
NVIDIA Detonator v5.08 or higher for Windows2000/XP.

Send to tray on close

This allows you to minimize RivaTuner to the tray instead of closing it when you press either **Close** or **OK** buttons.

Tip: You can always close RivaTuner regardless of this option by holding **Ctrl** and pressing **Close** or **OK**.

Interface layout

This option allows you to select a preferred interface layout. The **classic interface** and **skinned background** layouts use Windows 9x styled controls whilst the **skinned controls** mode allows you to use Windows XP themed interface.

Always on top

This option allows RivaTuner to appear always on top. This feature is useful when adjusting overlay color scheme and watching a movie at the same time.

Run at Windows startup

This option allows you to start RivaTuner automatically with Windows.

Startup mode

This allows you to select RivaTuner's startup mode. There are two modes available:

• **Via Startup registry key** – the link to RivaTuner is added to the startup registry key [HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run].
• **Via Startup folder** – the shortcut is added to the Startup folder. Use this mode if you are using Yamaha YMF724 or Yamaha Sondius S-YSG100 and RivaTuner doesn't properly start via the startup registry key (this problem can be caused by the incompatibility between the Detonator and Yamaha drivers).

Tip: This option is disabled on Windows 95 if Microsoft Internet Explorer 4.0 Desktop Update is not installed.

Emulate hotkeys handler every … ms

RivaTuner uses native Windows hotkey handler by default. This works flawlessly under Windows 2000 / XP, but under Windows 9x almost all the keyboard related messages are suppressed when any DirectInput application use the keyboard in exclusive mode. This results in ignoring RivaTuner's hotkeys when the most of games are running.
You may use this setting to force RivaTuner to emulate hotkeys handler and use hotkeys with such applications. When emulation mode is activated RivaTuner scans the keyboard within specified time intervals and generates required hotkey related messages if it is necessary.

Tip: RivaTuner doesn't check up crosslinked hotkeys in emulation mode.

Emulate hotkeys handler via low-level keyboard hook

RivaTuner uses native Windows hotkey handler by default. This works flawlessly under Windows 2000 / XP, but under Windows 9x almost all the keyboard related messages are suppressed when any DirectInput application use the keyboard in exclusive mode. This results in ignoring RivaTuner's hotkeys when the most of games are running.
You may use this setting to force RivaTuner to emulate hotkeys handler and use hotkeys with such applications. When emulation mode is activated RivaTuner installs low-level keyboard hook in system, monitors all keyboard related activity at low level, and generates required hotkey related messages if it is necessary.

Tip: RivaTuner doesn't check up crosslinked hotkeys in emulation mode.

Current database

Displays the path to currently loaded *.rtd database file. You can capture this window with mouse cursor and drag it to scroll window if the path doesn't fit in it.

Built-in registry editor

This is an advanced built-in registry editor. It can easily add, edit, remove or track changes into any specified registry entries. The list of registry entries, which can be edited via this built-in editor, is stored into special file called a database. Database files contain default values and descriptions for all registry entries. This editor can be also used to save and load the state of some registry entries. The state of registry entries is stored into special file called a preset.

Tips:

• Use **0**-**9** keys to set a selected entry to a corresponding numeric value. It works with string values as well, as with DWORD and binary values.
• You can either press **Enter** or double click on any registry entry to edit it. You can use spin buttons to increase or decrease DWORD and binary values in edit mode.
• You can click right mouse button on any registry entry to display the **context toolbar**. Some useful actions can be performed via its' buttons.
• You can refresh the current database with the **F5** key. All values which have been changed will be marked with red exclamation sign. RivaTuner automatically refreshes the database when you switch to the editor from another application or from another tab.
• You can select entries by clicking to the left of its' name. The state of selected entries can be stored into a preset.

Open database

This button opens a new database, stored into *.rtd file.

Tip: You don't make any changes in the registry when opening a new database. When opening a database RivaTuner just loads the list of registry entries, which can be viewed and edited via the built-in registry editor.

Open matched Detonator database

This button allows you to open the best matched database for currently installed Detonator drivers automatically.

Tips:

• This button is inactive if you have not selected NVIDIA-based device as target adapter. This button is also inactive if you have no Detonator drivers installed.
• You can hold **Ctrl** key and press this button to open default database, which allows RivaTuner to tweak itself.

Open patch script

This button allows you to open and execute a patch script, stored into *.rts file. Patch scripts can perform masked search and replace functions in specified files, create backups, regenerate CRC for Windows NT4/2000 drivers, use file version and (or) size dependent patch scripts and so on. You can easily create your own patch scripts or use RivaTuner's NVIDIA related scripts.

Tips:

• Patch scripts are located in the **.\PatchScripts** subfolder.
• Don't be afraid to press this button. RivaTuner will not execute script without your permission so you may safely read detailed description for each patch script.

Merge preset

This button allows to merge a preset with current database. It loads a preset, scans all the preset entries and only handles those entries, which are found in currently loaded database. RivaTuner adds such entries to the registry and also selects it in current database. The rest of the entries are ignored. This button is useful when combining two or more presets.

Tips:

• Presets are located in the **.\Presets** subfolder.
• This function is intended for preset editing, don't use it to load a preset. To load the preset use the **Launcher** tab.

Save selected values to preset

This button allows you to save selected registry entries to *.rtp preset file. The preset file format is similar to *.reg file, but it has some advantages:

• Both Windows9x and Windows2000 store the driver's settings in system dependent registry keys. It means that your *.reg files may work incorrectly on another system or even on your system after reinstalling display drivers. RivaTuner uses macro names to export and import presets so it always works correctly.
• *.reg files cannot remove entries from the registry. Presets can do it.

Tips:

• Presets are located in **.\Presets** subfolder.
• You can select entries by clicking to the left of its' name. You can use the **Select all values** and **Unselect all values** buttons to perform selection related operations.
• You can hold the **Ctrl** key before pressing this button to create a *.reg file instead of preset.

Hexadecimal display

This button switches between decimal and hexadecimal view.

Protect database entries

Some registry entries can be marked as erase protected in the registry database. Such entries are added to the registry by the driver's installer and they must always exist in the registry. RivaTuner doesn't remove such entries from the registry when you use the **Reset all the driver's settings to their default values** function. You can press this button to highlight all erase protected entries in the built-in registry editor. RivaTuner will warn you that the entry is erase protected if you will try to remove it from the built-in editor. Erase protected entries are treated as usual registry entries if this button is not pressed.

Select all entries

This button allows you to select all the registry entries in currently loaded database.
Selected entries can be saved in preset.

Tips:

• You can select entries by clicking to the left of its' name.
• You can hold **Ctrl** while pressing this button to select entries into the current root key only. This feature is useful when creating preset from Direct3D or OpenGL entries only.
• You can hold **Shift** while pressing this button to select only existing entries. Take a note, that you can use **Ctrl** and **Shift** modifiers simultaneously.

Find next selected entry

This button allows you to find the next selected entry starting from the currently highlighted entry.

Tip: This button is grayed if there are no selected entries in the database.

Unselect all entries

This button allows to unselect all the registry entries in currently loaded database.

Tips:

• You can unselect selected entries by clicking selection marker.
• You can hold **Ctrl** before pressing this button to unselect entries into the current root key only.

Unmark updated entries

This button allows you to unmark all updated entries. RivaTuner marks an entry as updated when it detects that it was changed outside the built-in registry editor. Such registry entries are marked with red exclamation signs. RivaTuner unmarks the updated entry when you select it.

Tips:

• RivaTuner scans the database for the updated entries on each refresh operation. It automatically refreshes currently loaded database when you are switching to the editor from another application. You can refresh the currently loaded database by pressing the **F5** key.
• You can hold **Ctrl** before pressing this button to unmark updated values into current root key only.

Find next updated entry

This button allows you to find the next updated entry starting from the currently highlighted entry.

Tip: This button is grayed if there are no updated entries in the database.

Database status

Displays a status of currently loaded database. It includes the total number of registry entries in this database, the number of selected entries and the number of updated entries.

Tips:

• You can select entries by clicking to the left of its' name. The state of selected registry entries can be stored into a preset.
• RivaTuner marks an entry as updated when it detects that it was changed outside the built-in editor. Such entries are marked with red exclamation signs.

Launcher item status

Displays a status of currently selected launcher item. It allows to see what have you associated with it.

Launcher items list

This is the list of launcher items. Press **Add new item** button to create new launcher item.

Tips:

• You can double click a launcher item to run it. You can also run the launcher items by pressing an associated hotkey or via the context menu when RivaTuner is minimized to tray.
• You can drag and drop launcher items to rearrange them.
• You can click right mouse button on any item to display the **context toolbar**. Some useful actions can be performed via its' buttons.

Add new item

This button allows you to add a new launcher item. Each launcher item is just a menu item. You can associate a preset, low-level overclocking profile, display mode, low-level / driver-level color scheme and application with each launcher item.

Enable submenu

Press this button to allow launcher menu items to be displayed in separate submenu in tray menu. Otherwise menu items will be displayed in root of tray menu.

Regular menu item type

Regular menu items can be used for launching applications and (or) associated presets, low-level overclocking profiles, display modes and low-level / driver-level color schemes. You can also use blank regular menu items as separators to improve menu readability.

Tips: Once created, menu item type cannot be changed.

RivaTuner module activation menu item type

You can create RivaTuner module activation menu items for quick access to the most frequently used modules of RivaTuner. Once created module activation item, simply select it from tray menu or use associated hotkey to get to selected module immediately.

Tips: Once created, menu item type cannot be changed.

About RivaTuner

Displays information about RivaTuner and its' author.

NVIDIA World logo

Click it to visit NVIDIA World website.

Vacant place for your advertisement

Want to see your advertisement here? Contact me.

Author’s email

Click it to make contact with me via email.

Author’s ICQ

Click it to make contact with me via ICQ.

Application specific settings list

This is the list of settings which can be associated with the currently selected application profile. If the setting is selected, it's included in the application profile. Otherwise it's global for all applications.

Tips:

- You can select a setting by clicking to the left of its' name.
- Some settings can be global only. If the setting cannot be associated with an application, you will not find it in this list.

Mipmap LOD bias adjustment slider

This allows you to adjust the level of detail bias for mipmaps. Level of detail (LOD) bias adjustments make a mipmap appear crisper or less sharp than it normally would. Each unit of LOD bias alters the selection by exactly one mipmap level. A negative bias causes the use of larger levels, the result is a sharper but more aliased image. A positive LOD bias causes the use of smaller mipmap levels, the result is a more blurred image. A positive bias also causes less texels to be referenced, which can boost performance on some systems.

Tips:

• Direct3D driver can adjust LOD bias in [–15.0;+15.0] range, but in RivaTuner it is limited to [-3.0;+3.0] in order to simplify LOD bias scale calibration. You can change it from [-1.0;+1.0] till [-15.0;+15.0] with the **LODBiasRange** registry entry.
• Different versions of the Detonator drivers treat LOD bias in the different ways. Detonator 2.xx has lower relative LOD bias than the rest versions of the Detonator drivers. The value 0.0 in the Detonator 2.xx is equal to 0.5 in the rest versions of drivers.
• Different versions of the drivers use different strategies when a Direct3D application specifies its' own LOD bias via the Direct3D API. In this case the Detonator 2.xx ignore user defined bias and the rest versions add this value to user defined bias.
• Due to the Detonator's bug you will not be able to set negative LOD bias correctly on the Detonator 23.10 and higher. However, you may use RivaTuner's **LODBiasFix** patch script in order to fix it and allow the driver to set negative LOD bias correctly.

Clamp negative LOD bias

Enabling this option allows the driver to prevent Direct3D applications from using negative LOD bias in conjunction with anisotropic filtering. This helps to avoid unwanted texture aliasing or shimmering.

Current mipmap LOD bias

This is the current mipmap LOD bias. You can use the LOD bias slider to adjust it.

Minimum mipmap LOD bias

This is the minimum mipmap LOD bias you can set.

Tip: The drivers can adjust LOD bias in [–15.0;+15.0] range, but in RivaTuner it is limited to [-3.0;+3.0] in order to simplify LOD bias scale calibration. You can change it from [-1.0;+1.0] till [-15.0;+15.0] with the **LODBiasRange** registry entry.

Maximum mipmap LOD bias

This is the maximum mipmap LOD bias you can set.

Tip: The drivers can adjust LOD bias in [–15.0;+15.0] range, but in RivaTuner it is limited to [-3.0;+3.0] in order to simplify LOD bias scale calibration. You can change it from [-1.0;+1.0] till [-15.0;+15.0] with the **LODBiasRange** registry entry.

Enable user mipmaps

This option allows the Direct3D driver to utilize user (application defined) mipmap levels. Disabling this option doesn't prevent the driver from creating the DDCAPS_MIPMAP surfaces, but forces it to ignore user mipmap levels during texture filtering.
Disabling user mipmap levels dramatically increases fillrate stress, therefore and it can cause the major performance drop, especially on the weak graphics processors like TNT/TNT2. Sometimes it can improve image quality and make the textures appear more crisper.

Tip: Always keep this option enabled. Use a negative LOD bias to improve image sharpness.

Allow mipmap dithering

TNT/TNT2 graphics processors cannot perform strict trilinear filtering in multitexturing mode. In this case the driver performs mipmap dithering instead of strict color blending.
This setting affects the IDirect3DDevice's ValidateDevice method, which is frequently used by the Direct3D applications to check the device's ability to render the current texture blending operations and arguments in a single pass. By default the driver doesn't validate the device on NV4 and NV5 boards when multitexturing mode is used in conjunction with trilinear filtering. This prevents these boards from mipmap dithering and forces either strict trilinear filtering and single texturing mode or multitexturing mode in conjunction with bilinear filtering depending on the application's preferences. When this setting is enabled the driver is allowed to validate the device even when the application requests multitexturing in conjunction with trilinear filtering. It may help to enable trilinear filtering in certain applications and improve image quality.

Tip: This option is active on TNT/TNT2 graphics processors only when the Detonator 3.xx or higher is installed.

Number of automatic mipmap levels

TNT/TNT2 graphics processors can automatically generate up to 12 mipmap levels for textures. It can improve performance in old applications, which has no own mipmapping support. The greater number of automatic mipmap levels means higher performance but the lower amount of free videomemory.
Automatic mipmapping can often make in-game head-up-displays (HUD) and fonts distorted/blurred. You may decrease the number of automatic mipmap levels or disable automatic mipmapping at all if you are experiencing such problems.

Tip: GeForce256 and higher GPUs do not support automatic mipmapping. Automatic mipmapping is also not supported by the Detonator 6.xx and higher.

Automatic mipmap filtering mode

This allows to select filtering mode for automatic mipmap levels. Bilinear filtering will provide better performance, whilst trilinear filtering will increase image quality in the cost of some performance.

Tip: GeForce256 and higher GPUs do not support automatic mipmapping. Automatic mipmapping is also not supported by Detonator 6.xx and higher.

Enable trilinear optimization

This option allows you to control trilinear filtering optimization. When this option is enabled, the driver is allowed to reduce trilinear filtering quality for better filtering performance depending on currently active Intellisample mode.

Tip: The driver automatically disables all optimizations if **High Quality** Intellisample mode is used.

Enable anisotropic mip filter optimization

This option allows you to control anisotropic mip filter optimization. When this option is enabled, the driver forces point mip filter on all but the primary texturing stages if anisotropic filtering is used.

Tip: The driver automatically disables all optimizations if **High Quality** Intellisample mode is used.

Enable anisotropic sample optimization

This option allows you to control anisotropic sample optimization. When this option is enabled, the driver is allowed to use a variety of anisotropic sample optimizations on all but the primary texturing stages. The variety of optimizations to be used depends on currently active Intellisample mode.

Tip: The driver automatically disables all optimizations if **High Quality** Intellisample mode is used.

Enable 24-bit Z-buffer

This setting allows the Direct3D driver to report 24-bit Z-buffering capability. It gives better image quality than 16-bit in cost of some performance drop. Usually the Direct3D applications check the driver's capabilities so disabling 24-bit Z-buffering forces such applications to use 16-bit Z-buffer.

Tips:

- Some applications cannot run when 24-bit Z-buffer is disabled.
- Windows 2000/XP must be restarted after changing this option.
- This setting doesn't have an effect on DirectX 8 applications.

Z-buffer validation method

This allows you to select Z-buffer validation method. The Direct3D driver performs validation when application is trying to attach a Z-buffer to a render target surface.
You can select one of the following Z-buffer validation methods:

• **Always** mode allows the driver to validate Z-buffer always, even if its' bit depth is not equal to the frame buffer's bit depth. In this case application will be able to run in 32-bit modes with 16-bit Z-buffer and vice versa.
• **When equal to render depth** allows the driver to validate Z-buffer only if its' bit depth is equal to the frame buffer's bit depth. The driver will fail on the attempt to attach 16-bit Z-buffer to 32-bit surface and vice versa if this mode is selected. This mode can be used by software developers for testing purposes.

Tips:

• Some graphics processors cannot use 24-bit Z-buffer in 16-bit modes in DirectX7 and DirectX8 applications.
• This option is inactive if the Detonator 21.xx or higher is installed.

Enable W-buffer

This option allows Direct3D driver to report W-buffer capability. W-buffer is a kind of depth buffer, which is using as an alternative to Z-buffer.
Due to the mathematics involved, the generated Z values in a Z-buffer tend not to be distributed evenly across the Z-buffer range, so this can cause hidden surface artifacts in distant objects, especially when using 16-bit Z-buffers.
W-buffer is more evenly distributed between the near and far clip planes than a Z-buffer. It allows the Direct3D applications to support large maximum ranges and still getting relatively accurate depth buffering close to the eye point. W-buffer is not perfect, and can sometimes exhibit hidden surface artifacts for near objects.
Usually applications (e.g. Messiah) check the driver's capabilities and allow user to select a depth buffer type. However, some applications (e.g. Unreal engine based games) don't allow to select a depth buffer type and always use W-buffer if Direct3D driver reports W-buffer capacity. You can disable W-buffer to force such applications to use Z-buffer.

Tips:

• Unreal engine based games may exhibit hidden surface artifacts on weapon models when using W-buffer in 16-bit modes. You can disable it to remove these artifacts, but it will sometimes cause Z-buffer artifacts on the distant objects. TNT/TNT2 owners can also select 16-bit floating point W-buffer in order to fix these artifacts without causing Z-buffer artifacts on the distant objects.
• Windows 2000/XP must be restarted after changing this option.
• This setting doesn't have an effect on DirectX 8 applications.

W-buffer format

There are two ways to represent the values stored in W-buffer: scaled integer and floating point. Integer format is a bit faster, but floating point is more accurate and can give better image quality.

Tips:

• Unreal engine based games may exhibit hidden surface artifacts on weapon models when using W-buffer in 16-bit modes. You can select 16-bit floating point W-buffer format in order to improve depth buffering quality.
• Due to the driver's limitation GeForce256 and higher GPUs can only use scaled integer W-buffer so W-buffer format options have an effect and can be changed on TNT/TNT2 boards only.

Enable lossless Z-buffer compression

This option allows to enable lossless Z-buffer compression. The Z-buffer represents the depth or visibility information for the pixels ultimately to be displayed after being rendered. All the traditional graphics processors read and potentially write Z data for every pixel they render. This makes the Z-buffer traffic one of the largest "consumers" of memory bandwidth in a graphics system.
By implementing the 4:1 lossless data compression the memory bandwidth consumed by Z-buffer traffic is reduced by a factor of four. It is implemented in hardware transparently to applications, with both compression and decompression taking place in real time by the LMA's Z-compression and decompression engines.
Because Z-buffer compression is completely lossless there is no reduction in image quality. The result of the use of this technology is a dramatically more efficient use of memory bandwidth for dramatically improved performance with no compromise in image quality.

Tip: Always keep this option enabled. Disable it for testing purposes only.

Override pixel shader version

This option allows you to override the maximum supported pixel shader version. You may use it for troubleshooting and performance comparison.

Tip: The value you can set is limited with the maximum shader version supported by your hardware.

Override vertex shader version

This option allows you to override the maximum supported vertex shader version. You may use it for troubleshooting and performance comparison.

Tip: The value you can set is limited with the maximum shader version supported by your hardware.

Enable early Z-occlusion culling

Almost all the traditional graphics architectures render every pixel of every triangle as it receives them, accessing the frame buffer with each pixel to determine the appropriate values for the color and Z for each of those pixels. This method produces correct results, but requires all of the pixels to be rendered, regardless of their visibility or not.
Typical content today has an average depth complexity (overdraw ratio) of two. It means that for every pixel that ends up being visible, two pixels have to be rendered to come up with that result. It means that for every visible pixel, the GPU is forced to access the frame buffer twice, spending valuable frame buffer bandwidth essentially rendering pixels that the viewer will never see.
GeForce3 implements a Z-occlusion culling technology, whereby it attempts to determine early if a pixel is going to be visible. If it is going to be occluded and the Z-occlusion culling unit determines this, the pixel is not rendered, the frame buffer is not accessed, and the frame buffer bandwidth is saved. Depending on the overdraw ratio of the scene this can mean tremendous improvements in efficiency.
This option allows to enable early Z-occlusion culling.

Tips:

• Always keep this option enabled. Disable it for testing purposes only.
• This option is only active on GeForce3 and higher boards when the Detonator 12.60 – 27.xx, 40.xx or 41.80-43.xx is installed.

Enable wait after blit when DDBLT_WAIT flag is specified

This option allows the DirectDraw driver to handle DDBLT_WAIT flag. According to the DirectDraw specifications, an application may specify this flag when calling blitting functions. In this case, any blitting function must not return until the end of blitting.
NVIDIA DirectDraw driver never waits the end of blitting by default and always returns immediately, even when DDBLT_WAIT flag is specified. It improves performance, but may cause problems with some applications. You can enable this option in order to force the driver to work according to the DirectDraw specifications.

Tips:

• Blitting settings are useful for the games that cannot blit correctly. All these settings will reduce performance so it is recommended to use it only if you have problems with blitting.
The term blitting is shorthand for "bit block transfer", which is a process of copying blocks of data from one place in memory to another.
All the modern graphics processors can perform blitting in hardware. It is widely used in old games to perform sprite related operations. It is also used in 3D games to transfer a data from one texture to another.
• This option is only active on the Detonator 5.xx and higher.

Always wait after blit

This option forces the DirectDraw driver to wait after blit always regardless of DDBLT_WAIT flag. According to the DirectDraw specifications, an application may specify this flag when it is calling blitting functions. In this case any blitting function must not return until the end of blitting.
This option forces this flag to be set for all blits, which prevents the games that do not check the return value from unexpectedly losing blits.

Tips:

• Blitting settings are useful for the games that cannot blit correctly. All these settings will reduce performance so it is recommended to use it only if you have problems with blitting.
The term blitting is shorthand for "bit block transfer", which is a process of copying blocks of data from one place in memory to another.
All the modern graphics processors can perform blitting in hardware. It is widely used in old games to perform sprite related operations. It is also used in 3D games to transfer a data from one texture to another.
• This option is only active on the Detonator 6.xx and higher.

Limit the maximum number of queued blits to the framebuffer

You can use this option to limit the maximum number of queued blits to the front buffer to prerender limit value (you can adjust this value in the **VSync** tab).
NVIDIA graphics processors can perform queued blitting. This means that blitting calls will always be succeeded, even if hardware blitter is currently busy. In this case requests will be queued and performed when hardware will be able to do it.
Some applications (especially windowed) constantly blit to the front buffer. Such applications can overflow the blitting queue and it can cause unpredictable results.
You can use this option in order to prevent the blitting queue from overflowing.

Tips:

• Blitting settings are useful for the games that cannot blit correctly. All these settings will reduce performance so it is recommended to use it only if you have problems with blitting.
The term blitting is shorthand for "bit block transfer", which is a process of copying blocks of data from one place in memory to another.
All the modern graphics processors can perform blitting in hardware. It is widely used in old games to perform sprite related operations. It is also used in 3D games to transfer a data from one texture to another.
• This option is only active on the Detonator 6.xx and higher.

VSync mode

This option allows to control VSync mode. The term VSync is shorthand for “synchronization with vertical retrace” or in other words synchronization with the monitor’s refresh rate.
Usually the applications perform VSync when flipping the back and front buffers so enabling Vsync limits the framerate to the monitor’s refresh rate.
Enabled VSync can cause perfectly smooth animation if an application’s framerate is higher then the monitor’s refresh rate. Disabled VSync can boost performance but it can cause visual tearing effect. When synchronization with vertical retrace is disabled monitor can start to draw a frame even when the previous frame is not finished. In this case you can see a piece of the previous frame in the top of the screen.

You can choose one of the following VSync modes:

- **Auto** – VSync is enabled by default and can be controlled by application.
- **Always on** – VSync is enabled in all Direct3D applications
- **Always off** – VSync is disabled in all Direct3D applications

Tip: Some applications (e.g. Unreal engine based games) cannot disable VSync correctly when the **Auto** mode is selected. You can use the **Always off** mode to disable VSync in such applications.

Disable VSync

This option allows to control VSync mode. The term VSync is shorthand for “synchronization with vertical retrace” or in other words synchronization with the monitor’s refresh rate.
Usually the applications perform VSync when flipping the back and front buffers so enabling Vsync limits the framerate to the monitor’s refresh rate.
Enabled VSync can cause perfectly smooth animation if an application’s framerate is higher then the monitor’s refresh rate. Disabled VSync can boost performance but it can cause visual tearing effect. When synchronization with vertical retrace is disabled monitor can start to draw a frame even when the previous frame is not finished. In this case you can see a piece of the previous frame in the top of the screen.

Prerender limit

This allows to limit the number of frames which the CPU can prepare before they are processed by the graphics hardware. The greater prerender limit can improve performance, but it can cause input lag effect. Reduce the prerender limit if you experience a noticeable delay in a response to the input devices while you playing games.

Amount of system memory for PCI textures

This allows you to control amount of system memory for texture storage. The maximum amount of memory that can be reserved is equal to the half of system RAM installed in your computer.

Tip: This option only applies PCI display adapters or AGP display adapters running in PCI mode.

Enable DXTC texture compression

This setting allows the Direct3D driver to report DXT1 - DXT5 texture compression formats support. Disabling this option will prevent all the Direct3D applications from using texture compression. It can reduce performance but can improve image quality in the applications which exhibit the DXT1 quality problem.

Tips:

• DXT1 quality is a weak spot of pre-GeForce4 graphics processors. Hardware DXT1 decompressor uses 16-bit texel color interpolation. This often cause banding effects on the textures, which contain smooth color gradients (e.g. sky textures in Quake III). The rest compression formats use 32-bit texel interpolation and don't exhibit the banding effect.
• You can also fix DXT1 quality problem with RivaTuner's built in **S3TCFixPack** patch scripts.
• This option is only active on GeForce 256 and higher GPUs if the Detonator 5.xx or higher is installed.
• Windows 2000/XP must be restarted after changing this option.

Enable 8-bit palettized textures

This setting allows the Direct3D driver to report 8-bit palletized texture format support. Disabling this setting prevents all the Direct3D applications from using 8-bit palletized textures.
It can solve incompatibility problems with some applications, which use 8-bit textures incorrectly.

Tips:

• This option is only active on GeForce 256 and higher GPUs and only if the Detonator 10.xx or higher is installed.
• Windows 2000/XP must be restarted after changing this option.

Texture formats

This list allows you to control surface formats supported by the Direct3D7/Direct3D8+ driver.
You may disable FOURCC DXT1 - DXT5 texture formats support in order to prevent all the Direct3D applications from using the texture compression. It will reduce performance but may improve image quality in the applications which exhibit DXT1 quality problem.
It is strongly recommended to leave the rest surface formats unchanged.

Tips:

• The list of available surface formats depends on your graphics hardware.
• DXT1 quality is a weak spot of pre-GeForce4 graphics processors. Hardware DXT1 decompressor uses 16-bit texel color interpolation. This often cause banding effects on the textures, which contain smooth color gradients (e.g. sky textures in Quake III). The rest compression formats use 32-bit texel interpolation and don't exhibit the banding effect.
• You can disable 8-bit palettized textures support for DirectX8 in order to avoid font related problems in GTA III under Windows XP.
• The system must be restarted after changing surface format settings.

Texture formats API

This allows you to control surface formats separately for DirectX7 and DirectX8+ applications.

Degree of anisotropy

This setting allows to force the Direct3D driver to use anisotropic texture filtering regardless of the application's request.
Anisotropy is a distortion visible in the texels of any 3D object whose surface is oriented at an angle with respect to the plane of the screen. If a pixel from such surface is mapped to texels, its shape is distorted. Anisotropic filtering is the most advanced texture filtering technique, that compensates for anisotropic pixel distortion. This technique gives the best image quality, especially when it's combined with trilinear filtering.
The Direct3D applications specify the degree of anisotropy to enable anisotropic filtering. Anisotropic texture filtering can be disabled by setting the degree of anisotropy to one, any value larger than one enables it. The greater degree of anisotropy means higher image quality but the lower performance.

Tips:

• The maximum degree of anisotropy depends on your hardware installed and the driver's version. It is limited to **level 2** on all pre-GeForce3 boards and on GeForce3 boards if Detonator 10.xx-11.xx is installed.
• This setting only overrides the texture minification filter.
• This setting is only active on GeForce 256 and higher and only if the Detonator 10.xx or higher is installed.

Optimize

This button allows to optimize anisotropic filtering performance by controlling the maximum degree of anisotropy for specified texturing stages.

Tips:

• This option is active on NV2x based boards only when the Detonator 28.32 – 4x.xx is installed. By default the optimization settings work on NV25 GPUs only, you must use **AnisoBoosterD3D** patch script in order to unlock optimization settings on the rest GPUs.

Texturing stages list

The most of modern games use two and more texture layers (texturing stages in terminology of the Direct3D API) in their 3D engines. Forcing anisotropic filtering in such applications causes the driver to use anisotropic minification filter for each texturing stage. However, the most of applications use only one texturing stage to filter the primary(base) texture, and the rest stages are used to filter the secondary textures, which have no major impact on resulting image sharpness (e.g. lightmaps). Reducing the degree of anisotropy for such texturing stages is a good way of boosting anisotropy performance with no major image quality impact and preventing the GPU from wasting its' fillrate.

This list allows you to select the texturing stages, which will be optimized when an application use anisotropic filtering. Once selected at least one target texturing stage for optimization, you may limit the maximum degree of anisotropy for it as well, as disable trilinear filtering.

Always optimize selected stages

Check this option to force the Direct3D driver to use anisotropic filtering optimization for all selected texturing stages always, otherwise the driver will try to detect appropriate optimization conditions and optimize selected texturing stages when it is necessary. You must select at least one texturing stage in order to get access to this option.

The maximum anisotropy for optimized stages

This allows you to limit the maximum degree of anisotropy for optimized texturing stages. You must select at least one texturing stage in order to get access to this option.

Disable trilinear filtering for optimized stages

This allows to disable trilinear filtering in conjunction with anisotropy for optimized texturing stages. You must select at least one texturing stage in order to get access to this option.

Texel origin adjustment slider

This slider allows you to change hardware texture addressing scheme and adjust a texel origin. The Direct3D applications often need to apply textures to geometry in a scene so that texels map directly to on-screen pixels. For example, take an application that needs to display text within a texture on an object within a scene. In order to clearly display textual information in a texture, Direct3D applications need some way to ensure that the textured geometry receives texels undisrupted by texture filtering. Failing this, the resulting image is often blurred, or in case of nearest point sampling, can cause rough edges.
The Direct3D performs the following computations to map floating point texture coordinates to texel addresses:

Tx = u * Mx – 0.5
Ty = v * My – 0.5

In these formulas, **Tx** and **Ty** are output texel coordinates, and **u** and **v** are the horizontal and vertical texture coordinates supplied for the vertex. The **Mx**/**My** elements represent the number of horizontal or vertical texels at the current mipmap level. The reminders define the texel origin.
Some applications (e.g. Need For Speed series) use incorrect texture addressing schemes, which don't conform the Direct3D specifications.
You can change the texel origin in order to improve image quality in such applications.

Tips:

• You can set the texel origin to the top left corner of texel (7) in order to fix blurred fonts in the Need For Speed series.
• Due to the driver's limitation the texel alignment cannot be adjusted on GeForce3 and higher.

Current texel origin

This is the current texel origin. You can use the **Texel origin adjustment slider** to change it.

Enable table fog emulation

This setting allows the Direct3D driver to report pixel/table fog capability. According to the Direct3D specifications any Direct3D hardware accelerator can provide either vertex or pixel fog. TNT/TNT2 graphics processors support vertex fog in hardware. Table fog on these boards is a bit slower because it is emulated via the vertex fog. Usually the applications check up the D3D driver's capabilities and select supported fog type. Disabling this option forces such applications to use the vertex fog and improves performance a bit.

Tips:

• Some applications (e.g. Need For Speed series) use table fog even if the Direct3D driver doesn't report table fog support. The result, in this case, is the graphics almost entirely drawn in fog color. Try to enable table fog if you see such visual artifacts in any Direct3D application.
• GeForce 256 and higher owners may always keep this option enabled. Table fog is implemented in hardware on GeForce 256 and higher graphics processors.
• Windows 2000/XP must be restarted after changing this option.
• This setting doesn't have an effect on DirectX 8 applications.

Show NVIDIA logo when running Direct3D applications

Turning this on shows the NVIDIA logo in the bottom right corner of the screen when running any Direct3D application.

Tip: Always keep this option disabled. It is absolutely useless and it reduces performance a bit.

Enable antialiasing

This allows the Direct3D driver to report FSAA support. It allows some applications (e.g. Re-Volt) to control FSAA manually.

Tips:

• The term FSAA is shorthand for “full-scene antialiasing” , which is a technique that can be used to reduce the appearance of jaggies (stair-step pixels, aliasing artifacts) when drawing any line that is not exactly horizontal or vertical. In 3D scenes , these artifact are most noticeable on the boundaries between polygons.
Antialiasing improves image quality by reducing or eliminating the jaggies but it may cause noticeable performance drop.
• FSAA can often make in-game head-up-displays (HUDs) and fonts appear blurry. Try to disable it if you see abnormally blurred HUD or font.
• Windows 2000/XP must be restarted after changing this option.
• This setting doesn’t have an effect on DirectX 8 applications.

Antialiasing method

This allows to select the antialiasing method. **1 x 2** and **2x** give the best performance while the rest give better image quality at the expense of some performance drop.

Tip: The list of available antialiasing methods depends on the hardware installed in your system and the version of the Detonator drivers.

Force antialiasing in all Direct3D applications

This allows to force all the Direct3D applications to use antialiasing. Some applications which have no own antialiasing support may exhibit rendering artifacts when this option is enabled.

Enable multisample masking

Disabling this option forces the driver to report the D3DPRASTERCAPS_STRETCHBLTMULTISAMPLE capability. This capability bit indicates that the driver provides limited multisampling support through a stretch-blt implementation and prevents the applications from turning multisampling on/off in the middle of a scene as well, as disables multisample masking.

Tips:

• Multisample masking is the Direct3D feature, which allows an application to control write enables for each of the samples when rendering to a multisample buffer. This enables use of a multisample buffer as an accumulation buffer, doing multipass rendering of geometry where each pass updates a subset of samples.
• Avoid enabling multisample masking for pre-DirectX8 applications, it will prevent the driver from using multisampling.
• This setting is only active on GeForce3 and higher GPUs on pre-42.51 Detonators.

Transparency antialiasing

This option allows you to enable transparency antialiasing. Transparency antialiasing is advanced technique, aimed to reduce aliasing on transparent textures (e.g. vegetation, fences). There are two transparency antialiasing modes available:

- **Multisampling** – gives compromised image quality and performance.
- **Supersampling** – gives maximum image quality.

Tip: This option is only available on NV47 GPUs when the ForceWare 71.xx or newer is installed.

Enable gamma correction

This option allows you to enable gamma correct antialiasing.

Tip: This option is only available on NV47 GPUs when the ForceWare 71.xx or newer is installed.

Enable texture sharpening

This option allows the Direct3D driver to double the current degree of anisotropy when FSAA is enabled. It may help to increase image sharpness and reduce blurring side effect, especially in conjunction with SSAA and **Quincunx** antialiasing modes.

Tips:

• This setting is only active on GeForce256 or higher GPUs when the Detonator 40.xx and higher is installed.
• Due to bugs in the currently available betas of the Detonator 4x.xx this option doesn't work as supposed and always double the current degree of anisotropy regardless of FSAA usage. It is strongly recommended to avoid using texture sharpening with these driver revisions.

Mipmap LOD bias adjustment slider

This allows you to adjust the level of detail bias for mipmaps. Level of detail (LOD) bias adjustments make a mipmap appear crisper or less sharp than it normally would. Each unit of LOD bias alters the selection by exactly one mipmap level. A negative bias causes the use of larger levels, the result is a sharper but more aliased image. A positive LOD bias causes the use of smaller mipmap levels, the result is a more blurred image. A positive bias also causes less texels to be referenced, which can boost performance on some systems.

Tips:

• OpenGL driver can adjust LOD bias in [–15.0;+15.0] range, but in RivaTuner it is limited to [-3.0;+3.0] in order to simplify LOD bias scale calibration. You can change it from [-1.0;+1.0] till [-15.0;+15.0] with the **LODBiasRange** registry entry.

VSync mode

This option allows to control VSync mode. The term VSync is shorthand for “synchronization with vertical retrace” or in other words synchronization with the monitor’s refresh rate.
Usually the applications perform VSync when flipping the back and front buffers so enabling Vsync limits the framerate to the monitor’s refresh rate.
Enabled VSync can cause perfectly smooth animation if an application’s framerate is higher then the monitor’s refresh rate. Disabled VSync can boost performance but it can cause visual tearing effect. When synchronization with vertical retrace is disabled monitor can start to draw a frame even when the previous frame is not finished. In this case you can see a piece of the previous frame in the top of the screen.

You can choose one of the following VSync modes:

- **On by default** – VSync is enabled by default and can be controlled by OpenGL application.
- **Off by default** – VSync is disabled by default and can be controlled by OpenGL application.
- **Always off** – VSync is disabled in all OpenGL applications.

Tip: Quake III cannot enable VSync via the game options even if **On by default** mode is selected. You can use its’ **\r_swapinerval 1** console command to enable VSync.

Buffer flipping mode

This allows you to control buffer flipping mode in full-screen OpenGL applications. You can select the **block transfer** mode, the **page flipping** mode or allow the driver to select the proper mode. The **page flipping** mode gives better performance, but the **block transfer** mode can solve some compatibility problems.

Tip: This setting also allows to control buffer flipping mode in windowed OpenGL applications on Quadro boards when the **Use page flipping in windowed mode when possible** is enabled.

Force triple buffering

This allows you to force OpenGL applications to use triple buffering. Triple buffering forcing will increase videomemory usage, but may help to improve framerate when **VSync** is enabled and framerate is below the monitor's refresh rate.

Tip: You may use RivaTuner's hardware monitoring module and **Local videomemory usage** graph to estimate impact on videomemory usage caused by triple buffering.

Force 16-bit Z-buffer

This allows you to save memory bandwidth by forcing 16-buffer in all OpenGL applications. It will improve performance a bit at the expense of image quality.

Tips:

• This setting is only active on GeForce2 MX and GeForce3 based boards on pre-4x.xx drivers.
• It looks like the driver performs additional CPU calculations when using this option. This results in boosting performance on fillrate dependent systems but dropping performance on CPU dependent systems.

Default bit depth for textures

This setting determines default bit depth for textures in OpenGL. An OpenGL application can either explicitly specify a bit depth for textures or use default bit depth. Usually applications allow you to select 16 bits per pixel, 32 bits per pixel or default bit depth. However, certain games (e.g. Quake II) don't allow user to select it and always use default bit depth. This option allows you to select a bit depth for such applications. You can select one of the following modes:

- **As desktop** – use textures of the bit depth at which your Windows desktop is currently running.
- **16 bits per pixel** - use 16 bits per pixel regardless of the desktop settings.
- **32 bits per pixel** - use 32 bits per pixel regardless of the desktop settings.

Tips:

- You can select **16 bits per pixel** in order to reduce texture thrashing in MDK 2.
- **32 bits per pixel** mode is not available on the Detonator 2.xx.

S3TC quality settings

This list of options allow you to customize texture compression quality. You can tune the following texture compression quality related options:

• **Compress to DXT3 instead of DXT1** option forces the driver to create DXT3 textures instead of DTX1. Unlike DXT1, DXT3 uses 32-bit text color interpolation and gives slightly better image quality in the cost of performance. You can enable this option in order to improve image quality in some games, which use runtime DXT1 texture compression (e.g. Quake III, Soldier of fortune, Serious Sam, DroneZ and others). This option will not improve image quality in the games, which use precompressed DXT1 textures (e.g. Unreal Tournament).
• **Disable dithering when decompressing DXT1 textures** option allows to disable color space dithering in NV17 and NV25 hardware DXT1 decompressors. This boosts performance in the cost of image quality.

Tips:

• DXT1 quality is a weak spot of pre-GeForce4 graphics processors. Hardware DXT1 decompressor uses 16-bit texel color interpolation. This often cause banding effects on the textures, which contain smooth color gradients (e.g. sky textures in Quake III). The rest compression formats use 32-bit texel interpolation and don't exhibit the banding effect. This issue is partially addressed in GeForce4 GPUs with color space dithering, implemented in hardware DXT1 decompressor.
• You can also fix DXT1 quality problem with RivaTuner's built-in **S3TCFixPack** patch scripts.
• These options are only active on GeForce 256 and higher graphics processors if the Detonator 6.47 or higher is installed.

Force fast trilinear filtering

TNT/TNT2 graphics processors cannot perform strict trilinear filtering in multitexturing mode. In this case the driver performs mipmap dithering instead of strict color blending. This setting allows you to force the driver to use mipmap dithering always, even in singletexturing mode.
It will improve performance at the expense of image quality.

Tip: This setting is only active on TNT/TNT2 boards. It doesn't have an effect on GeForce boards.

Intellisample settings

This slider allows you to control Intellisample texture filtering optimization algorithms, aimed to save fillrate due to texture filtering simplifications. There are three modes available:

• **High quality** mode forces the driver to follow the application's filtering requests with no filtering optimizations at all.
• **Quality** mode forces the driver to follow the application's filtering requests, but still allows you to enable optimizations separately.
However, this mode is handled differently in case of forcing anisotropic filtering on different driver versions. Pre-43.45 Detonators use strict anisotropic filtering for all rendered polygons, whilst new drivers reduce the amount of filtered pixels with sophisticated polygon rejection algorithm, allowing ignoring anisotropic filtering for certain polygons, filtered with non-primary texturing units.
This slightly improves performance at cost of image quality, which is not noticeable in the most of cases. If the application specifies degree of anisotropy itself, rejection algorithm is not used.
• **Performance** mode enables polygon rejection algorithm on pre-43.45 drivers, and additionally activates balanced texture contents based balanced anisotopic filtering simplification algorithm (on all Detonators for NV3x boards and on 43.51 and higher drivers for the rest boards).
On GeForceFX boards the driver also boosts performance via lowering trilinear filtering quality.
• **High performance** mode also enables polygon rejection algorithm, activates aggressive texture contents based anisotopic filtering simplification algorithm (on all Detonators for NV3x boards and on 43.51 and higher drivers for the rest graphics processors) and forces texture compression (on all Detonator versions for NV3x boards and on 43.45 or higher drivers on the rest boards).
On GeForceFX boards the driver also boosts performance via setting minimum trilinear filtering quality.

Tips:

• This setting is only active on GeForce 256 and higher boards if the Detonator 28.90 and higher is installed.
• This setting is locked on professional Quadro boards, however you may unlock it with RivaTuner's **AnisoBoosterOGL** patch script with pre-4x.xx Detonator drivers.
• Polygon rejection algorithm is disabled on pre-43.45 drivers when the application overrides default degree of anisotropy, but you may use RivaTuner's **AnisoBoosterOGL** patch script to remove this limitation.
• You may also use RivaTuner's **AnisoBoosterOGL** patch script in order to boost polygon rejection algorithm by allowing the driver to optimize filtering for the primary texturing unit too.

Override Intellisample trilinear filtering quality

This option allows you to override trilinear filtering quality on GeForceFX graphics processors family. By default trilinear filtering quality is determined by selected Intellisample mode, but you may select different trilinear filtering quality mode and use it regardless of selected Intellisample mode using this option and the **Trilinear filtering quality slider**.

Tip: This setting is only active on GeForce FX boards if the Detonator 41.80 or higher is installed.

Trilinear filtering quality slider

This slider allows you to override trilinear filtering quality on GeForceFX graphics processors family. By default trilinear filtering quality is determined by selected Intellisample mode, but you may select different trilinear filtering quality mode and use it regardless of selected Intellisample mode using this slider. You must check the **Override Inteliisample trilinear filtering quality** option to get access to this slider.

Tip: This slider is only active on GeForce FX boards if the Detonator 41.80 or higher is installed and.

Override Intellisample texture compression control

This option allows you to override Intellisample texture compression control. By default the OpenGL driver forces texture compression when you select the **High performance** Intellisample mode, but you override this behavior using this option and the T**exture compression mode** dropdown list.

Tip: This setting is only active on GeForce FX boards if the Detonator 41.80 or higher is installed and on the rest boards when the Detonator 43.51 or higher is installed.

Texture compression mode

This dropdown list allows you to override Intellisample texture compression control. By default the OpenGL driver forces texture compression when you select the **High performance** Intellisample mode, but you can override this behavior using this option and either disable texture compression forcing at all or force it regardless of used Intellisample mode.

Tip: This setting is only active on GeForce FX boards if the Detonator 41.80 or higher is installed and on the rest boards when the Detonator 43.51 or higher is installed.

Default degree of anisotropy

This setting determines default degree of anisotropy for OpenGL textures. In other word, this option enables anisotropic filtering by default but applications are still able to override it.
Anisotropy is a distortion visible in the texels of any 3D object whose surface is oriented at an angle with respect to the plane of the screen. If a pixel from such surface is mapped to texels, its shape is distorted. Anisotropic filtering is the most advanced texture filtering technique, that compensates for anisotropic pixel distortion. This technique gives the best image quality, especially when it's combined with trilinear filtering.
The OpenGL applications specify the degree of anisotropy to enable anisotropic filtering. Anisotropic texture filtering can be disabled by setting the degree of anisotropy to one, any value larger than one enables it. The greater degree of anisotropy means higher image quality but the lower performance.

Tips:

• The maximum degree of anisotropy depends on your hardware installed and the driver's version. It is limited to **level 2** on all pre-GeForce3 boards and on GeForce3 boards if the Detonator 10.xx or higher is not installed.
• This setting is only active on GeForce 256 and higher GPUs.

OpenGL hardware acceleration mode

This setting allows you to select the OpenGL hardware acceleration mode. You can select one of the following modes:

• **Max acceleration mode** allows the driver to enable all the features of your graphics hardware.
• **NVx compatibility mode** prevents the OpenGL driver from using NV1x+ specific features (e.g. hardware transformation and lighting, texture compression etc.) and forces it to detect your board as Riva TNT. You can use this mode for testing purposes.
• **NV1x compatibility mode** prevents the OpenGL driver from using NV2x+ specific features (e.g. pixel shaders, depth buffer compression etc.) and forces it to detect your board as GeForce2 MX. You can use this mode for testing purposes.
• **NV20 emulation mode** allows the OpenGL driver to emulate NV20 specific extensions/features. This mode is only available on GeForce256 and higher when the Detonator 7.xx - 7.49 or 10.xx and higher installed.
• **NV25 emulation mode** allows the OpenGL driver to emulate NV25 specific extensions/features. This mode is only available on GeForce256 and higher when the Detonator 2x.xx - 21.87 or 27.xx and higher is installed.
• **NV30 emulation mode** allows the OpenGL driver to emulate NV30 specific extensions/features. This mode is only available on GeForce256 and higher when the Detonator 40.41 and higher is installed.
• **NV40 emulation mode** allows the OpenGL driver to emulate NV40 specific extensions/features. This mode is only available on GeForce256 and higher when the Detonator 51.75 and higher is installed.

Tips:

• It’s not recommended to use the **NVx compatibility mode** on GeForce3 and higher boards.
• You can use the **NV1x compatibility mode** to force the driver to use SSAA instead of MSAA on GeForce3 and higher.

OpenGL version string override

This setting allows you to override version string, reported by the OpenGL ICD. It can be used for troubleshooting and working around the problems with certain applications, expecting hardcoded OpenGL version.

Tips:

• You can select **force 1.5** to run **The Chronicles of the Riddick – Escape from Butcher Bay** on OpenGL 2.0 capable 75.xx drivers.

Enable GL_SGIS_MULTITEXTURE in 16-bit modes

This setting allows TNT/TNT2 boards to use the GL_SGIS_MULTITEXTURE extension in 16-bit display modes. It was disabled in the Detonator 2.08 in order to drop down performance in 16-bit modes and promote 32-bit rendering. This option boosts some games (e.g. Quake II) which perform multitexturing via the GL_SGIS_MULTITEXTURE extension.

Tip: This setting is inactive on GeForce boards unless you use the **TNT compatibility mode**. This setting is also inactive on the Detonator 5x.xx and higher drivers.

Disable support for enhanced CPU instruction sets

You can use this option to prevent the OpenGL driver from using enhanced CPU instruction sets (e.g. 3DNow!, SSE).

Tip: Always keep enhanced CPU instructions support enabled. Disable it for testing purposes and troubleshooting only.

Disable Quadro specific features and driver optimization

This option forces the OpenGL driver to detect a Quadro boards as a GeForce. It prevents it from using CAD optimized code as well as disables all Quadro specific features.
It can improve performance in some games at the expense of performance in CAD applications.

Tips:

• This setting is only active on Quadro boards if the Detonator 14.10 or older is installed. NVIDIA have blocked this option in the Detonator 14.20 and higher.

Hardware accelerated lines support

This option allows to enable hardware accelerated lines on NV15 based boards in order to boost performance in wireframe OpenGL applications. Hardware accelerated lines have been disabled on NV15 boards revision A2 and A3 in order to promote Quadro2 Pro and increase performance gap between game oriented and CAD oriented products.
Almost all CAD oriented applications intensively use wireframe rendering so limitation of it causes a noticeable performance drop on NV15 based boards.

Tips:

• Always keep this option set to **force on**.
• This option is only active on NV11/NV15 boards when the Detonator 3x.xx or older is installed.

Enable unified back / depth buffer

This option allows the OpenGL driver to use unified back / depth buffer on Quadro boards. When this option is disabled the driver allocates a back buffer and depth buffer for each window. When this option is enabled the OpenGL driver allocates one back buffer and depth buffer at the same resolution of the display. It helps the OpenGL driver to use video memory more efficient and often may improve performance in the applications which create a lot of OpenGL windows.

Tips:

- This option is only active on Quadro boards.
- This option is only active on Windows 2000/XP if the Detonator 5.xx or higher is installed.
- FSAA and the use of unified back / depth buffer are mutually exclusive.

Enable page flipping in windowed modes when possible

This option allows the OpenGL driver to use page flipping in windowed mode when it is possible. It may improve the performance of some windowed OpenGL applications.

Tips:

• This option is only active on Quadro boards.
• This option is only active on Windows 2000/XP if the Detonator 6.xx or higher is installed.

Enable overlay

This option allows the OpenGL driver to export overlay pixel format. It will boost some applications (e.g. Maya) which can use overlays.

Tips:

• This option is only active on Quadro boards.
• This option is only active on Windows 2000/XP if the Detonator 6.47 or higher is installed.
• FSAA and the use of overlays are mutually exclusive.

Enable overlay

This option allows the OpenGL driver to export stereo pixel format. It will allow some applications (e.g. StereoVue-3ds plug-in) to use professional quadbuffered stereo.

Tips:

• This option is only active on Quadro boards.
• This option is only active on Windows 2000/XP if the Detonator 11.xx or higher is installed.
• FSAA and the use of stereo pixel format are mutually exclusive.

Quadbuffered stereo method

This option determines quadbuffered stereo method, which will be used by the driver when you enable stereo pixel format support.

Tips:

- This option is only active on Quadro boards.
- This option is only active on Windows 2000/XP if the Detonator 12.xx or higher is installed.
- FSAA and the use of stereo pixel format are mutually exclusive.

Swap left and rigth buffers (R becomes L; L becomes R)

This option allows to swap left and right framebuffers for quadbuffered stereo.

Tips:

• This option is only active on Quadro boards when the **Enable stereo** option is checked.
• This option is only active on Windows 2000/XP if the Detonator 21.81 or higher is installed.
• FSAA and the use of stereo pixel format are mutually exclusive.

Enable antialiasing

This option allows you to enable FSAA in all OpenGL applications.

Tips:

• The term FSAA is shorthand for “full-scene antialiasing” , which is a technique that can be used to reduce the appearance of jaggies (stair-step pixels, aliasing artifacts) when drawing any line that is not exactly horizontal or vertical. In 3D scenes , these artifact are most noticeable on the boundaries between polygons.
Antialiasing improves image quality by reducing or eliminating the jaggies but it may cause noticeable performance drop.
• FSAA can often make in-game head-up-displays (HUDs) and fonts appear blurry. Try to disable it if you see abnormally blurred HUD or font.
• FSAA and the use of unified back / depth buffer or stereo / overlay pixel format on Quadro boards are mutually exclusive.

Antialiasing method

This allows to select the antialiasing method. **1.5 x 1.5** and **2x** give the best performance while the rest give better image quality at the expense of some performance drop.

Tip: The list of available antialiasing methods depends on the hardware installed in your system and the version of the Detonator drivers.

Enable texture sharpening

This option allows the OpenGL driver to double the current degree of anisotropy when FSAA is enabled. It may help to increase image sharpness and reduce blurring side effect, especially in conjunction with SSAA and **Quincunx** / **4x 9-tap** antialiasing modes.

Tip: This setting is only active on the Detonator 27.xx and higher.

Apply this color scheme at Windows startup

Enabling this option will automatically apply the current color scheme at Windows startup.

Tip: In order to avoid color correction interference don't try to use this option simultaneously with low-level color correction.

Color adjustment sliders

These sliders allow you to adjust the brightness, contrast and gamma values for each or for all of the RGB channels. These adjustments apply to Direct3D/OpenGL games as well as to the desktop.

Tips:

• The adjustments don't have an effect on the applications which perform own gamma correction (e.g. Quake III).
• Try to change the **ExtEscapeColorControl** registry entry if color correction doesn't work on your system (Windows 2000/XP only).

Minimum brightness

This is the minimum brightness value you can set.

Maximum brightness

This is the maximum brightness value you can set.

Current brightness

Displays the current brightness value. You can adjust it with the **Brightness** slider.

Minimum contrast

This is the minimum contrast value you can set

Maximum contrast

This is the maximum contrast value you can set.

Current contrast

Displays current contrast value. You can adjust it with the **Contrast** slider.

Minimum gamma

This is the minimum gamma value you can set.

Maximum gamma

This is the maximum gamma value you can set.

Current gamma

Displays current gamma value. You can adjust it with the **Gamma** slider.

Digital vibrance control

This allows to change DVC mode. The term DVC is shorthand for “Digital Vibrance Control”, which is a hardware technique that boosts color saturation without changing the gamma ramp.

Tips:

• This option is only available on GeForce2 MX and GeForce3 based boards.
• Try to change the **ExtEscapeColorControl** registry entry if color correction doesn’t work on your system (Windows 2000/XP only).

Current digital vibrance control

Displays current DVC mode. You can change it with the **Digital vibrance control** dropdown list.

Gamma ramp window

Displays current gamma ramp. It shows how each (R, G and B) or all channels are distributed when you move the **Brightness**, **Contrast** or **Gamma** slider to make your adjustments.

Color channel

This allows to select the color channel, which will be controlled by the **Brightness**, **Contrast** or **Gamma** sliders. You can adjust the R, G or B channels individually or all three channels at once.

Save color scheme

This button allows you to save the current color adjustments as a color scheme, which will then be Added to the adjacent list. You can activate this color scheme later by selecting it in this list.

Tips:

• You can associate a color scheme you have created with a launcher item via the **Launcher** tab.
• RivaTuner's color scheme manager is fully compatible with the Detonator's one. You can use the schemes you have created in RivaTuner via the Detonator's **Color correction** tab and vice versa.

Delete color scheme

This button allows you to remove a custom color scheme. It is only active when you select a color scheme in the adjacent list.

Color schemes

This is a list of the color schemes you have created. Selecting an item from this list will activate the scheme.

Overlay color adjustment sliders

These sliders allow you to adjust the brightness, contrast, hue and saturation for overlay surfaces.

Tips:

• Overlay surfaces are the areas in video memory with the special hardware-supported capabilities, which can be displayed over the area in memory containing the image being displayed on the monitor without blitting to it or changing the its' contents in any way. Overlay surfaces are frequently used to display video.
• Use these controls if the movies appear too dark on your PC.

Minimum saturation

This is the minimum saturation value you can set.

Maximum saturation

This is the maximum saturation value you can set.

Current saturation

Displays the current saturation value. You can adjust it with the **Saturation** slider.

Minimum hue

This is the minimum hue value you can set.

Maximum hue

This is the maximum hue value you can set.

Current hue

Displays current hue value. You can adjust it with the **Hue** slider.

Launcher item name

This is the name of launcher item you are editing. This is displayed in the list of launcher items and in RivaTuner's tray menu and it just helps you to identify the settings associated with this item. You can leave this field blank to create separator items. Such items cannot be launched but can be used to separate different types of launcher items, for example applications, color schemes, display modes and so on. It makes the menu more readable.

Associated hotkey

This allows you to associate a hotkey with this launcher item. You can select this field then press the combination of keys you want to associate with it.

Tips:

• RivaTuner will not allow you to duplicate the hotkeys. You will not be able to use the combination of keys, which is already registered in your system.
• RivaTuner uses native Windows hotkey handler by default. It works flawlessly under Windows 2000 / XP, but under Windows 9x almost all standard input messages are suppressed when any DirectInput application use the keyboard in exclusive mode. This results in ignoring hotkeys when the most of games are running.
You may use the **Emulate hotkeys handler** option to force RivaTuner to emulate hotkeys when running such applications. It allows RivaTuner to scan the keyboard within specified time intervals and generate required hotkey related messages when it is necessary.

Associated preset

This option allows to associate a preset with the current launcher item. It will be automatically Applied when you select this item.

Tips:

• Preset is a script, which can add or remove the registry entries. RivaTuner uses presets to store the driver's settings.
• You can either use pre-created presets, or create your own preset via the **Power user** page or preset creation wizard. You can press the **Create** button in order to create a preset via the preset creation wizard.
• **Restore settings after terminating application** option allows to restore the registry entries changed by the preset when associated application is closed.

Preset path

This field allows you to type a path to the preset file associated with the current launcher item.

Tip: RivaTuner's presets are located in the **.\Presets** subfolder.

Browse

This button allows you to browse for the preset file associated with current launcher item.

Tip: RivaTuner's presets are located in the **.\Presets** subfolder.

Create

This button allows you to create a preset via the preset creation wizard. It will automatically find the best matched database for the currently installed Detonator drivers and create a preset from the driver's settings using this database.
You will be prompted to enter the path to the file containing new preset.

Tips:

• This button is inactive if an NVIDIA based display adapter is not selected in the **Target adapter** dropdown list or if a supported Detonator driver is not detected.
• This button is inactive if RivaTuner cannot find the database for currently installed drivers.

Associated color scheme

This option allows you to associate a **low-level** or **driver-level color scheme** with the current launcher item. It will be automatically applied when you select this item.

Tips:

• This option is inactive if neither **low-level** nor **drivel-level color schemes** are available on your system.
• **Low-level color schemes** are available on any **Get/SetDeviceGammaRamp** capable display adapters.
• **Driver-level color schemes** are available on NVIDIA display adapters only.
• **Restore settings after terminating application** setting allows restoring the previous color scheme when associated application is closed.

Associated color scheme type

This dropdown list allows selecting type of associated **color scheme**. Depending on your system configuration, it may support **low-level** and (or) **driver-level color schemes**.

Tips:

• This option is inactive if neither **low-level** nor **drivel-level color schemes** are available on your system.
• **Low-level color schemes** are available on any **Get/SetDeviceGammaRamp** capable display adapters.
• **Driver-level color schemes** are available on NVIDIA display adapters only.

Associated color scheme name

This is the list of **color schemes** you can associate with the current launcher item.

Tips:

• The list is inactive if you have not created any **color schemes** of type you have selected.
• You can create a **driver-level color scheme** via RivaTuner's d**river-level color adjustment** dialog or directly via NVIDIA driver's control panel interface.
• You can create a **low-level color scheme** via RivaTuner's **low-level color adjustment** dialog.

Lock color scheme

This is option allows RivaTuner to check gamma ramp during application runtime and forcibly set the associated color scheme when application attempts to make own gamma ramp adjustments.
This ensures that the associated color scheme will affect even the applications performing internal gamma correction via GDI or DirectX functions.

Tips:

• This option is active only when you associate low-level color scheme with an application running in RAT mode.
• By default RivaTuner scans gamma ramp for changes once per second during application runtime. However, you may override checking period with **ColorSchemeOverrideInterval** registry entry.

Associated overclocking profile

This option allows you to associate a **low-level** or **driver-level overclocking profile** with the current launcher item. It will be automatically applied when you select this item.

Tips:

• This option is inactive if neither **low-level** nor **drivel-level overclocking profiles** are available on your system.
• **Low-level overclocking profiles** are available on all supported ATI display adapters and on all pre-NV30 NVIDIA display adapters.
• **Driver-level overclocking profiles** are available on NVIDIA display adapters only.
• **Restore settings after terminating application** setting allows to restore the previous clocks when associated application is closed.

Associated overclocking profile type

This dropdown list allows selecting type of associated **overclocking profile**. Depending on your system configuration, it may support **low-level** and (or) **driver-level overclocking profiles**.

Tips:

• This option is inactive if neither **low-level** nor **drivel-level overclocking profiles** are available on your system.
• **Low-level overclocking profiles** are available on all supported ATI display adapters and on all pre-NV30 NVIDIA display adapters.
• **Driver-level overclocking profiles** are available on NVIDIA display adapters only.

Associated overclocking profile name

This is the list of **overclocking profiles** you can associate with the current launcher item.

Tips:

• The list is inactive if you have not created any **overclocking profiles** of type you have selected.
• You can create **low-level overclocking profiles** via RivaTuner's **low-level system settings** dialog.
• You can create a **driver-level overclocking profiles** via RivaTuner's **driver-level system settings** dialog.

Associated fan profile

This option allows you to associate a **low-level** or **driver-level fan profile** with the current launcher item. It will be automatically applied when you select this item.

Tips:

• This option is inactive if neither **low-level** nor **drivel-level fan profiles** are available on your system.
• **Low-level fan profiles** are available on display adapters equipped with **LM63**, **LM64**, **F75363S** or **ADT7473** I2C PWM controllers.
• **Driver-level fan profiles** are available on NV30 and newer display adapters with fan connected to reference PCB's integrated PWM controller.
• **Restore settings after terminating application** setting allows to restore the previous fan settings when associated application is closed.

Associated fan profile type

This dropdown list allows selecting type of associated **fan profile**. Depending on your system configuration, it may support **low-level** and (or) **driver-level fan profiles**.

Tips:

• This option is inactive if neither **low-level** nor **drivel-level fan profiles** are available on your system.
• **Low-level fan profiles** are available on display adapters equipped with **LM63**, **LM64**, **F75363S** or **ADT7473** I2C PWM controllers.
• **Driver-level fan profiles** are available on NV30 and newer display adapters with fan connected to reference PCB's integrated PWM controller.

Associated fan profile name

This is the list of **fan profiles** you can associate with the current launcher item.

Tips:

• The list is inactive if you have not created any **fan profiles** of type you have selected.
• You can create **low-level fan profiles** via RivaTuner's **low-level system settings** dialog.
• You can create a **driver-level fan profiles** via RivaTuner's **driver-level system settings** dialog.

Associated display mode

This allows you to associate a display mode with the current launcher item. It will be automatically set when you select this item.

Tips:

• This option can virtually fix the mouse cursor bug, which has been introduced in the Detonator 6.47. Due to unknown reason mouse cursor appears on the screen after display mode switch in fullscreen Direct3D applications on Windows 9x.
You can associate a display mode with any application, which exhibits such problem and so prevent It from switching display mode.
• **Restore settings after terminating application** setting allows to set the previous display mode when associated application is closed.

Display modes list

This is the list of display modes you can associate with the current launcher item.

Tip: Refresh rate control is only available under Windows 2000/XP.

Associated application

This option allows you to associate an application with the current launcher item. This application will be automatically executed when you select this item.

Restore settings after terminating application

This option allows you to restore the previous display mode, color scheme and the registry entries changed by the preset after terminating associated application.

Tip: You must associate an application and a preset, display mode or color scheme with the current launcher item in order to activate this option.

Associated application path

This field allows you to type a path to the application associated with current launcher item.

Browse

This button allows you to browse for the application associated with current launcher item.

Associated application command line

This field allows you to type the command line parameters, which will be passed to the associated application.

Associated application priority class

This allows you to select a priority class for the associated application. Higher priority classes can boost performance.

Tip: It is not recommended to use the **Realtime priority class** under Windows 2000 / XP.

Launcher item name

This is the name of launcher item you are editing. This is displayed in the list of launcher items and in RivaTuner's tray menu and it just helps you to identify the module associated with this item.
You can leave this field blank to force RivaTuner to use the name of the associated module as item name.

Module type

This option allows you to select type of module you are about to activate with this item. There are three module types available:

• **Main property sheet** can be used for quick access to any tab available in RivaTuner's main property sheet. Once selected this module type, you may select required tab in the list below.
• **Driver-level module** can be used for quick access to any driver-level tweaking module available for currently selected display adapter and display driver. Once selected this module type, you may select required module name and tab name if available.
• **Low-level module** can be used for quick access to any low-level tweaking module available for currently selected display adapter. Once selected this module type, you may select required module name and tab name if available.

Module name

This option allows you to select name of module you are about to activate. The list of module names entirely depends on your system configuration and selected module type.

Tab name

This option allows you to select name of tab you are about to activate.

Enable driver-level hardware overclocking

This option allows to enable driver-level hardware overclocking. It can adjust the clock frequencies via the driver's overclocking interface.
To ensure the proper operation of the clock frequency controls RivaTuner must detect default core and memory clock frequencies. Reboot is strongly recommended in order to detect them correctly. You will be prompted to reboot your PC after enabling this option. However, the power users may ignore this recommendation and detect defaults without reboot. Don't use this mode if you are not sure that RivaTuner shows the real defaults in the **Current core / memory clock frequency** fields.

Tips:

• RivaTuner automatically reverts the clock frequencies to their default values after disabling driver level hardware overclocking.
• Due to the bug in certain versions of the Detonators they can report wrong clock frequencies (e.g. 100 MHz core and 83 MHz memory) after resuming from standby mode. In this case, enabling driver level hardware overclocking without reboot will cause the driver to detect invalid defaults.
Any attempt to disable hardware overclocking will cause wrong default frequencies to be restored and the system may hang. In this case, you may use the **Restore the driver's settings to their default values** button on the **driver customization toolbar**.

Current core clock frequency

Displays the current core clock frequency reported by the driver.

Tips:

• Due to the bug in certain versions of the Detonators they can report wrong clock frequencies (e.g. 100 MHz core and 83 MHz memory) after resuming from standby mode. You can use the **Restore clock frequencies after suspended mode** option in order to address this problem.

Current memory clock frequency

Displays the current memory clock frequency reported by the driver.

Tips:

• Due to the bug in certain versions of the Detonators they can report wrong clock frequencies (e.g. 100 MHz core and 83 MHz memory) after resuming from standby mode. You can use the **Restore clock frequencies after suspended mode** option in order to address this problem.
• The drivers may report double memory clock frequency on certain GeForce2 MX200/MX400 and GeForce4 MX420 boards. This problem is caused by a hardware feature of the clock frequency generator, which can be hardwired to halve the MPLL clock frequency. MPLL doubling is used on the boards equipped with DDR memory, but some hardware vendors also use it on GeForce2 MX 200 / MX400 and GeForce4 MX420 with 4Mx16 SDR memory modules.

Overclocking mode

This option allows setting independent clock frequencies for **standard 2D** mode, **low power 3D** and **performance 3D** mode.

Tip: By default **low power 3D** mode controls are hidden to prevent beginners from making wrong settings. Power users may enable clock frequency controls for low power 3D mode by setting **EnableLowPower3DControl** registry entry to 1.

Clock frequency adjustment sliders

These sliders allow you to adjust the core and memory clock frequencies.

Tips:

• The yellow exclamation signs on the core/memory icons indicate that the clock frequency is beyond the driver's overclocking range. It is highlighted on both sliders (from 90% till 125% in reference to the default clock frequency).
• The range of the clock frequencies you can set is calculated using the default clock frequency. By default, it is allowed to adjust it from 50% till 150% in reference to the default clock frequency. However, you can override both limits with **MinClockLimit** / **MaxClockLimit** registry entries.

Minimum core clock

This is the minimum core clock frequency you can set.

Tips:

• The range of the clock frequencies you can set is calculated using the default clock frequency. By default, it is allowed to adjust it from 50% till 150% in reference to the default clock frequency. However, you can override the bottom limit with the **MinClockLimit** registry entry.

Maximum core clock

This is the maximum core clock frequency you can set.

Tips:

• The range of the clock frequencies you can set is calculated using the default clock frequency. By default, it is allowed to adjust it from 50% till 150% in reference to the default clock frequency. However, you can override the top limit with the **MaxClockLimit** registry entry.

Minimum memory clock

This is the minimum memory clock frequency you can set.

Tips:

• The range of the clock frequencies you can set is calculated using the default clock frequency.

Maximum memory clock

This is the maximum memory clock frequency you can set.

Tips:

• The range of the clock frequencies you can set is calculated using the default clock frequency. By default, it is allowed to adjust it from 50% till 150% in reference to the default clock frequency. However, you can override the top limit with the **MaxClockLimit** registry entry.
• The maximum memory clock frequency is limited to 400MHz on SDR boards under Windows 9x.
• Due to the bug in the drivers (wrong SDR / DDR detection) the maximum memory clock is limited to 400MHz on all boards (except for GeForce DDR) under Windows 2000 / XP on the Detonators older than 12.90.
You may use RivaTuner's built-in **NvXTInitFix** patch script in order to fix this bug.

Force constant performance level

This option allows you to disable performance levels switching on the boards with separate 2D/3D clock frequency adjustment and force display adapter to work constantly at desired performance level.

Tip: The driver's internal throttling mechanisms are disabled by forcing constant performance level. This greatly increases the risk of hardware failures caused by too aggressive overclocking, so this option is disabled by default to prevent beginners from potential problems caused by blind tweaking. However, power users may unlock this option with **EnablePerfLevelForcing** registry entry.

Apply overclocking at Windows startup

This option allows you to apply overclocking settings at Windows startup.

Tips:

• Overclocking settings may be ignored at Windows 2000 / XP startup if memory clock frequency is above 400MHz. It is a know bug of the drivers and it was fixed in the Detonator 12.90. You can also fix this bug on any older drivers with RivaTuner's built-in **NvXTInitFix** patch script.
• In order to avoid clock settings interference don't try to use this option simultaneously with low-level overclocking.
• You can press and hold **Ctrl** during Windows startup to reset startup overclocking settings.

Use alternate startup daemon

This option forces RivaTuner to use its' own startup daemon to apply driver-level clock frequencies at Windows startup. It must be checked when you use the features like custom startup clocks for **low power 3D** mode, which are not supported by standard control panel's startup daemon.

Save

This button allows to save startup overclocking settings. This button is only active when the **Apply overclocking at Windows startup** setting is enabled and current clock frequencies differ from current startup settings.

Tips:

• Overclocking settings may be ignored at Windows 2000 / XP startup if memory clock frequency is above 400MHz. It is a know bug of the drivers and it was fixed in the Detonator 12.90. You can also fix this bug on any older drivers with RivaTuner's built-in **NvXTInitFix** patch script.

Test

This button allows you to test the current lock frequencies. You will not be able to press the **Apply** and **Save** buttons until you test new clock frequencies.

Tip: Power users may disable testing via the **DisableClockTest** registry entry.

Overclocking profiles list

This list allows you to select **overclocking profile** you have created. Once selected, you may apply the clock frequencies stored in the profile without testing them.

Tips:

• When you use this list to load **overclocking profile** on the system with separate 2D/3D clock frequency adjustment, RivaTuner applies changes to the currently selected **performance level** only to prevent you from applying the clocks you don't see. However, the changes are applied to all **performance levels** when the **overclocking profile** is loaded via **Launcher** tab.

Restore clock frequencies after suspended mode

This option allows you to fix post-suspended mode underclocking problem. Due to the bug in certain versions of the Detonator drivers they can report wrong clock frequencies (e.g. 100 MHz core and 83 MHz memory) after resuming from standby mode.
Enabling this option allows RivaTuner to save the clock frequencies when the system is entering in standby mode and restore it after resuming.

Tip: RivaTuner must reside in memory when using this option.

Current fan speed

Displays the current fan speed for the corresponding **performance level**.

Tips:

• This value is trustworthy only when the fan is physically connected to the PCB's integrated PWM (**Pulse Width Modulation**) controller. If external PWM controller is used (e.g. **F75373S/F75375S** on **ASUS V9999** series, **ADT7473** on **GeForce 7800GTX** series) or the fan is not connected to any PWM controller at all (e.g. **MSI NX6800** series), then the driver will not be able to control fan speed, as well as will not be able to report trustworthy values.
• **NV30/NV35/NV40** owners may also directly retrieve programmed fan speed from the hardware and monitor it at **Reference fan duty cycle** graph via **hardware monitoring module**.

Fan speed adjustment sliders

These sliders allow you to adjust fan speed for the corresponding **performance level**.

Tips:

• This value is trustworthy only when the fan is physically connected to the PCB's integrated PWM (**Pulse Width Modulation**) controller. If external PWM controller is used (e.g. **F75373S/F75375S** on **ASUS V9999** series, **ADT7473** on **GeForce 7800GTX** series) or the fan is not connected to any PWM controller at all (e.g. **MSI NX6800** series), then the driver will not be able to control fan speed, as well as will not be able to report trustworthy values.
• **NV30/NV35/NV40** owners may also directly retrieve programmed fan speed from the hardware and monitor it at **Reference fan duty cycle** graph via **hardware monitoring module**.
• Take a note that fan speed is not specified directly, value you set is a **fan duty cycle** passed to PCB's integrated PWM controller. **Fan duty cycle** is defined with **Ton/T** formula, were **T** is pulse width modulation frequency period and **Ton** is a part of period **T**, during which controller supplies voltage to the fan's input. Depending on their constructive specifics, different fans have different minimum fan duty cycles, causing fan to rotate. So be careful and take a note that some fans can work at low duty cycles below **10%**, but some can stop even at **50%**.
Due to this specific, by default the minimum value you can set is limited with **25%**. However, you may override it with **MinFanSpeedLimit** registry entry.

Minimum fan speed

This is minimum fan speed you are allowed to set for the corresponding **performance level**.

Tips:

• Take a note that fan speed is not specified directly, value you set is a **fan duty cycle** passed to PCB's integrated PWM controller. **Fan duty cycle** is defined with **Ton/T** formula, were **T** is pulse width modulation frequency period and **Ton** is a part of period **T**, during which controller supplies voltage to the fan's input. Depending on their constructive specifics, different fans have different minimum fan duty cycles, causing fan to rotate. So be careful and take a note that some fans can work at low duty cycles below **10%**, but some can stop even at **50%**.
Due to this specific, by default the minimum value you can set is limited with **25%**. However, you may override it with **MinFanSpeedLimit** registry entry.

Apply fan settings at Windows startup

This option allows you to apply fan settings at Windows startup.

Tips:

• You can press and hold **Ctrl** during Windows startup to reset startup fan settings.

Save

This button allows you to save current fan speeds as startup fan speeds. This button is only active when the **Apply fan settings at Windows startup** is enabled and current fan speeds differ from current startup fan speeds.

Save fan profile

This button allows you to save the current fan settings as **fan profile**, which will then be added to the adjacent **fan profiles list**. You can use profiles list later for quick applying these settings.

Tips:

• You can associate a **fan profile** you have created with a launcher item via the **Launcher** tab.

Delete fan profile

This button allows you to remove a **fan profile**. It is only active when you select a profile in the adjacent list.

Fan profiles list

This list allows you to select a **fan profile** you have created. Once selected, you may apply the settings stored in the profile.

Command buffer location

This setting allows you to select the command buffer location in order to improve compatibility with certain Super Socket 7 boards. You can select one of the following modes:

• **Strict AGP** – allows the driver to store the command buffer in AGP memory. This mode is selected by default and it gives the best performance.
• **PCI** – the driver uses PCI memory for the command buffer. It improves compatibility at the expense of performance.
• **PCI with reduced heap** – the same as the previous mode but the size of AGP texture heap will be reduced on the size of this buffer. This mode is a bit slower than the **PCI**, but it gives maximum compatibility.

Tips:

• This option works with Windows 9x only.
• This option is only active on ALi chipset based motherboards when the Detonator 3.xx or higher is installed.
• This option is always active on the Detonator 2.xx regardless of the motherboard chipset.

Enable AGP 4x transfer rate

This setting allows you to enable AGP 4x transfer rate on VIA 694X chipset based motherboards. On such motherboards (e.g. ASUS P3V4X) NVIDIA drivers default to AGP 2X transfer rate to work around insufficient drive strength on one of the signals.
This should only be used if there is reason to believe that the motherboard has the appropriate drive strength. Enabling this setting can often cause 3D applications to hang, especially on GeForce2 GTS based systems.

Tips:

• This option is only active on VIA 694X based platforms if the Detonator 5.22 or higher is installed.
• Due to the bug in the drivers this option can be ignored on some systems after installing VIA 4in1 service pack.

Enable AGP sideband addressing

This option allows to enable sideband addressing on AMD Irongate based motherboards. On such motherboards NVIDIA drivers disable it by default to work around a problem with the signal integrity of the chipset.

Tip: This option is only active on AMD Irongate based motherboards if the Detonator 3.69 or higher is installed.

Enable AGP 2x transfer rate

This option allows to enable AGP 2x transfer rate on AMD Irongate based motherboards. On such motherboards NVIDIA drivers default to 1X transfer rate to work around a problem with the signal integrity of the chipset.

Tip: This option is only active on AMD Irongate based motherboards if the Detonator 3.68 or higher is installed.

Enable motherboard chipset compatibility mode

This option allows the driver to improve compatibility with certain motherboard chipsets. The driver can detect the most of motherboard chipsets and automatically self-configure in order to improve compatibility with detected hardware. This option can be only enabled when RivaTuner detects a motherboard, which cannot be automatically configured by the driver. Enabling this option forces the driver to use common vendor-specific internal compatibility configuration.

Tip: This option is only active if the Detonator 6.xx or higher is installed.

Disable accelerated graphics port

This setting allows you to disable AGP and force the graphics subsystem to work in PCI mode.

Tips:

• Disabling AGP causes noticeable performance drop. Use this option for troubleshooting only.
• This option is only active on Windows 2000 / XP.

AGP transfer rate

This setting allows you to set AGP transfer rate manually. The maximum AGP transfer rate you can set depends on your motherboard's and graphics adapter's capabilities. Usually the drivers analyze hardware capabilities and automatically set the maximum transfer rate supported by your graphics subsystem.
However, sometimes you may need to decrease transfer rate in order to improve stability on certain systems (e.g. Intel 440LX based platforms may work unstable in AGP 2x mode).

Tips:

• **WARNING!** This option may work incorrectly on some non-Intel based motherboards and may cause your system to hang during booting. Don't use this option unless you completely understand what are your doing. If your system hanged during loading, you will need to load your operating system in safe mode then search and delete the **ReqAGPRate** registry entry.
• This option is protected by the **LockDangerSettings** registry entry. You must set it to 0 in order to get access to this option. I did it to ensure that you have read this information before using this option.
• This option is only active on AGP display adapters if the Detonator 3.xx or higher is installed.

AGP sideband addressing

This setting allows you to control AGP sideband addressing. Usually the drivers analyze hardware capabilities and enable sideband addressing if it is supported by the motherboard and the graphics adapter. But in some cases you may need to control it manually.

Tips:

• **WARNING!** This option may work incorrectly on some non-Intel based motherboards and may cause your system to hang during booting. Don't use this option unless you completely understand what are your doing. If your system hanged during loading, you will need to load your operating system in safe mode then search and delete the **EnableAGPSBA** registry entry.
• This option is protected by the **LockDangerSettings** registry entry. You must set it to 0 in order to get access to this option. I did it to ensure that you have read this information before using this option.
• This option is only active on AGP display adapters if Detonator 6.47 or higher is installed.

AGP fastwrites

This setting allows you to control AGP fastwrites protocol. Usually the drivers analyze hardware and enable fastwrites protocol if it is supported by the motherboard and the graphics adapter. But in some cases you may need to control it manually.

Tips:

• **WARNING!** This option may work incorrectly on some non-Intel based motherboards and may cause your system to hang during booting. Don't use this option unless you completely understand what are your doing. If your system hanged during loading, you will need to load your operating system in safe mode then search and delete the **EnableAGPFW** and **EnableAGPFWStrap** registry entries.
• This option is protected by the **LockDangerSettings** registry entry. You must set it to 0 in order to get access to this option. I did it to ensure that you have read this information before using this option.
• This option is only active on AGP display adapters if Detonator 5.32 or higher is installed.
• This option is inactive if motherboard doesn't support AGP fastwrites protocol.

Low-level AGP transfer rate monitor

Displays the current AGP transfer rate. RivaTuner reads this value directly from PCI configuration space so you may use it to verify the driver's settings.

Low-level AGP sideband addressing monitor

Displays the current AGP sideband addressing state. RivaTuner reads this value directly from PCI configuration space so you may use it to verify the driver's settings.

Low-level AGP fastwrites monitor

Displays the current AGP fastwrites protocol state. RivaTuner reads this value directly from PCI configuration space so you may use it to verify the driver's settings.

Overlay zooming area

This allows you to select the area of overlay you would like to zoom. Once selected, you can zoom by moving the **Overlay zooming ratio** slider.

Tips:

• Overlay surfaces are the areas in video memory with the special hardware-supported capabilities, which can be displayed over the area in memory containing the image being displayed on the monitor without blitting to it or changing the its' contents in any way. Overlay surfaces are frequently used to display video.
• This setting is only active on the Detonator 6.xx and higher.

Overlay zooming ratio

This allows to zoom in or out on the selected area of overlay. You must select the area of overlay you would like to zoom before using this control.

Tips:

• Overlay surfaces are the areas in video memory with the special hardware-supported capabilities, which can be displayed over the area in memory containing the image being displayed on the monitor without blitting to it or changing the its' contents in any way. Overlay surfaces are frequently used to display video.
• This setting is only active on the Detonator 6.xx and higher.

Force hardware overlay support on spanning modes

This option enables hardware overlay to be forced on for the NV11 spanning modes.

Tips:

• Overlay surfaces are the areas in video memory with the special hardware-supported capabilities, which can be displayed over the area in memory containing the image being displayed on the monitor without blitting to it or changing the its' contents in any way. Overlay surfaces are frequently used to display video.
• This setting is only active on NV11 based boards (GeForce2 MX, Quadro2 EX, Quadro2 MXR) when the Detonator 7.52 - 8.xx or 10.50 and higher is installed.
• This setting works with Windows 2000 / XP only.

Use busmastering mode for video

Bus-mastering TV tuners that don’t flip can leave artifacts when they are scaled. This problem can be addressed by enabling this option.
It causes the DirectDraw driver to look for flips occurring within half-second intervals. If none are found, an overlay automatically starts flipping at 30 fps.

Tips:

• Overlay surfaces are the areas in video memory with the special hardware-supported capabilities, which can be displayed over the area in memory containing the image being displayed on the monitor without blitting to it or changing the its' contents in any way. Overlay surfaces are frequently used to display video.
• This setting works with Windows 9x only.

Refresh rates list

This is the list of display modes and the minimum refresh rate for each display mode, supported by your graphics subsystem. You can edit the list of refresh rates by specifying the minimum refresh rate for each display mode. After editing it you may press the **Install** button to save changes in the registry. Reboot will be required in order to apply changes.

Tips:

• Due to Windows 2000/XP kernel specific all fullscreen applications set the first refresh rate from the list of supported refresh rates when switching to fullscreen mode. This kernel specific causes well-known 60Hz refresh rate problem. Luckily, many display drivers for Windows 2000 / XP allow user to customize the list of supported display modes/refresh rates. This allows you to edit the list and change the first supported refresh rate for each display mode.
Unfortunately, it's impossible to change the order of supported refresh rates in this list, so the only way to work around 60Hz problem is to remove this refresh rate from the list at all. In this case the next supported refresh rate (usually 70Hz) will be used for all fullscreen applications. If you want to increase it further, you'll need to remove this refresh from the list too. This means that in order to set 100Hz refresh rate in all fullscreen applications you will need to remove all refresh rates below 100Hz from the list.
• RivaTuner uses Win32 API in order to fill the list of supported display modes and refresh rates. It means, that it may detect the maximum supported refresh rate incorrectly if you have not properly installed the drivers for your monitor. RivaTuner will display warning message if your monitor isn't configured properly (this applies to non-localized versions of Windows 2000 / XP only).
• Avoid removing 60Hz refresh rate for those display modes, which you are using for TV-out. This may block TV-out functionality.
• Fixing 60Hz problem may cause abnormally low scores in 3Dmark 2001 (SE)/Game 4/Nature test and some other applications. This issue is caused by the Detonator's display mode switch related bug, which was fixed in the Detonator 40.xx. This results in jerky framerate in certain applications if display mode hadn't been switched at least once after reboot. When 60Hz refresh rate fix is not installed any Direct3D application (e.g. 3DMark) switch display mode when switching to fullscreen mode and setting 60Hz refresh rate (except for those, who use 1024x768x32, 60Hz for desktop). After installing refresh rate fix 2D/3D display modes and refresh rates can match so the Direct3D can just skip display mode switching and result in this problem.
As a workaround for this problem you may either upgrade your drivers to version 40.xx or switch display mode manually before running 3DMark.
As an alternative, you may use RivaTuner's **launcher** and associate any 16-bit display mode with 3DMark.

Select all modes

This button allows you to select all the supported display modes. Once selected, you can press the following group buttons: **Set minimum refresh rates for all selected display modes**, **Enter minimum refresh rates for all selected display modes** and **Set maximum refresh rates for all selected display modes**.

Tip: You can toggle display mode selection by clicking to the left of its' name.

Unselect all modes

This button allows you to unselect all the supported display modes.

Tip: You can toggle display mode selection by clicking to the left of its' name.

Set minimum refresh rates for all selected display modes

This button allows you to set the minimum supported refresh rates for all selected display modes.

Tips:

• This button is active if you have selected at least one display mode. You can toggle display mode selection by clicking to the left of its' name or use the **Select all modes** and **Unselect all modes** buttons.
• RivaTuner uses Win32 API in order to fill the list of supported display modes and refresh rates. It means that you will not be able to restore the previously removed refresh rates and detect the real minimum supported refresh rate by pressing this button. You can use the **Uninstall** button instead.

Enter minimum refresh rates for all selected display modes

This button displays the dialog, which allows you to enter the minimum supported refresh rates for all selected display modes.

Tips:

• This button is active if you have selected at least one display mode. You can toggle display mode selection by clicking to the left of its' name or use the **Select all modes** and **Unselect all modes** buttons.
• RivaTuner automatically analyzes your graphics subsystem and selects the closest refresh rates for those display modes, which don't support requested refresh rate.

Set maximum refresh rates for all selected display modes

This button allows you to set the maximum supported refresh rates for all selected display modes.

Tips:

• This button is active if you have selected at least one display mode. You can toggle display mode selection by clicking to the left of its' name or use the **Select all modes** and **Unselect all modes** buttons.
• RivaTuner uses Win32 API in order to fill the list of supported display modes and refresh rates. It means, that it may detect the maximum supported refresh rate incorrectly if you have not properly installed the drivers for your monitor. RivaTuner will display warning message if your monitor isn't configured properly (this applies to non-localized versions of Windows 2000 / XP only).

Synchronize changes for all color depths

This button allows to synchronize all the changes you made for all color depths. When this button is pressed all the changes you made in any color depth are automatically echoed in all color depths.

Install

You can press this button to save changes, which you have made in the **Refresh rates list**.
Reboot will be required in order to apply changes.

Tips:

• Due to Windows 2000/XP kernel specific all fullscreen applications set the first refresh rate from the list of supported refresh rates when switching to fullscreen mode. This kernel specific causes well-known 60Hz refresh rate problem. Luckily, many display drivers for Windows 2000 / XP allow user to customize the list of supported display modes/refresh rates. This allows you to edit the list and change the first supported refresh rate for each display mode.
Unfortunately, it's impossible to change the order of supported refresh rates in this list, so the only way to work around 60Hz problem is to remove this refresh rate from the list at all. In this case the next supported refresh rate (usually 70Hz) will be used for all fullscreen applications. If you want to increase it further, you'll need to remove this refresh from the list too. This means that in order to set 100Hz refresh rate in all fullscreen applications you will need to remove all refresh rates below 100Hz from the list.
• RivaTuner uses Win32 API in order to fill the list of supported display modes and refresh rates. It means, that it may detect the maximum supported refresh rate incorrectly if you have not properly installed the drivers for your monitor. RivaTuner will display warning message if your monitor isn't configured properly (this applies to non-localized versions of Windows 2000 / XP only).
• Avoid removing 60Hz refresh rate for those display modes, which you are using for TV-out. This may block TV-out functionality.
• Fixing 60Hz problem may cause abnormally low scores in 3Dmark 2001 (SE)/Game 4/Nature test and some other applications. This issue is caused by the Detonator's display mode switch related bug, which was fixed in the Detonator 40.xx. This results in jerky framerate in certain applications if display mode hadn't been switched at least once after reboot. When 60Hz refresh rate fix is not installed any Direct3D application (e.g. 3DMark) switch display mode when switching to fullscreen mode and setting 60Hz refresh rate (except for those, who use 1024x768x32, 60Hz for desktop). After installing refresh rate fix 2D/3D display modes and refresh rates can match so the Direct3D can just skip display mode switching and result in this problem.
As a workaround for this problem you may either upgrade your drivers to version 40.xx or switch display mode manually before running 3DMark.
As an alternative, you may use RivaTuner's **launcher** and associate any 16-bit display mode with 3DMark.

Uninstall

This button allows you to uninstall RivaTuner's 60Hz fix. It is inactive if you haven't installed the fix.

Tips:

• It's strongly recommended to uninstall the fix before installing new drivers.
• Both **Custom display modes / refresh rates wizard** and **60Hz refresh rate fix wizard** use backup-rollback strategy for uninstalling. Please ensure that you use the proper uninstall sequence if you use both of these tools simultaneously.

Monitor limits

This window displays the monitor's frequency limits. This information can be useful when creating custom display modes and adding custom refresh rates. RivaTuner reads frequency limits from the EDID so this information will be unavailable if your monitor is not DDC capable or the driver failed to store EDID in Windows registry.
The frequency limits may include the maximum **pixel clock** (MHz), minimum / maximum **horizontal scan frequencies** (KHz) and minimum / maximum **vertical scan frequencies** (Hz).

Custom mode width

In this field you can enter width of the custom mode in pixels. This value must be within [320,2048] interval.

Tips:

• You will not be able to press the **Add** button until you enter the valid value in this field.

Custom mode height

In this field you can enter height of the custom mode in pixels. This value must be within [200,2048] interval.

Tips:

• This field is grayed and calculated automatically if **4 : 3**, **5 : 4** or **19 : 0** aspect ratio is selected. Use **Custom** aspect ratio to edit this field manually.
• You will not be able to press the **Add** button until you enter the valid value in this field.

Predefined custom modes list

This list allows you to select one of the most frequently used custom display modes.

Tip: You can customize the list of predefined modes by editing the **[PredefinedCustomModes]** section in the **RivaTuner.cfg** file.

Standard mode

This button allows to select standard display modes, which are already enumerated by Windows. This feature is useful for adding custom refresh rates for already existing display modes.

Aspect ratio

These radio-buttons allow you to select the width / height aspect ratio for the custom display mode you are creating. You may either select the **Custom** aspect ratio to change both mode dimensions manually or select **4 : 3**, **5 : 4** or **19 : 0** aspect ratio for automatic custom mode height calculation. In this case RivaTuner automatically recalculates it each time, when you change the **Custom mode width** field.

Bit depth

This group of options allow to select a combination of supported bit depths for the custom display mode you are creating.

Tip: You will not be able to press the **Add** button until you select at least one bit depth.

Refresh rates list

This edit field allows you to enter the list of supported refresh rates for the custom mode you are creating. You can either enter a single value (e.g. 60) or the list of values, separated with spaces (e.g. 60 70 72 75).

Tips:

• You will not be able to press the **Add** button until you enter a numeric value or the list of numeric values in this field.
• You can press the **Standard refresh rates** button to populate this list automatically.

Standard refresh rates

This button allows you to populate the **Refresh rates list** automatically. If the monitor's EDID and limits are available and you've entered valid display mode dimensions you may press this button to populate the list of refresh rates with the standard values starting from 60 Hz up to the maximum supported refresh rate.

Tips:

• RivaTuner approximates back / front porch and sync width when calculating the maximum refresh rate. This may cause 1-2 Hz deviations.

Custom mode approximated refresh rate limit

This window displays an approximated refresh rate limit for the custom mode you are creating. The limit can be displayed only when the monitor's EDID and limits are available and you've entered valid display mode dimensions.

Tips:

• RivaTuner approximates back / front porch and sync width when calculating the maximum refresh rate. This may cause 1-2 Hz deviations.

Add

This button allows you to add the custom mode or refresh rate. You will be prompted to reboot the system after adding the custom mode or refresh rate.

Tips:

• This button is grayed until you enter valid mode dimensions in the **Custom mode width / height** fields, select at least one bit depth and fill the **Refresh rates list** field.
• RivaTuner creates backup of the original set of display modes and refresh rates when you press this button first time. You may always press the **Uninstall** button in order to restore the original set of supported display modes and refresh rates.
• Take a note that not all the custom modes and refresh rates can be accepted by the drivers. You can use the **Uninstall** button if the driver failed to accept it.

Uninstall

This button allows you to uninstall all the custom modes and refresh rates you've created. You will be prompted to reboot the system after uninstalling the custom modes. This button is inactive if you have not created any custom modes.

Tips:

• It's strongly recommended to uninstall the custom modes before installing new drivers.
• Both **Custom display modes / refresh rates wizard** and **60Hz refresh rate fix wizard** use backup-rollback strategy for uninstalling. Please ensure that you use the proper uninstall sequence if you use both of these tools simultaneously.

Defaults

This button allows you to restore all the setting to their default values.