RivaTuner Statistics Server v7.3.3

Initially designed as a small helper application for RivaTuner graphics card utility, RivaTuner Statistics Server became de-facto framerate monitoring, On-Screen Display and high-performance videocapture service provider for other graphics card utilities. Please visit http://www.guru3d.com to get more information about the product and download new versions

System requirements:

- Windows XP, Windows Vista, Windows 7, Windows 8, Windows 8.1, Windows 10 or Windows 11 (both x86 and x64 versions) with full administrative rights
- Any DirectX, OpenGL or Vulkan compatible graphics card[1]

Features:

- The server provides framerate and frametime monitoring support to the client applications. Framerate and frametime statistics is being collected for DirectX, OpenGL and Vulkan applications. The statistics can be rendered in On-Screen Display or provided to client applications connected to the server.
- The server provides 3D acceleration usage statistics to the client applications. The clients can use the statistics to determine if any 3D applications are currently running and apply different hardware profiles depending on it.
- The server provides On-Screen Display support to the client applications. The clients can display any text info in the On-Screen Display in DirectX, OpenGL and Vulkan applications. The server can be also used as a standalone framerate monitoring solution and display own framerate statistics in the On-Screen Display.
- The server provides desktop and in-game screen capture support to the client applications. BMP, PNG and JPG screen capture formats are supported.
- The server provides high-performance realtime desktop and in-game video capture support to the client applications. Uncompressed, high-performance custom RTV1 and native MJPG video encoding, encoding with third-party external VFW compatible codecs (e.g. Lagarith or x264vfw) and hardware accelerated H.264 encoding via Intel QuickSync, NVIDIA NVENC and AMD VCE are supported in conjunction with wide range of additional video capture related options, multisource stereo and multichannel (for Window Vista and newer) audio capture with Push-To-Talk support. The functionality of expensive commercial video capture products is now available to everyone absolutely for free! There is no need to buy dedicated video capture software anymore!
- Framerate limiting support. Limiting the framerate during gaming can help to reduce the power consumption as well as it can improve gaming experience due to removing unwanted microstuttering effect caused by framerate fluctuations.
- User extendable architecture. You may express your creativity and design your own skins for RivaTuner Statistics Server, create localization for your native language, use the server to display any custom text in On-Screen Display directly from your own application and many, many more!

[1] Desktop NVIDIA or AMD graphics cards are recommended for maximum compatibility

Known limitations:

- RivaTuner Statistics Server can be incompatible with some third-party On-Screen Display software (e.g. XFire or Steam In-Game Chat). The limitation is not specific to our product, many third-party products are designed to be the only On-Screen Display products in the system and to block their own functionality when any other On-Screen Display software is running
- Anticheat systems of some online games may restrict On-Screen Display usage and block connection to the server when the RivaTuner Statistics Server is running
- RTV1 encoder performance in 64-bit applications is currently lower than in 32-bit applications
- Stealth hooking mode is currently not supported in 64-bit applications, so it is strongly not recommended to run other 64-bit On-Screen Display software in conjunction with RivaTuner Statistics Server

Revision history:

Version 7.3.3

- Added compatibility profile switch allowing ignoring nested IDXGISwapChain1::Present1 -> IDXGISwapChain::Present hook calls. This feature is intended to bypass issues with Epic Games Social Overlay, which is adding extra presentation call to application's rendering flow and causing wrong cumulative framerate to be monitored
- Added debug/compatibility profile switch allowing disabling IDXGISwapChain1::Present1 hooking
- Added On-Screen Display shared memory interface unlocking logic to OverlayEditor plugin. Now the plugin is detecting cases when On-Screen Display shared memory stays locked too long (e.g. due to crash in other On-Screen Display client application) and forcibly unlocks it
- Improved protection for pending hook library copy operations. Now RivaTuner Statistics Server is not allowed to be started without rebooting the system if installer schedule pending 64-bit hook library copy operation on the next reboot
- Updated profiles list

Version 7.3.2 (published on 17.11.2021)

- Recently introduced Direct3D11 swapchain latching mode is no longer enabled by default, now it is profile based and applied in compatibility profiles only when it is really necessary (e.g. in Microsoft Flight Simulator 2020, which may simultaneously use multiple swapchains). Enabling it globally could cause the On-Screen Display to disappear on *<Alt>+<Tab>* in some games with bugged renderers, which leak swapchain during display mode switch
- Fixed Windows XP/Vista compatibility regression, introduced in the previous version due to adding Windows 7+ specific in-process RAM usage performance counters
- Fixed issue in video capture module, which could cause some 64-bit applications to crash when trying to capture video encoded with external VFW codecs
- Improved compatibility with multithreaded Direct3D12 applications, which concurrently create new swapchains while presenting a frame on a swapchain from different thread (e.g. World of Warcraft)
- Framerate limiter's passive waiting mode, introduced in the previous version, is no longer power user oriented. Now it is available in GUI under compatibility properties, so you can enable passive waiting if you prefer reduced CPU load and power consumption or disable it if you prefer maximum framepacing precision
- Various compatibility improvements in the hook engine:
  - Added hooking support for Microsoft DirectX 12 Agility SDK based Direct3D12 applications (e.g. Halo Infinite insider tech preview and possibly other future Direct3D 12 applications compiled with Agility). New DirectX 12 Agility model assumes that the game can be shipped with a local copy of Direct3D 12 runtimes, which can be newer than your system Direct3D12 runtimes. By default RivaTuner Statistics Server's hook engine is configured to block injection into any custom Direct3D runtimes located outside OS system folder because such case is typical to Direct3D proxy libraries used in third party game mods, which are frequently fundamentally incompatible with overlays. So hooks were blocked on purpose in such environment, making overlay invisible. Running RivaTuner Statistics Server in such environment also reduced performance due to periodically repeating and failing overlay injection attempts. Previously this could be solved by creating application profile for such game with enabled *"Custom Direct3D support"* option, which is intended to allow injecting custom Direct3D runtimes located outside system OS folders. New Agility compatible hooking path automatically addresses it in the following way:
    - Simplified form of *"Custom Direct3D support"* mode is now internally engaged by RivaTuner Statistics Server when Agility SDK based Direct3D12 application is detected. Full *"Custom Direct3D support"* mode functionality is overabundant and not necessary for Agility case. New Agility compatible hooking path is optional and can be disabled by power users at application profiles level for troubleshooting or performance testing
    - Added retry counter for reinjection attempts, aimed to minimize performance penalty for situations when Agility SDK based Direct3D12 application cannot be injected
    - DXGI swapchain hooks are now suspended during dynamic hook offsets initialization, this change is aimed to reduce risk of incompatibilities caused by enabling *"Custom Direct3D support"* profile in conjunction with application detection level set to *"High"* for Agility based Direct3D12 applications
  - Added debug/compatibility profile switch allowing disabling hook reinjection mode
  - Added debug/compatibility option allowing disabling hook integrity control mode and enabling hook entry point relocation
  - Changed unconditional injection delay handling approach. Now it overrides conditional trigger module based delay,
  - previously unconditional delay didn't affect the cases when conditional delay was triggered
  - D3D12 command queue hook is affected by delayed injection now, but the rest D3D12 swap chain creation hooks are still injected immediately and ignore any delays by default

  - All API hook handlers have been slightly refactored to allow switching between different API hooking implementations
  - Added new *"Use Microsoft Detours API hooking"* option, which is allowing RivaTuner Statistics Server to switch to Microsoft Detours API hooking library instead of own API hooking engine. This option doesn't help RivaTuner Statistics Server itself, but it may help to fix other third party applications which also hook 3D API calls and use vanilla Microsoft Detours for that (e.g. OBS 27.1.0 and higher). So you may try to enable it you're using RivaTuner Statistics Server with some third party overlay or videocapture software and it refuses to work
- Added unified Direct3D12 command queue caching based algorithm for handling periodic swapchain recreation in some Blizzard games (e.g. Diablo 2 : Resurrected and World of Warcraft)
- Improved On-Screen Display hypertext formatting implementation:
  - Fixed covering extent calculation for layers with embedded objects resized to layer extent
  - Improved dynamic tabbing implementation. Dynamic tabulation value is no longer global for whole hypertext, now each layer calculates it independently
  - Added sticky layer position tags. Sticky layers are intended for displaying latency markers for luminance sensor based systems similar to LDAT, which are expected to be displayed in a fixed position unaffected by On-Screen Display origin selection
  - Improved concept of cursor position placeholder layers. Hypertext is no longer being automatically appended with LF symbol when the last rendered symbol is a backspace. Such approach allows using cursor position placeholder layers to define exact desired position of text output for the next hypertext clients instead of defining position one line above it
  - Added new *<RES>* hypertext tag, allowing displaying framebuffer resolution in On-Screen Display
  - Added new *<ARCH>* hypertext tag, allowing displaying application architecture info (x64 or UWP) in On-Screen Display
  - Added new *<API>* hypertext tag, allowing displaying application 3D API info in On-Screen Display. This tag is quite similar to *<APP>* tag, but unlike existing tag it displays full 3D API name instead of abbreviated one (e.g. OpenGL instead of OGL)
  - Added new *<TIME=format_string>* hypertext tag, allowing displaying custom formatted system date/time in On-Screen Display
  - Added dynamic color formulas support for <C> hypertext tag
  - Added GUIDs to internal graph autoscaling cache implementation. GUIDs ensure that autoscaling is not applied to wrong graph when you switch between different OSD layouts containing different sets of displayed graphs
  - Added new <GRMIN> and <GRMAX> hypertext tags, aimed to display minimum and maximum limits of the last embedded graph. New tags are intended to be used in conjunction with autoscaling graphs to display their dynamic limits. Please pay attention to hypertext rendering Z order and ensure that you specify the tags after rendering target embedded graph to make them work as expected
  - Improved load image <LI> tag format. Now the tag parameter can be wrapped with quotation marks if the filename contains some reserved symbols (round brackets, triangle brackets and comma)
- Improved SDK:
  - Improved RivaTuner Statistics Server host API:
    - Added helper PickColorEx API function, allowing the plugins to use extended version of host color picker dialog window. Extended version may display *">>"* button, allowing the clients to implement some additional color adjustment functionality
    - Added helper GetColorPreview function, allowing the plugins to generate RGBA color preview in client specified image buffer
  - Improved shared memory layout:
    - Now each running 3D application entry contains current framebuffer resolution info. This feature is required to allow displaying foreground application resolution info in DesktopOverlayHost
    - Additional extended 32KB hypertext slot is now available to OSD client applications
  - Improved HotkeyHandler plugin:
    - Added workaround for internal DirectInput issue, which could cause hotkey handler to stop processing hotkeys correctly after locking/unlocking PC from keyboard with <Ctrl>+<Alt>+<Del> or <Win>+<L>. To bypass it the plugin is resetting hotkey handler state after lock screen transition now
    - Added optional RawInput based hotkey handler implementation
  - Improved OverlayEditor plugin:
    - Added layers list editor window. You can access it via *"Edit list"* command in *"Layers"* menu or via *<Ctrl>+<Shift>+<L>* keyboard shortcut to adjust layers Z-order visually with drag and drop or select desired layer
    - Added *<Apply>* button to *<Layer properties>* window. Now you can test your changes without closing the properties
    - Added sticky layer position options to *<Layer properties>* window
    - Added *%Date%* macro for embedding system date into hypertext
    - Added maximum CPU core load data source to HAL. Sample overlay is displaying maximum CPU core load history graph under per-core CPU load barchart graphs
    - Decreased timer resolution to improve editor's framepacing
    - Introduced concept of dynamic color attributes, which brings MSI Afterburner's OSD alarm thresholds feature support natively to the plugin:

            - Any color attribute (e.g. layer text or background color) can be switched to dynamic mode by pressing *">>"* button inside the color picker dialog or switched back to static color mode by pressing *"<<"* button
            - Dynamic mode is reflected by *"D"* letter displayed on top of color preview box
            - Dynamic colors can be linked with any data source visible to the plugin, can have up to 5 open or closed data source value ranges mapped to different colors
            - Dynamic colors can either select fixed color by range or blend colors for nearby ranges allowing implementing smooth gradients for dynamic color changes if necessary
            - Dynamic colors can be either calculated inside the plugin and applied to formatted hypertext as static color on the fly or you can optionally embed dynamic color formula into the hypertext layer. In this case embedded objects (e.g. graphs) can use this formula to recalculate colors independently for each displayed graph point
        - Added new *%CPUShort%* macro. This macro definition is similar to previously existing *%CPU%* macro, containing compacted CPU branding string, but *%CPUShort%* additionally strips CPU clock frequency info starting from @ symbol
        - Text table specific color attributes are now displayed in recent colors selection panel in color picker dialog
        - Text table line name and cell text can use special symbols now (e.g. newline \n symbol)
        - Hypertext edit field in layer properties dialog is now multiline, so it is more convenient to work with complex layers (e.g. layers combining multiple embedded objects)
        - Fixed issue which could cause skipping some overlay refresh iterations
        - Added new keyboard shortcut to the overlay editor window. Now you may press *<Ctrl>+<Shift>+<M>* to apply layout master settings. The same can be done from menu: *Layouts -> Edit -> Master settings*
        - Now all embedded graphs are using buffered update approach instead of asynchronous update for each graph before. This change is aimed to eliminate risk of seeing short flickering when switching between different overlay layouts containing different sets of displayed graphs
        - Overlay layout timer is now reinitialized properly when switching between different overlay layouts with hotkeys
        - Sample overlay layout supplied with the plugin was designed as a technodemo, showing you as many complex overlay creation techniques as it is possible. Now the plugin includes two more built-in overlay layouts, which suit better for everyday usage:
            - Classic layout is close to classic native MSI Afterburner's layout. It combines traditional text table based sensors representation with frametime graph and contains the most frequently used and the most useful sensors including process specific ones
            - Benchmark layout is entirely focused on framerate/frametime monitoring. It displays autoscaling frametime graph and dynamic distribution of the slowest frames with highlighted 1% zone. Such representation gives better visual demonstration of 1% low framerate to beginners. Dynamic distribution of the slowest frames is also useful when you're altering percentile calculation related options
    - Improved hotkey handler plugin:
        - Added DirectInput initialization mutex, aimed to prevent possible deadlock in IDirectInputDevice8::Acquire during application start when it creates one more DirectInput device for FCAT overlay working in "latency marker" mode
    - Improved RTSSSharedMemorySample sample:
        - Fixed sample crash on the systems with more than 8 logical processors
        - Now the sample demonstrates new sticky layer position tags usage
- Added new *"latency marker"* rendering mode for FCAT overlay. New mode is aimed to be used in conjunction with third party luminance sensor based input latency monitoring systems similar to LDAT. In this mode RivaTuner Statistics Server displays color latency event markers reflecting current input state when keyboard or mouse buttons are pressed or released. In this mode RivaTuner Statistics Server additionally stores high precision latency marker registration and latency marker presentation timestamps into shared memory and provides the timestamps to client applications. Client applications may use this data as a backend for input latency calculation in conjunction with additional data they receive from luminance sensor
- Added new *"Percentile buffer"* option to general properties. New option allows switching between unlimited and rolling ring buffer modes for 1% low and 0.1% low metrics calculation. Unlimited mode is preferred if you manually start benchmarking session with a hotkey. Ring mode can be preferred if you permanently keep the benchmark mode enabled and want to see 1% and 0.1% low metrics reflecting just a few last seconds of gameplay
- Slightly altered geometry batching implementation in On-Screen Display renderer. Previously RivaTuner Statistics Server internally represented On-Screen Display contents as a few uber batches, containing all On-Screen Display geometry grouped by primitive type (for example, the first batch containing triangles representing all On-Screen Display symbols, solid color bars, and embedded images and the second batch containing all lines used to render all graphs). Such batching provides optimal rendering performance and minimizes draw calls count and rendering pipeline state changes. With such implementation it was possible to render On-Screen Display contents in expected Z-order for the same type of graphics primitives (e.g. for text, images or solid color bars), but line primitives were always rendered on top of everything. New altered geometry batching implementation allows splitting uber batches on smaller batches, so it is now possible to render both lines and triangle primitives in expected Z-order. Please take a note that batches splitting mode is enabled by default, but power users may enable old uber batch rendering mode at profile level if necessary

- Improved scanline sync implementation:
    - Now index of target display device for multimonitor systems is also displayed in scanline sync info panel
    - SyncDisplay profile switch, which is defining index of target display device for multimonitor systems, can be set to -1 now. In this case target display device will be selected automatically by monitor displaying 3D application's foreground window
- Added power user oriented profile switch, allowing enabling HDR specific color space conversion for On-Screen Display in D3D11/D3D12 HDR applications. This profile switch can be used to correct On-Screen Display appearance when it looks too dim in scRGB HDR applications or oversaturated in HDR10 applications
- D3D12 videocapture queue depth has been increased from 3 to 8 frames. This change is aimed to improve captured videostream smoothness in applications like Forza Horizon 4, which render more than 3 frames ahead
- Added hooking profile for Deathloop. The profile disables *"High"* application detection level setting for this game, which is incompatible with its protective system implementation. Please take a note that application detection level settings in RivaTuner Statistics Server are application compatibility options, which are intended to be used at profiles level only for games dynamically loading DirectX runtimes. It is recommended to use it only for applications, which cannot be hooked with default *"Low"* application detection level. It is strongly not recommended to enable *"High"* level globally for all processes running in the system
- Added hooking profile for Fallout 76. Similar to Deathloop profile, it also disables *"High"* application detection level setting for this game, which is incompatible with its protective system implementation
- Added On-Screen Display profile for Diablo 2 : Resurrected
- Added On-Screen Display profile for World of Warcraft
- Added On-Screen Display profile for Forza Horizon 5
- Application tray icon is DPI aware now
- Application installer is DPI aware now
- Updated profiles list

Version 7.3.1 (published on 10.03.2021)

- Fixed compatibility regression with some Direct3D9/Direct3D9Ex applications, introduced in the previous version
- Multi-device oriented DXVK compatibility path introduced in the previous version is now optional and can be disabled at profile level
- Added user configurable unconditional delay to delayed injection system. It is no longer necessary to use tricky ways and specify one of OS kernel modules as injection delay triggers to apply delayed injection globally for all hooked applications
- Slightly improved profile templates architecture. Now the profiles allows disabling hooking for UWP version of some application while enable hooking for Win32 version of the same application
- Altered On-Screen Display profile for Forza Horizon 4. Now the profile disables hooking for UWP version of the game, but enables it for Steam version
- Added On-Screen Display profile for Tom Clancy's Ghost Recon: Wildlands
- Added On-Screen Display profile for Microsoft Flight Simulator 2020

Version 7.3.0 (published on 28.02.2021)

- Scanline sync no longer stops working in OpenGL applications when OSD support is disabled at application profile level
- Fixed Vulkan device and instance handle leak in Vulkan bootstrap layer
- Fixed crash on capturing screenshots in Cyberpunk 2077 when scRGB HDR mode is enabled
- Improved compatibility with Vulkan applications using multiple Vulkan instances and devices
- Various Vulkan bootstrap layer and Vulkan overlay renderer cleanups aimed to improve compatibility with Vulkan validation layer
- Added alternate asynchronous On-Screen Display renderer for Vulkan applications presenting frames from compute queue (id Tech 6 and newer engine games like Doom 2016 and Dooom Eternal). The implementation is using original AMD's high performance concept of asynchronous offscreen overlay rendering and the principle of asynchronous combining with framebuffer directly from compute pipeline with compute shader without stalling compute/graphics pipelines. PresentFromCompute entry in global profile template is set to 1 by default now and it enables new asynchronous implementation. The previous synchronous and more performance consuming implementation is also available and can be enabled by setting PresentFromCompute to 2. It is recommended to enable the previous synchronous implementation for performance comparisons only and for estimating difference in performance hit between new and old implementations
- Improved compatibility with multithreaded Direct3D1x applications, using multiple Direct3D devices and swapchains and concurrently presenting frames on them from different threads (e.g. Windows Terminal)
- Various compatibility improvements in the hook engine:
    - Now any application may manifest itself as incompatible with RivaTuner Statistics Server overlay/hooks either via declaring special named exported variable (recommended and preferred way for EXE-level implementation) or via setting process environment variable (alternate way for DLL-level implementation). So the applications incompatible with

    RivaTuner Statistics Server may prevent its API hooking functionality with just a couple lines of code at compile time without need to add exclusion profile for it from RivaTuner Statistics Server side
  - Added user extendable profile mapper to profile loader. The mapper is allowing RivaTuner Statistics Server to map multiple executable names matching with specified wildcard (e.g. vegas130.exe, vegas140.exe and so on for different versions of Sony Vegas) to a single profile file, so it is no longer necessary to create exclusion profiles for each version of such application independently
  - Added user extendable injection ignore triggers list. Similar to injection delay triggers list, injection ignore triggers list allows defining the set of DLL modules, which will prevent RivaTuner Statistics Server from injecting target process when any of such modules is detected in the process context. This feature is aimed to exclude applications using typical GPU accelerated GUI libraries from hooking. The list currently includes WPF framework libraries (PresentationFramework.dll and PresentationFramework.NI.dll), so all WPF applications are excluded from hooking now
  - Improved 32-bit runtime disassembler provides better compatibility with 32-bit applications when stealth mode is enabled
  - Added Direct3D12 hooking support for Direct3D12 runtime changes introduced with KB4598291
- Improved RivaTuner Statistics Server host API:
  - New localization API is allowing the plugins to use host multilanguage translation functionality for localizing the plugin's GUI
  - New On-Screen Display preview API is allowing the plugins to render On-Screen Display on top of Direct3D frambuffer specified by caller and report back each rendered layer's screen position and extent. On-Screen Display preview API can be used to implement visual On-Screen Display editing functionality
  - Added helper PickColor API function, allowing the plugins to use advanced host color picker dialog window
- Improved On-Screen Display hypertext formatting implementation:
  - Various fixes in alignment, dynamic tabbing, positioning and backspace tags parsing implementation
  - Added layer definition tags. Each layer defined inside the hypertext is treated by parser as an independent block of text and new On-Screen Display preview API is able to report each rendered layer's screen position and extent independently. Please take a note that layer definition tags also affect text extent calculation, so embedded objects with zero height and width defined inside the layer are covering just their layer area instead of whole On-Screen Display area
  - Added extent override tags support. By default the layer extent is defined by hypertext content, but you may force the layer extent to be greater than actual content extent with new extent override tag. The tag also allows aligning the layer content by left/right/top/bottom or centering it horizontally/vertically
  - Added image loading tags support. The tags are allowing the parser to load and embed a custom PNG image into On-Screen Display font texture
  - Added image embedding tags support. The tags are allowing you to render a part of On-Screen Display font texture image into hypertext. This allows you to display custom logos or animated image sequences into On-Screen Display
  - Added defaults override tags support. The tags are aimed to be used in pairs, to start and stop defaults override blocks inside the hypertext. The first tag starts the block, stores default hypertext formatting attributes (text color, size and so on) and saves currently applied hypertext formatting attributes as new defaults. This way user defined defaults are being applied with default text color or size tags inside defaults override block. The next tag ends the block and restores previously saved default hypertext formatting attributes
  - Changed embedded object size interpretation. Now positive size values are treated as zoomed pixels instead of fixed pixels
  - Added new <EXE> hypertext tag, allowing displaying hooked process executable name in On-Screen Display
  - Added range autoscaling flag support for embedded graphs
- Added new type of plugins: client plugins. RivaTuner Statistics Server is designed to act as a server process, which is running passively and providing different functionality (On-Screen Display rendering, screen and video capture, benchmarking etc) to multiple client applications connected to it (e.g. MSI Afterburner). GUI for such functionality is normally located at client application side. New client plugins allow integrating GUI for such functionality directly into RivaTuner Statistics Server, without the need to run additional client applications, so new client plugins architecture is intended for those who prefer to use RivaTuner Statistics Server as a standalone solution without MSI Afterburner. SDK is now including the following open source client plugins:
  - HotkeyHandler plugin is a built in hotkey processor, which is providing and demonstrating implementation of the following functionality:
    - Low-latency DirectInput based hotkey handler, similar to MSI Afterburner's internal one
    - Full set of standard hotkey based functionality, available in MSI Afterburner client and based on RivaTuner Statistics Server API. This includes On-Screen Display visibility control, framerate limiter activation/deactivation, screen capture, video capture and benchmarking functionality. Now all those features can be also used in standalone RivaTuner Statistics Server usage scenario
    - Up to 4 additional general-purpose programmable profile modifier hotkeys. You may bind almost any profile modification related actions to those hotkeys, e.g. increase of decrease framerate limit, toggle between multiple predefined framerate limits, increase or decrease scanline index for scanline sync mode, zoom OSD in/out, enable or disable global hooks and many more
  - OverlayEditor plugin is advanced visual hypertext editor, providing you more than any currently existing overlay can do. There were a lot of fair and independent commercial overlay reviews recently, claiming that it is absolutely impossible to develop visual overlay editing in free applications and that feature wise all free overlays are second tier products comparing to commercial ones. We decided to take it as a challenge, burst that marketing bubble and provide the plugin

with such functionality free of charge to anyone who still believes in free system software distribution principles like us. So this plugin was developed from scratch in just a few weeks. It comes with fully open source code included in SDK, so you can peek inside and improve it or just see if such kind of functionality is really a rocket science development:

  - Unified and extremely flexible overlay layout editing GUI, which can be used in combo with any major and most popular hardware monitoring cores available on the market: HwINFO64, AIDA64 or MSI Afterburner. Do you wish to attach something else, e.g. GPU-Z? That's not a problem, it can be easily done by any beginner developer in just a few hours because the plugin is completely open source. Your overlay layout can be easily adapted to any hardware monitoring data provider application and it will look exactly the same and stay exactly the same feature rich. With such modular architecture overlay rendering and editing application is completely decoupled from hardware monitoring core so it is extremely easy to troubleshoot hardware monitoring related compatibility issues and switch to a different hardware monitoring data provider application. Such modular architecture is a pro, not a con
  - The plugin provides internal data sources, which are measured by RivaTuner Statistics Server itself (e.g. framerate, frametime, minimum/maximum/average/1%/0.1% low framerates for benchmark mode). Such data sources can be used in overlay layout as is, without running any additional hardware monitoring data provider application in background
  - The plugin provides internal HAL (Hardware Abstraction Layer) and minimalistic built-in hardware monitoring core based on it. The HAL provides basic CPU and RAM usage related data sources and architecture specific GPU related monitoring data sources (such as graphics engine usage, videomemory usage, graphics processor and videomemory clocks, temperatures, power consumption, fan speeds and so on) available on modern discrete AMD and NVIDIA GPUs and integrated Intel GPUs. Internal HAL do not rely on low-level access to hardware in any form, it uses only native GPU monitoring functionality embedded in OS and GPU display drivers: NVAPI for modern NVIDIA GPUs, AMD ADL for modern AMD GPUs and D3DKMT for Intel iGPUs and legacy GPUs. Such approach don not break our main principle of decoupling overlay from hardware monitoring application, so HAL data sources have close to zero chances to affect hardware compatibility and they can be safely used in conjunction with any external data provider application of your choice
  - The plugin provides internal ping monitoring data source, implementation is a direct clone of MSI Afterburner's ping monitoring plugin. Ping monitoring is not natively available in HwINFO64 or AIDA64, so this internal data source is aimed to help those who plan to use this plugin without MSI Afterburner in combo with HwINFO64 or AIDA64
  - The plugin provides an ability to import data sources from Windows performance counters, implementation is a direct clone of MSI Afterburner's PerfConter plugin. Windows performance counters provide you built-in monitoring of HDD usage and read/write speeds, network interface download and upload rates, a lot of detailed global and per-process CPU, memory and pagefile related metrics as well as hundreds of other performance counters visible to OS
  - The plugin provides MSI Afterburner styled correction formulas for all external data sources. Which means that you can transform data received from external hardware monitoring data provider applications in any form, for example convert memory usage from megabytes to gigabytes and many more
  - The plugin provides layer based overlay layout representation, which is typical for graphics editors. Such architecture suits best for creating more complex and artistic On-Screen Display layouts, but requires a bit more time and efforts
  - Each layer is independently formatted and positioned hypertext block, which can display both static and dynamic hypertext. In addition to standard customization of each layer's hypertext formatting attributes (text and background color, font size, subscript or superscript text size, text alignment etc), you can also embed the following objects directly into hypertext of each layer:
    - Rich set of different macro definitions, displaying system time in different formats, programmable timers, your PC hardware specs and of course current values of internal plugin data sources (e.g. framerate or frametime) of external data sources exported from different hardware monitoring applications (e.g. GPU temperature exported from MSI Afterburner)
    - Traditional, diagram or barchart styled graphs attached to any data source, either internal or external one
    - Static images, displaying your hardware manufacturer logos and so in
    - Dynamic animated images, changing depending on data source connected to animation input. This way you can create absolutely uniquely looking graphics indicators in your overlay layouts, e.g. round progress indicators, gauges etc
    - You're not limited to embed just a single object per layer. This means that for example you may create a layer containing all CPU load barchart graphs and make them share the same settings template, which makes it much easier to modify such overlay layouts
  - Complete freedom of choice, you can still customize On-Screen Display layout via built-in GUI of MSI Afterburner, HwINFO64 or AIDA64. Use new plugin only if you need it and wish to create something more complex than native MSI Afterburner or HwINFO64 On-Screen Display layout
- Added asynchronous process specific performance counters access interface. The interface is integrated into hooks library, which is running inside protected process context and may report hooked process specific RAM and VRAM usages to client applications

- Added new open source DesktopOverlayHost tool to SDK. DesktopOverlayHost is a simple blank borderless 3D window with adjustable size, position, transparency and chroma keying support. You can use it as a platform for displaying any 3D API hook based overlay right on top of your Windows desktop. Implementation is overlay vendor agnostic, so you can use it with RivaTuner Statistics Server, as well as with other third party overlays like EVGA Precision X1 and so on
- Added ShowForegroundStat profile switch, which is allowing any 3D application to display foreground 3D process framerate and frametime statistics instead of application's own ones. This switch is used by new DesktopOverlayHost tool profile to let it to display foreground 3D process statistics on desktop
- Added RenderDelay profile compatibility switch, allowing delaying On-Screen Display initialization and rendering for some applications when it is necessary (e.g. Resident Evil 3 in Direct3D11 mode when HDR mode is enabled)
- FCAT overlay update rate is no longer limited to overlay content update rate in offscreen overlay rendering mode
- Added new hypertext tags for displaying process specific 1% low and 0.1% low framerate graphs. Now multiple simultaneously running 3D applications can display their own independent 1% low and 0.1% low framerate graphs instead of foreground 3D application's 1% low and 0.1% low framerate graphs in the previous version
- Improved 1% low and 0.1% low framerate graphs rendering implementation for graph and diagram rendering modes. The graphs showing the history of 1% low and 0.1% low framerate changes has no statistical sense, so now RivaTuner Statistics Server is showing you more informative dynamic histogram of sorted and the most slowest immediate framerates with highlighted 1% or 0.1% areas on it. For barchart rendering mode 1% low and 0.1% framerate graphs still display the current value like before
- Added new *"moving bar"* rendering mode for FCAT overlay. New mode is aimed to simplify visual identification of tearline position and it can be used when calibrating Scanline Sync settings
- Added new *"Frametime calculation point"* option to *"General"* application properties. This option is aimed to help those who try to directly compare frame rendering start timestamp based frametimes with frame presentation timestamp based frametimes. Please refer to new option context help to get more detailed info
- Added new *"Percentile calculation mode"* option to *"General"* application properties. This option is aimed to help those who try to compare different implementations of 1% low and 0.1% low framerate metrics in different applications. Please refer to new option context help to get more detailed info
- Added new *"Framerate limiting mode"* option to *"General"* application properties. Two alternate framerate limiting modes selectable with this option (*"front edge sync"* and *"back edge sync"*) are intended to be used in conjunction with scanline sync mode. Using those options in tandem allows enabling so called hybrid scanline sync mode. In this case actual target scanline synchronization is performed just once for initial tearline positioning then tearline position can be steered with high precision system clock. This option can also help those who try to compare flatness of frametime graphs measured at different points (frame start vs frame presentation timestamp based)
- Added 3 more decimal digits to fractional framerate limit adjustment control. Extended framerate limit adjustment precision can be necessary for new hybrid scanline sync mode, where the previous 0.001 FPS adjustment precision can be insufficient
- Added fractional frametime limit adjustment support. Extended frametime limit adjustment precision can be necessary for new hybrid scanline sync mode, where the previous 1 microsecond adjustment precision can be insufficient
- Now you may hold <Alt> and click framerate limit adjustment control to set framerate limit to your refresh rate
- Now up/down spin buttons associated with framerate limit adjustment control tune the limit in minimum adjustment step instead of fixed 1 FPS step (e.g. in 0.1 FPS step if single decimal digit after comma is specified)
- Added power user controllable passive wait stage to framerate limiter's busy waiting loop. It is aimed to help those who is ready to sacrifice timing precision in favor of lower CPU load
- Improved power user oriented scanline sync info panel. New performance counters are aimed to improve the process of scanline sync calibration and help you to diagnose tearline jittering. The following new performance counters have been added to it:
  - Sync start – index of scanline where 3D application called 3D API frame presentation function and scanline sync engine started the process of waiting for the target scanline
  - Sync end – index of scanline where scanline sync engine ended the process of waiting for target scanline. Ideally it must be as close to expected target scanline as it is possible
  - Present – index of scanline where 3D API frame presentation function was actually called after performing scanline synchronization. For normal scanline sync modes it is pretty close to the previous performance counter. For hybrid scanline sync mode it can drift depending on your framerate limit, if it doesn't match with your display refresh rate
  - Present latency – time spent inside 3D API frame presentation call
- Updated hardware encoding plugins:
  - All hardware encoding plugins are using new host localization API, so all plugins support multilanguage settings GUI
  - Updated Intel QuickSync H.264 encoder plugin. Now you may manually select target display device in the plugin's settings. Please take a note that manual device selection can be required on Intel DCH drivers to address problems with wrong automatic Intel iGPU device selection, which could prevent the encoder from working properly
- Added alternate and user configurable CPU yielding implementation to busy wait loops used in both framerate limiter and scanline sync implementations. Alternate CPU yielding implementation is used by default now, it can improve previously existing and close to ideal framepacing accuracy even further under heavy CPU load conditions due to minimizing context switching related timing penalties
- Added alternate named pipe interface for streaming frametime statistics to third party applications in real time
- Updated profiles list

Version 7.2.3 (published on 18.09.2019)

- Updated digital signature for the installer and the main executable. Considering that whitelisting new digital signature in anticheat systems may take significant amount of time, the previous whitelisted MSI GlobalSign digital signature is still being used for hook libraries, media encoding core and plugins in order to provide continuous gaming experience to the community
- Interoperability D3D11 page flips on some systems are now filtered by framerate calculation module in OpenGL/Vulkan applications. This change should fix issues with On-Screen Display flickering and double displayed framerate in new experimental Vulkan codepath in No Man's Sky under certain conditions
- Added workaround for Intel HD Graphics Direct3D driver bug, causing some parts of On-Screen Display to be invisible or rendered in solid black color in Direct3D8 and Direct3D9 applications on integrated Intel GPUs
- Added Vulkan bootstrap layer for Vulkan On-Screen Display implementation. Now On-Screen Display in Vulkan applications is supported for dynamic RivaTuner Statistics Server start scenario (i.e. when you start RivaTuner Statistics Server while target Vulkan application is already running). Bootstrap layer usage also eliminates the need of setting application detection level to high for some Vulkan applications loading Vulkan runtimes dynamically (e.g. Dota 2 or Rage 2)
- Improved implementation of hook library connection to shared memory. New implementation improves On-Screen Display compatibility with some third party applications (e.g. PPSSPP) for dynamic RivaTuner Statistics Server start scenario
- Improved raster 3D font rendering quality for some fonts with overhang symbols
- Improved compatibility with some Direct3D12 applications, which dynamically recreate swapchains during changing graphics settings (e.g. the most recent patch for Metro Exodus during toggling DLSS on/off)
- Added On-Screen Display profile for Apex Legends. The profile is adding the game's executable to delayed injection system triggers list in order to improve compatibility with customized implementation of Origin In-Game overlay used in Apex Legends
- Improved function offsets cache validation algorithm in hooking system. The improvements are aimed to minimize the risk of cache corruption:
  - Now RivaTuner Statistics Server is zeroing all cached offsets prior to offsets reinitialization after detecting new DirectX runtimes installation
  - Now RivaTuner Statistics Server is forcibly resetting and rebuilding the cache during 3 subsequent application restarts after detecting new DirectX runtimes installation
  - Cache version has been incremented to force it to be reset on all systems
- Improved skin engine:
  - Optimized skin rendering performance for skins using multiple overlapped animated indicators
  - Added position smoothing support to skinned indicators
  - Now skinned applications can forcibly disable indicator animation if necessary
  - Skin format reference guide has been updated to v1.8 to document these changes

Version 7.2.2 (published on 09.03.2019)

- Changed digital signature to fix issues with revoked StarCom code signing certificate on some systems. RivaTuner Statistics Server is temporarily signed by MSI GlobalSign digital signature
- Font size in the profiles list is now scaled properly when adjusting skin scaling ratio

Version 7.2.1 (published on 07.03.2019)

- Fixed issue with the latest NVIDIA Adaptive Shading update of Woflenstein II: The New Colossus locking on startup. Please take a note that now Woflenstein II: The New Colossus is presenting frames from compute queue in NVIDIA codepath similar to AMD codepath. Due to that reason, On-Screen Display rendering is performance expensive and disabled by default in such environment. Experienced users, understanding and accepting On-Screen Display related performance penalty, may reenable it with PresentFromCompute profile switch
- Fixed black screen issue in Vulkan applications on NVIDIA Kepler GPU family under 4xx.xx series drivers family
- Fixed multhithreaded active busy-wait loop based frame capture implementation, which could prevent desktop videocapture from working in the previous version
- Improved compatibility with multithreaded Direct3D1x applications, using multiple DXGI swapchains and concurrently accessing them from different threads (e.g. Microsoft Photos UWP application)

- Improved On-Screen Display hypertext formatting implementation:
    - Variable slot 250 is no longer used to define own On-Screen Display statistics format, now it is allowing the client applications to redefine default text format. Such implementation allows the client applications like MSI Afterburner to edit default text color and size directly inside the On-Screen Display layout editor
    - Improved subpixel precision for raster fonts provides better font scaling quality and improves rendering quality for small superscript/subscript-styled text
    - Improved font baseline positioning. Now subscript-styled text is aligned on the baseline of maximum sized font symbol in the line instead of the baseline for 100% font size
    - Improved font scaling implementation for large fonts. Now requested font size is scaled from the closest reference prerendered font size instead of 100% font size
    - Changed set of prerendered font sizes from 40%, 50%, 80% and 100% to 50%, 100% and 200%. Prerendered font sizes can be used as is without scaling and with the best possible rendering quality, any different font sizes are interpolated
    - Added background fill flag for embedded objects. Background fill is intended to simplify maximum value visualization with barchart-styled embedded graphs
    - Now it is allowed to use zero width or height in bar embedding tags to specify whole text extent
    - Embedded objects are now affected by text size and alignment tags
    - Added backspace symbols support. Backspaces are intended to be used in conjunction with bar embedding tags. Inserting backspace symbol immediately after the bar embedding tag results in returning the cursor position back to embedded bar origin. Such implementation provides simplified way of embedding custom background in the hypertext. Please take a note that backspace sequences are currently not supported, so the cursor cannot be moved beyond the previous position
    - Added new tag for displaying benchmark time
- Added suspending support to VTable hook handlers. Suspending is used to improve reference counter tracking implementation and make it more future proof
- Graphs are no longer displayed in On-Screen Display when Vector2D On-Screen Display rendering mode is selected
- Slightly refactored Direct3D and Vulkan On-Screen Display rendering codepaths due to implementation of optional offscreen rendering mode. This mode is currently disabled by default, it is reserved for future versions as a codebase for asynchronous On-Screen Display rendering implementation
- Multilanguage user interface library is DPI aware now. Main window skinned interface is no longer being scaled by OS and no longer looks blurred by default when greater than 100% DPI is selected. Now main skinned window interface can be manually zoomed with skin scaling slider in *“User interface”* tab, and the properties are automatically scaled by OS according to selected DPI
- Added new skin scaling mode for power users. New mode provides better scaling performance at the expense of some scaling image quality loss
- Skin scaling implementation is now asynchronous, so increasing skin scaling ratio doesn’t increase GUI update time and doesn’t decrease GUI response time
- Skin scaling slider range has been extended from 75%-150% to 75%-200%
- Some profile settings are treated as power user oriented and they are locked by default at global profile level now. Such settings include *“Enable Custom Direct3D support”* compatibility option and Vector2D On-Screen Display rendering mode selection. Context help was not recommending enabling those compatibility related options globally, however many users tended to enable them blindly then blamed the product for inappropriate performance or reduced 3D applications compatibility. Such settings are now ignored and locked for enabling in global profile unless you manually unlock them
- Added *“Enable framerate limiter”* option to general application properties. This option allow controlling RivaTuner Statistics Server’s framerate limiting support globally, so you may temporarily disable it if necessary without modifying generic framerate limiter or scanline sync related settings in the profiles. The client applications like MSI Afterburner can also distantly control this option via hotkeys
- Now RivaTuner Statistics Server validates DirectX runtimes installation at startup and displays warning message if required DirectX runtime components are missing. Power users may disable this warning message as well as any previously existing diagnostic warning messages displayed at RivaTuner Statistics Server startup and enable silent startup mode via the configuration file if necessary
- Added user extendable exclusions list for desktop windows notification performed during installation and uninstallation of system CBT hooks. Currently the only exclusion in the list is Microsoft Outlook process, which previously tended to display help popup window on closing RivaTuner Statistics Server
- Now RivaTuner Statistics Server hides own frametime history overlay when framerate or frametime is displayed by the client On-Screen Display application
- Now RivaTuner Statistics Server hides own benchmark statistics when benchmark time, minimum, average, maximum, 1% low, 0.1 low framerate is displayed by the client On-Screen Display application
- Added On-Screen Display compatibility profile for Splinter Cell Chaos Theory
- Updated profiles list

Version 7.2.0 (published on 30.10.2018)

- Added On-Screen Display performance profiler. Power users may enable it to measure and visualize CPU and GPU performance overhead added by On-Screen Display rendering. Two performance profiling modes are available:
    - Compact mode provides basic and the most important CPU prepare (On-Screen Display hypertext formatting, parsing and tessellation), CPU rendering and total CPU times, as well as GPU rendering time (currently supported for Direct3D9+ and OpenGL applications only)
    - Full mode provides additional and more detailed per-stage CPU times
- Improved built-in framerate limiter:
    - Fractional framerate limit adjustment functionality is no longer power user oriented, now you may specify fractional limit directly from GUI
    - Now you may click *"Framerate limit"* caption to switch framerate limiter to alternate *"Frametime limit"* mode. New mode allows you to specify the limit directly as a target frametime with 1 microsecond precision
    - Added alternate framerate limiting mode, based on synchronization with display rasterizer position. Now you may synchronize the framerate to up to two independent scanline indices per refresh interval. Combining with power user configurable scanline wait timeout and graphics pipeline flushing options, those settings provide experienced users vendor agnostic ultra low input lag adaptive VSync, half VSync or double VSync functionality on any hardware
    - Added power user oriented idle framerate limiting mode. Unlike traditional framerate limiting mode, idle framerate limiting mode is only affecting inactive 3D applications running in background. Idle framerate limit is specified as a target frametime with one microsecond precision. Idle framerate limiting mode helps to reduce power consumption when you minimize some heavy 3D applications and switch to other processes
- Various On-Screen Display optimizations and improvements:
    - Added adjustable minimum refresh period for On-Screen Display renderer. The period is set to 10 milliseconds by default, so now the On-Screen Display is not allowed to be refreshed more frequently than 100 times per second. Such implementation allows keeping smooth animation when On-Screen Display contents are being updated on each frame (e.g. when displaying realtime frametime graph) without wasting too much CPU time on it
    - Added alternate GPU copy based Vector2D On-Screen Display rendering mode implementation for Direct3D1x applications. New mode provides up to 5x Vector2D performance improvement on NVIDIA graphics cards, however it is disabled on AMD hardware due to slow implementation of *CopySubresourceRegion* in AMD display drivers
    - Vector2D rendering mode is now forcibly disabled in Vulkan applications on AMD graphics cards due to insanely slow implementation of *vkCmdClearAttachments* in AMD display drivers
    - Revamped geometry batching and vertex buffer usage strategy in pure Direct3D12 On-Screen Display renderer (currently used in Halo Wars 2 only)
    - Added Vector2D rendering mode support to pure Direct3D12 On-Screen Display renderer
    - Optimized On-Screen Display hypertext parsing and tessellation implementation
    - Optimized state changes in OpenGL On-Screen Display rendering implementation
    - Improved implementation of On-Screen Display rendering from separate OpenGL context (profile compatibility switch used in certain OpenGL applications, e.g. Pyre) on AMD graphics cards
    - Optimized state changes in Direct3D1x On-Screen Display rendering implementation
    - Solid rectangles and line primitives in Direct3D8 and Direct3D9 On-Screen Display rendering implementations are now rendered from vertex buffer instead of user memory
    - Improved OpenGL framebuffer dimensions detection when framebuffer coordinate space is selected
    - Increased static vertex buffer size for Vulkan and pure Direct3D12 renders to increase amount of primitives rendered in On-Screen Display in a single pass
- Improved desktop duplication based desktop video capture implementation (Windows 8 and newer OS versions):
    - Now desktop video recording sessions do not stop on display mode switch or on switch to exclusive fullscreen mode. Such approach allows you to start capturing video on desktop then launch some 3D application and create a video file containing both desktop and 3D application's video streams
    - Improved video capture API allows video capture frontend applications (e.g. MSI Afterburner) to force desktop or 3D application video capture modes in addition to default mixed desktop/3D application capture mode
    - Now desktop capture is using multhithreaded active busy-wait loop frame capture instead of timer driven frame capture in order to improve frame timing precision and resulting video smoothness. The previous timer driven frame capture can be enabled via configuration file if necessary
    - Decreased desktop duplication timeouts in order to improve RivaTuner Statistics Server GUI response time under certain conditions during desktop videocapture sessions in timer driven frame capture mode
- Improved SDK:
    - Improved RTSSFrametimePipeSample sample. Now the sample demonstrates frametime pipe connection for applications running with both full administrative and limited user rights
    - Improved NVENC plugin. Added NVIDIA 416.xx drivers family support. The plugin was recompiled with newer NVENC encoder API headers, because NVIDIA stopped supporting legacy v4 NVENC API in release 416 and newer series drivers. Due to this change NVENC plugin no longer supports pre-release 358 NVIDIA drivers
- Fixed On-Screen Display rendering in wrong colors when Vector2D mode is selected and Direct3D1x applications use 10-bit framebuffer
- Fixed Vulkan fence synchronization issue, which could cause GPU-limited Vulkan applications to hang due to attempt to reuse busy command buffer

- Active busy-wait loop in the framerate limiter module is now forcibly interrupted during unloading the hooks library to minimize the risk of deadlocking 3D application when dynamically closing RivaTuner Statistics Server during 3D application runtime
- Improved CBT hooks uninstallation routine to minimize the risk of deadlocking 3D application when dynamically closing RivaTuner Statistics Server during 3D application runtime
- Improved validation in OpenGL On-Screen Display rendering routine to minimize the risk of crashing OpenGL applications
- Changed OpenGL cleanup routines to improve compatibility with OpenGL applications using multiple rendering contexts (e.g. GPU Caps Viewer)
- Improved synchronization in 32-bit API hook uninstallation routines
- Added timeout to API hooks injection in CBT hook handler. The timeout is aimed to reduce injection related CPU overhead on some systems, related to high mouse polling rate combined with keyboard/mouse hooks installed by third party applications
- Interoperability D3D10 page flips on some systems are now filtered by framerate calculation module in OpenGL/Vulkan applications
- Hook engine is now using alternate double jump x64 hook trampoline to improve compatibility with third party 64-bit On-Screen Display applications
- Added compatibility profile switch for hooking IDXGISwapChain::ResizeBuffers via VTable instead of hotpatching
- Added hook epilogs support to 32-bit VTable hook handler
- Fixed instance checking implementation in 64-bit RTSSHooksLoader and EncoderServer helper applications
- Added exclusion profile for Forza Horizon 4. Please take a note that On-Screen Display is currently not supported in this game due to its protective system limitations
- Forcible graphics and compute queues synchronization is now disabled by default for Vulkan applications presenting frames from compute queue (AMD Vulkan rendering codepaths in DOOM and Wolfenstein II : The New Colossus). Due to this change, On-Screen Display will be invisible in those games on such platforms by default. Experienced users, understanding and accepting that On-Screen Display rendering will cause performance penalty, may reenable it with PresentFromCompute profile switch
- Added tri-state skinned buttons support in the skin engine
- Updated profiles list

Version 7.1.0 (published on 24.04.2018)

- Added On-Screen Display locking mechanism for third party On-Screen Display client applications. The mechanism is aimed to eliminate unwanted On-Screen Display flickering effect when some client application is performing risky two-stage On-Screen Display refresh while On-Screen Display is being actively refreshed by another client. Please take a note that third party On-Screen Display clients must be also updated in order to use this mechanism and get rid of flickering
- Added new text formatting tags support for displaying minimum, average, maximum, 1% low and 0.1% low framerates in benchmark mode. The tags are allowing On-Screen Display clients to display independent benchmark statistics simultaneously for multiple running 3D applications instead of displaying foreground 3D application statistics only
- Now it is possible to adjust frametime history graph size via RivaTuner Statistics Server properties. Please take a note that you may use positive values to specify the size in pixels or negative values to specify it in symbols
- Now it is possible to toggle benchmark mode state from RivaTuner Statistics Server properties. However, third party benchmark client applications like MSI Afterburner are still required to toggle benchmark mode state with hotkeys during 3D application runtime and to save benchmark results to a text file
- Now it is possible to toggle between averaged or instantaneous calculation modes for peak (i.e. minimum and maximum) framerates for benchmark mode via RivaTuner Statistics Server properties
- Fixed issue in context help system, which could cause the tooltip to flicker when it was displayed below mouse cursor
- Fixed screen capture feature in On-Screen Display preview window when RivaTuner Statistics Server is installed in UAC-protected folder
- Updated SDK:
  - Now RTSSSharedMemorySample sample code is demonstrating the implementation of On-Screen Display locking and flickering filter
- Updated profiles list

Version 7.0.2 (published on 14.12.2017)

- New unified geometry batching implementation for all supported 3D APIs provides more efficient On-Screen Display rendering and more easily extendable renderer architecture
- Due to new unified geometry batching implementation barchart graphs are now being rendered in a single pass with On-Screen Display text
- Added new formatting tags for displaying embedded bars
- Fixed multithreaded audio encoder uninitialization issue, which could cause temporary prerecord files not to be deleted when prerecording video to a file
- Updated profiles list

Version 7.0.1 (published on 09.12.2017)

- Improved built-in benchmarking engine:
  - Added 1%/0.1% low framerate statistics calculation. Unlike traditional calculation approach, which is assuming post-processing of whole set of collected frametimes at the final stage of benchmarking, RivaTuner Statistics Server introduces new concepts of dynamic 1%/0.1% low framerate statistics calculation. The implementation is open source and demonstrated in RivaTuner Statistics Server SDK.
  - 1%/0.1% low framerates are stored in the benchmark results file. 1%/0.1% low framerate are also displayed in benchmark On-Screen Display, so you may track 1%/0.1% low framerate dynamics in realtime.
- Updated SDK:
  - Added new open source RTSSFrametimePipeSample sample. The sample is demonstrating implementation of custom benchmarking software, which is receiving per-frame frametime statistics in realtime via named pipe connection to RivaTuner Statistics Server. Post-processing text frametime log files after running the benchmark is no longer necessary! Now you may create your own benchmarking software and receive frametime data from RivaTuner Statistics Server in realtime directly while 3D application is running. Such approach was in use by some major hardware reviewers for a few years, now it is open source and available to everyone! The sample is also demonstrating the following techniques:
    - Realtime frametime graph rendering
    - Realtime current framerate averaging with sliding window algorithm, minimum and maximum framerates calculation and display
    - Dynamic 1%/0.1% low framerate statistics calculation and display
    - Logging per-frame frametime, current, minimum, average, maximum, 1% low and 0.1% low framerates to CSV text file
- Improved text formatting tags support:
  - Now text position tags can use negative coordinates to move a text to a new position specified in character units instead of zoomed pixel units
  - Improved text position tags handling for cases when the tag is used explicitly before the embedded object
  - Added new visualization style support for embedded graph objects. Now it is possible to embed barchart styled graphs
- Slightly optimized On-Screen Display geometry batching for all 3D APIs in order to simplify the process of new functionality implementation in future versions

Version 7.0.0 (published on 01.11.2017)

- Added initial Microsoft UWP applications support. Now RivaTuner Statistics Server provides full On-Screen Display, framerate/frametime statistics collection and framerate limiting support for sandboxed Windows 10 store applications. Screencapture and videocapture are currently not supported in UWP environment
- Added new framerate statistics calculation and visualization options for those who use RivaTuner Statistics Server as standalone framerate monitoring solution:
  - Added new *"Framerate averaging interval"* option to *"General"* tab. You may use this option to adjust framerate calculation and to see instantaneous framerate instead of smoothed if necessary
  - Added new *"Refresh period"* option to *"General"* tab. You may use this option to adjust the refresh period for framerate and frametime counters and update them on each frame if necessary
  - Added new *"Display integer framerate"* option to *"General"* tab. You may use this option to switch between integer and floating point framerate representation format
  - Added new *"Display maximum frametime"* option to *"General"* tab. You may use this option to switch between current instantaneous frametime value or maximum frametime value during each refresh period. Maximum frametime value can be used for framerate stutters detection
  - Added new *"Enable frametime history overlay"* option to *"General"* tab. You may enable this option to display the graph or diagram, representing frametime history for up to 1024 the last rendered frames. The history graph can be used for framerate stutters detection
- New default Metro interface styled skin designed by 00pontiac
- Default skin composition mode is now set to layered mode with color key and light transparency
- Improved default On-Screen Display rendering configuration provides better default On-Screen Display appearance on modern mainstream PC display resolution. Now raster 3D rendering mode with Unispace font and shadow rendering are enabled
- Improved font size adjustment for raster 3D On-Screen Display rendering mode. Base font size (i.e. font size for 1x zooming ratio) is no longer hardcoded inside the profile, now you can adjust it directly in font selection window. After selecting the base font size, you may still apply fixed 1x-8x zooming ratio to it on the fly using On-Screen Display zoom adjustment slider

- Hook engine is now using alternate shorter x64 hook trampoline to provide compatibility with Windows 10 Creators Update Direct3D9 runtimes. The previous trampoline can be enabled via the configuration file if necessary
- Hook libraries, video encoding libraries and hardware encoding plugins are now digitally signed to simplify and speed up the process of new versions' whitelisting by third party anticheat systems. Thanks to EAC and BattleEye for making it possible!
- Fixed bug in Vulkan On-Screen Display cleanup code causing The Talos Principle to crash to desktop on changing video settings
- Fixed bug in Vulkan On-Screen Display cleanup code causing On-Screen Display text to look unexpectedly pixilated after starting Vulkan demos from Geeks 3D GPU Caps Viewer multiple times
- New reusable Vulkan vertex buffer implementation no longer limits the On-Screen Display geometry by 16K vertices
- Fixed bug is skin engine causing part of the profiles window to be rendered in solid black color when using layered skin composition mode with alpha channel
- Fixed crash on startup under Windows Vista with no platform update installed
- Fixed vertex buffer overflow in Direct3D8 and Direct3D9 raster 3D On-Screen Display rendering implementations causing the text to disappear when trying to display around 700 or more characters
- Improved On-Screen Display rendering quality in Direct3D1x applications using sRGB framebuffers
- Improved compatibility with Direct3D12 applications, which dynamically switch between single-GPU and multi-GPU AFR swapchains (e.g. Gears of War 4 with multi-GPU patch)
- On-Screen Display is no longer being rendered improperly when switching between vector and raster On-Screen Display rendering modes during Direct3D1x application runtime
- FCAT overlay support is no longer limited to vector On-Screen Display rendering modes in Direct3D1x applications, now it is also supported for Direct3D1x applications when raster 3D On-Screen Display rendering mode is enabled
- Fixed incorrect FCAT overlay rendering in OpenGL applications when using 2-bars / 3-bars / 4-bars FCAT overlay modes
- On-Screen Display shadow in now rendered in proper RGB colors instead of BGR in On-Screen Display preview window
- New On-Screen Display shadow rendering for raster 3D mode provides improved shadow rendering quality for some combinations of fonts and low On-Screen Display zooming ratios. The previous mode can be also enabled via profile if necessary
- Now *"Delete profile"* button is unlocked when *"Hide pre-created profiles"* option is enabled and template based profile is selected
- Now you may hold <*Ctrl*> when pressing *"Add profile"* button to create a profile for currently active 3D applications, or hold <*Shift*> to create a profile with disabled application detection (i.e. exclusion) for currently active 3D applications
- Optimized Direct3D state changes in situations where multiple batches of primitives are rendered in On-Screen Display (e.g. when rendering the text and FCAT overlay simultaneously)
- Added On-Screen Display transparency support. New On-Screen Display color picker window allows you to adjust color opacity level to make On-Screen Display to appear semi-transparent if necessary. Please take a note that transparency if fully supported for raster 3D On-Screen Display rendering mode. Some symbols of polygonal vector 3D font can be rendered with artifacts when transparency is selected and vector 2D mode doesn't support transparency at all
- Added On-Screen Display background color fill mode. New *"On-Screen Display fill"* option allows you fill underlying On-Screen Display area with solid or transparent color. Fill color and transparency level can be customized in *"On-Screen Display palette"* colors list
- Updated SDK:
    - Improved shared memory layout. Now each running 3D application entry contains x64/UWP application architecture usage flags in addition to previously available 3D API usage flags
    - Added text formatting tags support for On-Screen Display text. Now the client applications using RivaTuner Statistics Server to render a text in On-Screen Display (e.g. AIDA, HwInfo or MSI Afterburner itself) can apply custom text colors, alignment, size and position to some parts of displayed text. Format tags are fully supported for raster 3D On-Screen Display rendering mode only, some tags are ignored for vector On-Screen display rendering modes
    - Added embedded objects support. Now the client applications using RivaTuner Statistics Server to render a text in On-Screen Display (e.g. AIDA, HwInfo or MSI Afterburner itself) can embed graph objects (e.g. GPU usage history graph) directly into the text rendered in On-Screen Display
    - Improved On-Screen Display text macro translator. To provide more flexibility and customization %FRAMERATE% macro is now deprecated and replaced with new tags allowing printing application specific framerate, frametime and 3D API independently
    - Updated RTSSSharedMemorySample sample code:
        - Now the sample demonstrates text format tags usage and shows you how to apply custom On-Screen Display color, subscript style and custom alignment to a text rendered in On-Screen Display from a client application
        - Now the sample demonstrates new embedded objects usage and shows you how to render a graph from a client application
        - Now the sample also demonstrates framerate and frametime display via both %FRAMERATE% macro and new tags
        - The sample manifest has been changed to require administrative rights, that is necessary to modify the profiles via profile management API
- Improved On-Screen Display preview window:
    - Now you may detach and set custom position and size of preview window by double clicking preview window area. Increased size of preview window can be useful if you prefer to adjust On-Screen Display position visually via dragging it inside the preview
    - Now you may right click preview window area and select *"Capture screenshot"* command from the context menu to automatically capture screenshot from currently running 3D application and use it as On-Screen Display preview

background, or use *"Load"* command to open and load some previously captured screenshot. This feature can be useful if you prefer to adjust On-Screen Display position visually via dragging it inside the preview. Original On-Screen Display preview window background can be restored with *"Clear"* command from the context menu

- Improved dynamic Direct3D12 presentation command queue detection algorithm
- Added On-Screen Display support for Direct3D11 applications using alternate frames presentation technique (e.g. Halo Wars Definitive Edition)
- Added power user oriented profile compatibility setting, allowing enabling alternate VTable based hooking for Direct3D1x applications concurrently presenting frames from multiple threads (e. g. Forza Horizon 3 or Halo Wars 2)
- Added power user oriented profile compatibility setting, allowing enabling pure Direct3D12 On-Screen Display renderer for some Direct3D12 applications conflicting with D3D11on12 interop layer (e.g. Halo Wars 2)
- Added power user oriented profile compatibility setting for OpenGL applications rendering from forward compatible 3.x+ context with disabled deprecated immediate OpenGL mode functionality. The switch allows the server to fall back to rendering path using a separate legacy OpenGL context
- Added OpenGL context validation algorithm to improve compatibility of raster 3D On-Screen Display rendering mode with OpenGL applications, which dynamically recreate the context (e.g. on display mode switching)
- Added On-Screen Display support for Direct3D 11 applications using feature level 12.1 (e.g. Middle Earth : Shadow of War)
- Added On-Screen Display profile for Wolfenstein : The New Order
- Added On-Screen Display profile for Quake 2
- Added On-Screen Display profile for Pyre
- Added On-Screen Display profile for Euro Truck Simulator 2
- Added On-Screen Display profile for Citra Nintendo 3DS emulator
- Added On-Screen Display profile for CEMU Wii U emulator
- Added On-Screen Display profile for theHunter : Call of the Wild
- Revamped internal benchmarking engine
- Maximum framerate limit value has been increased to 300 FPS to provide more framerate limiting possibilities to 240 Hz monitor owners
- Now the installer is preserving installation path
- Updated profiles list

Version 6.6.0 (published on 14.02.2017)

- Added Vulkan On-Screen Display, screen capture and video capture support
- Fixed Direct3D12 function offsets cache validation algorithm in hooking system
- Improved compatibility with Direct3D9 applications, which dynamically recreate Direct3D devices
- Added new *"Injection properties"* compatibility settings section to *"General"* tab:
  - Added injection delay adjustment settings for delayed hooking engine. New settings are aimed to provide an alternate and easy to use way of disabling delayed injection for applications using 64-bit Steam and Origin overlays
  - Added *"Reset cache"* button. The button can be used for troubleshooting to invalidate function offsets cache without re-installing the application
- Updated SDK:
  - Added new profile management API. Profile management API allows third party applications to read, create or modify application-specific profiles and create own built-in GUI for RivaTuner Statistics Server profile management if necessary.
  - Updated RTSSSharedMemorySample sample code. Now the sample demonstrates new profile management API usage and shows how to adjust On-Screen Display position and color in global RivaTuner Statistics Server profile from a third party application
- Updated profiles list

Version 6.5.1 (published on 20.11.2016)

- Added power user oriented profile compatibility setting for DX1x applications concurrently accessing the swap chain from multiple rendering threads (e.g. DX12 rendering mode in Battlefield 1 Fall Update)
- Added compatibility profile for Battlefield 1 Fall Update to address application hang/crash on startup issues in DX12 mode with On-Screen Display enabled
- Added power user oriented profile setting, allowing you to customize framerate denominator for the built-in framerate limiter. The denominator can be customized to adjust the limit in fractional steps (e.g. denominator 10 to adjust the limit in 1/10 FPS steps)
- Updated localizations

Version 6.5.0 (published on 27.10.2016)

- Direct3D12 On-Screen Display, screen capture and video capture support. Currently RivaTuner Statistics Server is the only application providing On-Screen Display support for both single-GPU Direct3D12 applications and multi-GPU Direct3D12 applications running in explicit LDA mode on both AMD Crossfire and NVIDIA SLI systems. Please take a note that Direct3D12 support is provided for native Win32 applications only, Microsoft UWP applications are currently not supported
- Added *"Hide pre-created profiles"* option to *"General properties"* section in *"General"* tab. New option allows you to see your own profiles only and hide built-in pre-created 3D exclusion profiles supplied with the application
- 64-bit Steam overlay library has been removed from delayed hooking engine configuration. Now On-Screen Display is being rendered immediately after starting 64-bit Steam applications without 15 second delay. Power users may still add 64-bit Steam overlay library to delayed hooking engine configuration if necessary
- Fixed bug in the encoder server causing it to continue capturing audio and leak memory after recording a video through dedicated encoder server and closing 3D application without stopping video recording session
- Updated profiles list

Version 6.4.1 (published on 02.12.2015)

- Added On-Screen Display profile for Assassin's Creed Syndicate
- Added On-Screen Display profile for Just Cause 3
- Starting desktop recording sessions with *"Use dedicated encoder server"* option enabled no longer results in blocking subsequent in-game recording sessions until restarting the application
- Fixed inverted state of *"Low-level encoding"* option in NVIDIA NVENC H.264 encoder plugin
- Improved AMD AMF VCE H.264 encoder plugin:
    - Unlocked encoding in > 1080p resolutions for VCE 3.0 capable graphics cards (AMD Tonga and Fiji GPUs)
    - Added option for selecting host, Direct3D9 or Direct3D11 memory type for encoder. It is recommended to leave default Direct3D9 memory type for everyday usage and select the rest types for performance testing or troubleshooting scenarios only
- Added Brazilian Portuguese localization
- Updated profiles list

Version 6.4.0 (published on 30.10.2015)

- Fixed On-Screen Display rendering in Direct3D9Ex applications compiled with Windows 10 compatibility flags (e.g. recent update of World of Tanks)
- Fixed On-Screen Display rendering in 64-bit Direct3D9 applications using additional swapchains (e.g. DOTA 2 Reborn)
- Fixed crashing in Direct3D10 and Direct3D11 applications when FCAT indicator is enabled in conjunction with vector 2D On-Screen Display rendering mode
- Improved thread safety of delayed injection implementation
- Added On-Screen Display profiles for Batman : Arkham Knight and Daylight to provide correct On-Screen Display rendering on NVIDIA graphics cards when advanced NVIDIA GameWorks visual effects are enabled
- Added On-Screen Display profile for 64-bit version of StarCraft II
- Improved NVIDIA NVENC H.264 encoder plugin:
    - Increased upper adjustable bitrate limit to 130 Mbps for NVIDIA Maxwell graphics processors family
    - Added new option allowing you to disable low-latency encoding if necessary. Low-latency encoding mode is aimed for streaming and it configures NVENC hardware/software infrastructure to compress frames as fast as it is possible. So the driver may re-balance GPU load in graphics intensive applications in order to achieve as low compression latency as it is possible. You may disable low-latency encoding to prevent the driver from balancing GPU load and allow it to sacrifice compression latency under high GPU load
- Added AMD AMF VCE H.264 encoder plugin:
    - Considering that AMD officially dropped OpenEncode hardware acceleration API support since Catalyst 15.7, new plugin uses alternate AMD AMF hardware acceleration API. New plugin it is aimed to replace the previous AMD VCE H.264

plugin, based on deprecated AMD OpenEncode API. However, the previous OpenEncode API based plugin is still included in distributive and can be used for performance testing or solving compatibility issues on pre-Catalyst 15.7 drivers. Please take a note that AMD AMF libraries require installing Microsoft Visual C++ 2013 runtime libraries, those libraries are not included in distributive and must be downloaded and installed separately

- Source code of all hardware accelerated H.264 encoder plugins is now included in RivaTuner Statictics Server SDK. Now you may create your own hardware accelerated plugins or modify existing plugins if you wish to create a custom build with deeper customization of hardware encoder
- Improved encoder plugin loader
- Updated profiles list

Version 6.3.0 (published on 09.11.2014)

- Improved support for 64-bit Direct3D9 applications using multiple swapchains (Banished, Outlast, The Vanishing of Ethan Carter, Styx : Master of Shadows)
- Added delayed injection technology support to hooking engine. Delayed injection is aimed to bypass On-Screen Display exclusivity check implemented in some other On-Screen Display applications (e.g. 64-bit Steam Overlay) and prevent such applications from blocking their On-Screen Display functionality. Delayed injection engine is user configurable so power user may extend the list of DLL modules triggering delayed injection mechanism and adjust the delay. Currently the delayed injection is triggered by 64-bit Steam overlay and 64-bit Origin In-Game overlay libraries and injection delay is set to 15 seconds if those libraries are loaded in the application address space
- Improved skin engine:
  - Improved skin rendering performance for layered skin composition mode with alpha channel
  - Added backbuffering support to skinned windows to simplify implementation of various framebuffer processing related effects (e.g. skin scaling)
  - Added skins scaling support. The scaling may improve the appearance of some small third party skins on high resolution monitors. Please take a note that upscaling may reduce the performance a bit
- Added On-Screen Display profile for Planetary Annihilation
- Added On-Screen Display profile for Call of Duty : Advanced warfare
- Updated profiles list

Version 6.2.0 (published on 04.09.2014)

- Improved compatibility with FRAPS in 64-bit mode
- Removed MSI Afterburner exclusivity lock for 64-bit On-Screen Display service
- Drastically improved skin engine:
  - Improved skin compiler gives more detailed error messages when skin compilation fails due to error in some source image file
  - Source image file format is no longer limited to 24-bit BMP files only. Now skin compiler supports all possible bit depths for BMP format and fully supports PNG format with alpha channel
  - Added built-in bitmap effect for extracting alpha-channel from PNG image files
  - Skin format has been upgraded to v1.3. New format supports alpha channel based transparency for skinned window, allowing skin designers to define semi-transparent skin areas, apply antialiasing to the skin window edges and so on
  - Added new skinned window composition modes support and *"Skin composition mode"* settings to *"User interface"* tab. New settings allows you to use one of the following modes:
    - Traditional mode – suits best for backward compatibility with existing skins and performance testing
    - Layered mode with colorkey - provides much faster rendering of skins with non-rectangular window shape and additionally allows you to adjust transparency of skinned window
    - Layered mode with alpha – provides per-pixel alpha channel support and advanced visual effects for compatible skins and also allows you to adjust transparency of skinned window
- Skin format reference guide has been updated to v1.7 to document these changes

Version 6.1.3 (published on 15.08.2014)

- Improved handshaking algorithm reduces the risk of seeing multiple running instances of child processes (e.g. RTSSHooksLoader64)

- Added 340.xx drivers family support to NVIDIA NVENC H.264 encoder plugin
- Added exception handling to audio capture routine
- Added automatic prerecording support to video capture engine. Now video capture client applications (e.g. MSI Afterburner) can configure the server to start prerecording automatically when 3D application is started

Version 6.1.2 (published on 10.06.2014)

- Added raster 3D On-Screen Display rendering mode support for Wolfenstein : The New Order rendering engine

Version 6.1.1 (published on 21.05.2014)

- Fixed memory leak in Intel QuickSync H.264 encoder plugin running in Direct3D 11 mode
- Added 337.xx drivers family support to NVIDIA NVENC H.264 encoder plugin
- Added On-Screen Display profile for Banished Direct3D9 renderer
- Added On-Screen Display profile for Daylight

Version 6.1.0 (published on 27.03.2014)

- Improved encoding plugins architecture:
    - Encoder benchmark results are no longer affected by encoder initialization time
    - Extended encoder benchmark settings. Now you can run the benchmark in current display settings like before or select custom display mode to estimate encoding performance in lower resolution. This feature is useful for comparing peak encoding performance of different hardware H.264 encoders (e.g. Intel QuickSync vs AMD VCE)
    - Now the plugins are allowed to start encoding frames without specifying the header prior to encoding. MKV container implementation has been also improved to provide support for delayed header reporting. This feature is required to support H.264 encoding APIs with no native SPS/PPS headers reporting support, e.g. AMD OpenVideo API
    - Now plugin engine can provide input frames to the plugins in native NV12 format so the plugins do not need to perform internal RGB to NV12 color space conversion
- Improved dedicated encoder server service:
    - Seriously optimized data exchange path between video capture engine and dedicated encoder. Data exchange related performance overhead is seriously reduced and close to invisible now
- Improved Intel QuickSync H.264 encoder plugin:
    - Added wait loop to encoding routine to prevent frame dropping when VPP input queue is full. Such frame dropping caused by queue overflow could cause the benchmark to be stopped with MFX_ERR_MEMORY_ALLOC error
    - Increased upper adjustable bitrate limit to 50 Mbps
    - Output bitstream is now dynamically resizable to allow encoding video at 40 Mbps and higher bitrates
- Added AMD VCE H.264 encoder plugin:
    - New plugin uses AMD Accelerated Parallel Processing SDK and AMD OpenVideo API to access fixed-function H.264 encoder unit built into AMD RADEON HD 7xxx and newer AMD GPU families. The plugin is able to encode H.264 video in up to 1080p resolution at up to 60 FPS framerate. Please take a note that Catalyst 14.1 or newer AMD display drivers are recommended for proper plugin functionality
    - The pluging GUI provides you adjustable encoding bitrate and AMD’s recommended speed, balanced and quality presets. However, each preset is represented by independent CFG file and can be additionally tweaked by experienced users if necessary
- Added NVIDIA NVENC H.264 encoder plugin:
    - New plugin uses NVIDIA NVENC SDK to access fixed-function H.264 encoder unit built into NVIDIA Kepler and newer NVIDIA GPU families. Please take a note that release 334.67 or newer NVIDIA display drivers are required for proper plugin functionality
- Various protections and improvements for low memory conditions:
    - Added exception handling for memory allocation routines
    - Now encoder input queue size is limited to 256MB in 32-bit applications
    - Improved encoder input queue memory management and reduced memory footprint for VFW and external plugin compression modes

- Now video capture client applications (e.g. MSI Afterburner) can use raw NV12 video compression format. This format can be used to replace uncompressed video recording with fixed 2x compression ratio at the cost of some color information loss caused by RGB to NV12 color space conversion
- Framerate calculation fix for Outcast, introduced in the previous version, is no longer hardcoded. Now it is being applied at application profile level to selected executables only to avoid breaking support for other applications handling multiple swap chains differently (e.g. Dragon Age : Origins)
- Improved compatibility with some third party VFW codecs (e.g. PICVideo M-JPEG codec)
- Added raster 3D On-Screen Display rendering mode support for Rayman Legends and ARMA II rendering engines
- Added vector 3D and raster 3D On-Screen Display rendering modes support for DOOM 3 : BFG edition rendering engine
- Framerate, frametime and per-frame videocapture statistics (frame compression ratio and frame compression time) are no longer being refreshed in On-Screen Display on each frame. Now refresh interval is set to 500 ms and can be adjusted by power users via the profiles if necessary
- Updated profiles list

Version 6.0.0 (published on 22.12.2013)

- Added 64-bit applications support service. The service is optional and can be disabled in compatibility properties of RivaTuner Statistics Server if necessary. Please take a note that 64-bit applications support service development is exclusively sponsored by MSI, so due to licensing 64-bit On-Screen Display functionality is exclusively available in MSI Afterburner client application only. Please refer to *"Known issues"* section to see the list of additional 64-bit functionality related limitations
- Improved dedicated encoder server service:
  - Now the service is optional and can be disabled in compatibility properties of RivaTuner Statistics Server if necessary
  - Both 32-bit and 64-bit versions of the encoder server service are available and you can select preferred one in compatibility properties of RivaTuner Statistics Server
  - The service is no longer specific to external encoding plugins. Now dedicated encoding server can be used in any encoding scenario including but not limited to built-in encoding modes (uncompressed, MJPG and RTV1) or VFW encoding mode
  - Dedicated encoder server usage seriously extends the possibilities of prerecording to RAM due to performing encoding in separate address space. Now you can use up to 2GB of RAM for prerecord buffer when using 32-bit encoder server and >2GB of RAM for prerecord buffer when using 64-bit encoder server
- Improved Intel QuickSync H.264 encoder plugin:
  - Dedicated encoder service usage option has been removed from the plugin GUI, now it is available as independent option directly in video capture client application GUI
- Added raster 3D On-Screen Display rendering mode support for DirectX 11.2 applications
- Fixed issue causing wrong legacy index to be written to AVI files when recording multiple videos during single prerecording session
- Fixed framerate calculation for Direct3D9 applications using multiple swap chains (e.g. Outlast)
- Added On-Screen Display support for Direct3D11 applications using mixed Direct3D10/Direct3D11 rendering (e.g. Call of Duty : Ghosts with NVIDIA APEX libraries)
- Updated PNG compression library
- Added French localization
- Updated profiles list

Version 5.5.0 (published on 29.10.2013)

- Improved Direct3D9 video capture performance on NVIDIA Kepler series graphics cards under release 304.xx and newer NVIDIA drivers due to using hardware accelerated Inband Frame Readback (NvIFR) technology. NvIFR is low-latency GPU/DMA accelerated render target readback interface, which provides video capture applications an ability to capture frames from Direct3D pipeline with much lower performance penalty than any traditional Direct3D based render target readback implementations

Version 5.4.1 (published on 28.10.2013)

- Improved prerecord buffer saving algorithm. Realtime prerecord buffer saving speed no longer depends on video keyframes generation rate, now prerecord buffer is being saved to disk at fixed 30x recording speed while recording is in progress
- Fixed issue causing the second and subsequent captured videos to be corrupted when recording multiple videos during single prerecording session under certain conditions
- Improved function offsets cache validation algorithm in hooking system
- Now RivaTuner Statitics Server displays warning and recommends you to restart application when function offset cache initialization fail due to some reason (e.g. on the very first application start in antivirus “sandbox”)

Version 5.4.0 (published on 25.10.2013)

- Added prerecording support to video capture engine. Two different prerecording modes are available to the client applications:
    - Prerecord to a temporary file. This mode suits best for prerecording long videos (e.g. 10 minutes or more of H.264 video) or for prerecording videos with low compression ratio (e.g. lossless Lagarith video)
    - Prerecord to RAM. This mode suits best for prerecording short videos (e.g. a couple minutes of H.264 video) and it provides a bit better performance than prerecord to file. Also, unlike prerecord to file, prerecord to RAM minimizes writes so it can increase the lifetime of SSD if you plan to use the prerecording frequently
- Improved muxer architecture with better extendibility makes it much easier to add new video containers support in future versions
- Fixed incorrect minutes being shown in video capture timer in On-Screen Display for videos longer than 1 hour
- Updated profiles list

Version 5.3.2 (published on 15.09.2013)

- Improved encoding plugins architecture:
    - Now the plugins can be optionally loaded either in host 3D application address space or in dedicated encoder server application address space. Dedicated encoder server usage reduces the peak encoding performance a bit, however it may give you the following advantages in conjunction with Intel QuickSync H.264 encoder plugin:
        - Minimization of different Direct3D synchronization/multithreading related issues. Host application 3D engine NVIDIA/AMD Direct3D device and encoder Intel Direct3D device are now isolated from each other and run in different address spaces and have much less chances to affect each other. Multithreaded optimizations no longer being forcibly disabled when capturing videos in Direct3D9 applications via Intel QuickSync H.264 plugin working in Direct3D9 mode with dedicated encoder server
        - Now it is possible to use hardware accelerated Intel QuickSync H.264 encoder plugin to capture video from the games using modified Direct3D runtime libraries
        - Now it is possible to use software emulation mode for Intel QuickSync H.264 encoder without installing Intel Media SDK. Intel software emulation libraries are distributed with encoder server
    - Now the plugins can optionally provide extended error reporting
- Improved Intel QuickSync H.264 encoder plugin:
    - Added dedicated encoder server support
    - Added extended error reporting support
    - Recompiled under Intel Media SDK 2013 R2
- Fixed previously available compatibility mode with modified Direct3D runtime libraries broken version due to introducing alternate experimental compatibility mode with modified Direct3D libraries in the previous version
- Fixed incorrect compression ratio value displayed in OSD when capturing video in high resolutions (4 MB and higher framebuffer size)
- Added high performance desktop video capture for Windows 8 and newer OS via DXGI 1.2 desktop duplication API
- Audio track description is no longer being written to MKV file if audio track is empty, e.g. during silent desktop recording session
- Various capture performance related tweaks:
    - Tuned queued screen capture implementation
    - Tuned pipeline flushing implementation
    - Tuned dynamic pipeline flushing control
- Improved frames timestamping accuracy
- On-Screen Display text length limit has been extended from 256 bytes to 4 kilobytes per application slot. Third party developers using RTSS OSD support in their applications should refer to updated sample code in SDK
- Added On-Screen Display profile for Rayman Legends
- Updated profiles list

Version 5.3.1 (published on 22.08.2013)

- Direct3D9 applications no longer freeze for a few seconds after capturing video via Intel QuickSync H.264 plugin working in Direct3D11 mode
- VFW compression or external plugin configuration is now being saved properly from client applications on clean installation of RivaTuner Statisctics Server without the need to open RivaTuner Statistics Server GUI at least once
- Fake empty tracks and cues are no longer being written to MKV file after restarting video capture session multiple times during 3D application runtime
- Improved encoding plugins architecture. Now the plugins can report multithreading safety hint to the encoding engine to force mutitheraded optimizations to be disabled regardless of the client application settings. Intel QuickSync H.264 plugin working in Direct3D9 mode uses this hint when capturing videos in Direct3D9 applications to avoid application hanging problems. Please take a note that it is a temporary solution, there is still a lot of room left for optimization when capturing videos in Direct3D9 applications via Intel QuickSync H.264 plugin working in Direct3D9 mode. In order to avoid performance penalties, it is strongly recommended to configure Intel QuickSync H.264 plugin to use Direct3D11 codepath when possible
- Fixed screen capture routines corrupted in the previous version
- Added alternate experimental compatibility mode with modified Direct3D libraries. New mode is transparent for end users and enabled by default. Please take a note that “Custom Direct3D support” option in RivaTuner Statistics Server GUI disables new mode and forces the previous compatibility codepath

Version 5.3.0 (published on 14.08.2013)

- Added external encoding plugin modules support. Plugin modules architecture provides an easy way to implement different hardware accelerated video encoders in RivaTuner Statistics Server.
- Added benchmark mode for external encoding plugin modules. Benchmark mode allows you to verify if hardware encoder is configured and working properly as well as estimate expected hardware encoding performance.
- Added external QSV.DLL encoding plugin. The plugin provides you high-performance hardware accelerated H.264 encoding on Intel QuickSync Video capable platforms. Intel QuickSync H.264 encoder is able to compress 1080p video at 60 FPS with no major CPU performance hit. Hardware accelerated Intel QuickSync H.264 encoder was introduced special to compete with NVIDIA’s ShadowPlay hardware accelerated H.264 encoder. Free hardware accelerated H.264 video capture and encoding is no longer an exclusive selling point of NVIDIA Kepler GPU family, now the same functionality is available on much wider range of hardware platforms on both AMD and NVIDIA GPU based graphics cards absolutely for free!
- Tweaked encoder queue depth. The queue depth is no longer hardcoded to 32 frames, now it is equal to recording framerate
- Decreased server startup time on OS versions with no any service pack version installed
- Improved function offsets cache validation algorithm in hooking system
- Added proxy Direct3D library protection system. Now RivaTuner Statistics Server forcibly disables application hooking and blocks OSD, screen capture and video recording functionality if application is running via the proxy Direct3D runtime library (various game mods based on hacked D3D DLLs). Compatibility with most of such applications can still be improved by enabling custom Direct3D runtime libraries support in the application profile

Version 5.2.0 (published on 17.06.2013)

- Added Matroska (MKV) container support to video capture engine

Version 5.1.2 (published on 08.05.2013)

- Fixed Push-To-Talk hotkey handler, which was broken since version 5.0.0 after migration to new VC++ compiler

Version 5.1.1 (published on 26.04.2013)

- Improved keyboard navigation implementation for profiles list
- Improved reboot notifications for locked files installation/uninstallation
- Now RivaTuner Statistics Server prevents attempts to start application without rebooting the system in case of using delayed locked files update scenario during installation

- Added screen capture and video capture support for fullscreen Direct3D7 and DirectDraw applications (windowed mode is not supported). Please take a note that DirectDraw and Direct3D7 hooking is disabled by default in global profile, it is recommended to enable DirectDraw and Direct3D7 hooking at application profile level instead of enabling it globally
- Build number is now displayed in application tray icon tooltip
- Localizable installer files for third party translators are now included in distributive in .\SDK\Localization\Installer folder
- Updated Italian localization

Version 5.1.0 (published on 16.04.2013)

- Added update checking system
- Added version checking system to installer. Now the installer is giving you a warning when you're trying to install older or reinstall currently installed version of the server
- Added client notification system to installer. Now the installer is notifying client applications like MSI Afterburner and EVGA Precision in the end of the server installation process to allow the clients to restart themselves
- Added workaround for missing tray icon when starting application while EXPLORER.EXE is not responding
- Application tray icon is now being restored after restarting EXPLORER.EXE
- Fixed issue with newly added profiles not being automatically selected in the profiles list
- Added keyboard navigation support for profiles list
- Updated Italian installer
- Added On-Screen Display profile for Medal of Honor: Allied Assault
- Updated profiles list

Version 5.0.1 (published on 05.04.2013)

- Fixed color sequence pattern for NVIDIA FCAT system under Direct3D10 and Direct3D11
- Added alternate FCAT rendering modes for multi-GPU systems
- Added On-Screen Display profile for DOOM 3: BFG edition
- Updated Italian localization

Version 5.0.0 (published on 01.04.2013)

- The server is no longer being distributed as a part of client applications like MSI Afterburner and EVGA Precision. Now it comes with own installer and can be optionally installed when necessary. The clients will automatically hide any server dependent functionality (such as On-Screen Display, framerate monitoring, automatic 2D/3D profiles management, screen and video capture and so on) when the server is not installed
- Ported to new VC++ compiler, new core and skin engine libraries with better extendibility
- Slightly redesigned skin. Now framerate limiting and custom Direct3D runtime libraries support can be controlled directly from the main window
- Added optional frame color indicator overlay for NVIDIA FCAT system
- Updated RTSSSharedMemorySample sample code. Now the sample demonstrates how to detect RivaTuner Statistics Server installation path and launch the server from a third party client application

Version 4.5.0 (published on 23.01.2013 with MSI Afterburner v2.3.1)

- Improved audio mixer architecture provides better extendibility and better audio tracks synchronization when mixing multiple audio tracks in one
- Push-To-Talk audio recording support. Now you can assign a hotkey for each audio stream and record unmuted audio only when the hotkey is pressed
- Updated German localization

Version 4.4.0 (published on 19.11.2012 with MSI Afterburner v2.3.0)

- Improved RTV1 compression algorithm. New compression approach combines DXTC, RLE and uncompressed data saving algorithms and allows you to increase image quality at the cost of decreasing compression ratio, while keeping extremely fast compression performance. Please take a note that the previous RTV1 compression implementation is equal to the lowest possible 25% image quality/compression ratio balance in new implementation
- Added 64-bit version of RivaTuner Video Codec. Now RTV1 videos are playable and editable in 64-bit applications (e.g. 64-bit Sony Vegas PRO or Adobe Premiere PRO CS6)
- Huffman tables are no longer omitted by MJPG encoder to improve compatibility with applications using limited MJPG decoders (e.g. Sony Vegas PRO)
- Now both 32-bit and 64-bit versions of RivaTuner Video Codec can optionally decode MJPG videos. Third party MJPG codec installation is no longer required to play or edit MJPG videos with omitted Huffman tables in some applications with limited MJPG decoding support (e.g. Sony Vegas PRO). MJPG decoder can be enabled in video capture compatibility properties
- Multisource audio recording. Now it is possible to record audio from two independent audio sources (e.g. game audio and microphone sound) in two independent audio tracks then mix them later in video editing application (e.g. Sony Vegas PRO) or optionally mix multiple tracks in realtime during capture
- Now it is possible to downmix multichannel audio to stereo in stereo rip and stereo mix modes. Previously available stereo rip mode simply rips front left and right audio channels from multichannel audio stream, whilst new stereo mix mode merges audio from front left, right, center and surround channels
- WASAPI audio packets with timestamp error are no longer included into output audio stream
- External VFW codecs support. Now in addition to built-in uncompressed, RTV1 and MJPG encoders it is also possible to encode video using external VFW codecs installed in the system. It is recommended to download, install and use Lagarith Lossless Codec for lossless video capturing or x264vfw codec for the maximum compression ratio, external codecs support architecture was developed to provide the best compatibility with these codecs
- Video stream synchronization is now based on inserting NULL frames instead of indexing existing frames
- Frame dropping implementation has been improved to provide compatibility with external VFW codecs using temporal compression and deltaframes
- Now output video dimensions are always cropped to be multiples of 2 to provide compatibility with external VFW codecs
- Now framerate is calculated on each frame with sliding 1000ms window approach instead of calculating it once per second
- Improved rendering pipeline flushing implementation for Direct3D9 applications provides more stable framerate when capturing videos with VSync enabled
- New dynamic Direct3D rendering pipeline flushing approach can decrease videocapture related performance hit when capturing videos in Direct3D applications with high rendering framerate
- Added On-Screen Display support for Direct3D10 applications in Windows 8
- Added German localization
- Updated profiles list

Version 4.3.4 (published on 20.06.2012 with MSI Afterburner v2.2.2)

- Fixed framerate calculation for DirectX10/DirectX11 applications, which actively use presentation testing during rendering (e.g. Max Payne 3)

Version 4.3.3 (published on 19.04.2012 with MSI Afterburner v2.2.0)

- Added audio capture support to video capture engine. The engine supports stereo audio capture via DirectSound under Windows XP and both multichannel or downmixed stereo capture via WASAPI under Windows Vista and newer operating systems
- Primary desktop video capture support for Windows XP and newer operating systems. Please take a note that desktop video capture is not supported under Windows Vista and newer OS with Aero Glass interface enabled, so Aero Glass is being automatically disabled by the server while capturing a video from desktop. Please take a note that both desktop video capture support and forcible Aero Glass interface toggling can be disabled via the configuration files if necessary
- Improved bitrate calculation in AVI file headers
- Modified screen/videocapture progress indicator appearance
- Improved video frames timestamping approach may improve smoothness of videos captured on low framerates
- Improved RTV1 codec provides better compression ratio (significantly better under certain conditions) without affecting the image quality and encoding performance

- Added alternate RTV1 compression mode providing 1.5x - 2x better realtime compression performance at the cost of some minor image quality loss. Video quality slider is now unlocked for RTV1 format, 100% quality selects original compression mode whilst any other value selects new high performance compression mode
- Now video capture client applications display captured video file time, size, per-frame compression ratio and time in the On-Screen Display next to the video capture progress indicator when *"Show own statistics"* option is enabled in the server's properties
- Added gamma corrected video capture support
- Added workaround for AMD Direct3D driver issues causing On-Screen Display to be invisible in Direct3D8 and Direct3D9 applications when using Raster 3D On-Screen Display rendering mode
- Added workaround for AMD OpenGL driver PBO issues causing the driver to crash or perform abnormally slow during videocapture
- Added Direct3D9Ex support (DOTA 2 and Darkness 2 demo)
- Improved desktop windows notification mechanism eliminates unwanted beeps occurring on some systems during the server startup / shutdown
- Added video capture compatibility setting allowing cropping output video dimensions to make them multiples of 16 to improve compatibility with some third party MJPG decoders
- Improved tabbed text formatting provides better compatibility with custom On-Screen Display group names
- Dedicated 1/2, 1/3 and 1/4 frame downsampling routines have been replaced with unified arbitrary downsampling routine. Now the server provides arbitrary frame downsampling functionality and allows selecting 360p, 480p, 720p, 900p and 1080p frame sizes for both 16:9 and 16:10 aspect ratios. Please take a note that arbitrary frame sizes list in client applications is user extendable so you may add your own custom frame sizes to the configuration file if necessary
- Now multithreaded video capture optimization is forcibly disabled by default on single core CPUs
- Updated profiles list

Version 4.3.1 (published on 16.01.2012 with EVGA Precision v2.1.2)

- Added new profile setting "*Enable compatibility with modified Direct3D runtime libraries*" allowing the server to detect runtime function offsets dynamically on each 3D application startup. This provides On-Screen Display functionality support in applications using modified Direct3D runtime libraries (e.g. FXAA injection Direct3D runtimes)
- Added new profile settings allowing you to limit the framerate during gaming. Limiting the framerate during gaming can help to reduce the power consumption as well as it can improve gaming experience due to removing unwanted microstuttering effect caused by framerate fluctuations
- Now video capture clients can limit the framerate during video capture. Limiting the framerate during videocapture can improve resulting video smoothness
- Added Vector 3D On-Screen Display rendering mode support for OpenGL applications
- Added Raster 3D On-Screen Display rendering mode support for OpenGL applications using ARB shaders (e.g. ID Software's Rage)

Version 4.2.4 (published on 07.12.2011 with EVGA Precision v2.1.1)

- Added power user oriented application profile setting allowing the server to detect runtime function offsets dynamically on each 3D application startup. This provides On-Screen Display functionality support in applications using modified Direct3D runtime libraries (e.g. FXAA injection Direct3D runtimes)
- Updated profiles list

Version 4.2.3 (published on 22.11.2011 with EVGA Precision v2.1.0)

- Fixed vertex buffer overflow issue causing improper On-Screen Display rendering or crashing in Direct3D10 / Direct3D11 applications when too many items are selected to be displayed in On-Screen Display
- Updated profiles list

Version 4.2.2 (published on 02.02.2011 with EVGA Precision v2.0.4)

- Tuned vertex buffer usage strategy improves performance in Direct3D8 and Direct3D9 applications in raster 3D On-Screen Display rendering mode
- Fixed vertex buffer corruption bug, causing On-Screen Display to flicker randomly in Direct3D8 and Direct3D9 applications when raster 3D On-Screen Display rendering mode is enabled
- Now raster font uses managed texture instead of dynamic one to bypass broken high resolution dynamic texture pitch reporting on GTX 200 series graphics cards under NVIDIA 275.27 beta drivers
- Raster font texture size is no longer fixed. Now the server is dynamically selecting minimum possible texture size to fit selected font into it
- The skins have been changed slightly to provide more convenient raster font customization. Now raster font customization window can be open by clicking *"Raster 3D"* button. Holding *"Ctrl"* button while clicking *"Raster 3D"* button allows you to select pre-rendered font from bitmap file and holding *"Shift"* button while clicking *"Raster 3D"* button allows you to pre-render and save currently selected font to bitmap file
- Fixed bug causing multiple used On-Screen Display slots to overwrite each other instead of merging the slots correctly
- Added screencapture support for Direct3D10 and Direct3D11 applications using framebuffers with 10-bit RGB components (e.g. Dirt 3)
- Added screencapture support for Direct3D10 and Direct3D11 applications using BGR framebuffers (e.g. F.3.A.R.)
- Improved tabbed text formatting
- Updated profiles list

Version 4.2.0 (published on 05.05.2011 with EVGA Precision v2.0.3)

- Improved OpenGL On-Screen Display coordinates calculation for framebuffer coordinate space mode
- Added custom raster 3D fonts support to On-Screen Display in addition to the previously available vector 2D / 3D fonts. On-Screen Display 3D mode rendering mode switch has been replaced with new vector 2D / vector 3D / raster 3D rendering modes switch
- Now On-Screen Display coordinates are specified in pixels instead of zoomed units
- Now it is possible to use drag-and-drop in On-Screen Display preview window to adjust On-Screen Display position
- Now On-Screen Display supports tabulation symbols for better appearance in conjunction with new variable width custom fonts
- Improved desktop windows notification mechanism for faster server startup and shutdown
- Windows Sidebar is now captured properly on desktop screenshots
- Updated system profile template mapping rules for application profiles
- Updated profiles list

Version 4.0.1 (published on 24.02.2011 with MSI Afterburner v2.1.0)

- Various video capture engine fixes and optimizations
- Updated profiles list

Version 4.0.0 (published on 17.01.2011 with EVGA Precision v2.0.2)

- Reduced On-Screen Display rendering related CPU performance hit due to more effective geometry batching in On-Screen Display 3D rendering mode codepath
- Now the server supports RivaTuner's user extendable localization system
- Screenshot compression library SaveImage.dll has been replaced with more progressive SaveMedia.dll library providing unified screenshot and video encoding engine
- JPEG screenshot quality settings are no longer power user oriented and can be adjusted directly from the client application GUI
- Screen capture implementation is now multithreaded to minimize the impact on the game performance when saving screenshots on the systems with multicore CPUs
- Text indication of screen capture events in the On-Screen Display has been replaced with graphics progress indicator
- Fixed bug causing the On-Screen Display to be rendered in wrong colors in 3D mode in some multitextured Direct3D9 applications (e.g. several bumpmapping related samples from DirectX 9 SDK)

- Added high performance realtime in-game video capture engine. You no longer need to waste your money on purchasing an additional video capture application, now RivaTuner Statistics Server is providing such functionality to everyone and absolutely for free! The key features of video capture engine are:
    - Realtime video capture support for any Direct3D8, Direct3D9, Direct3D10, Direct3D11 and OpenGL applications
    - Queued frame capture algorithms are aimed to minimize the graphics pipeline stalling caused by transferring pixel data from GPU to CPU and keep high and smooth in-game framerate while video capture is in progress
    - Multiple video capture formats: uncompressed video capture for the systems with high performance disk I/O subsystems, two different compression modes for the systems with both mid-range and high performance multicore CPUs
    - Multithreaded SIMD optimized encoders are aimed to provide the maximum compression performance on modern multicore CPUs with SSE2 instructions support
    - Various options allowing you to tweak video capture performance on your system: customizable target video framerate, customizable target video quality, controllable multithreaded optimization and various frame downsampling modes
    - Audio stream capture and additional video capture related enhancements are coming in future versions
- Updated profiles list

Version 3.7.2 (updated on 10.08.2010 with EVGA Precision v1.9.6)

- Added On-Screen Display profile for Startcraft II : Wings of Liberty

Version 3.7.2 (published on 09.07.2010 with EVGA Precision v1.9.5)

- Now screen capture events are identified visually by text message flashing in On-Screen Display during 0.25s
- Built-in skin sizes have been reduced due to optimized compiled bitmap cache
- *"Start the task only when computer is running on AC power"* option is no longer set in the startup task settings to allow automatically starting application via the task scheduler on laptops or on some UPS models

Version 3.7.1 (published on 21.05.2010 with EVGA Precision v1.9.4)

- Improved On-Screen Display 3D rendering mode compatibility with Source engine based games and Star Trek Online
- Updated profiles list

Version 3.7.0 (published on 04.04.2010 with EVGA Precision v1.9.3)

- DirectX11 On-Screen Display and screen capture support
- DirectX screen capture engine is no longer using DirectX runtimes to capture and compress screenshots. Screen capture and compression to PNG/JPG formats in DirectX applications is now handled by own more effective LIBPNG/JPEGLIB based library
- PNG and JPG screen capture support for desktop and OpenGL applications
- Adjustable JPG screen capture quality for experienced users
- Screen capture support for DirectX10 and DirectX11 applications using multisampled antialiasing or non-standard frame buffer formats (e.g. sRGB)
- Optional On-Screen Display 3D rendering mode. New mode uses 3D accelerated DirectX functions to emulate previously used 2D framebuffer access functions and render 2D text. New rendering mode provides On-Screen Display support in DirectX10 and DirectX11 applications using multisampled antialiasing and can also drastically reduce On-Screen Display related performance hit on the systems with badly optimized 2D acceleration in display drivers
- Simplified On-Screen Display coordinate space settings. Rarely used desktop and window coordinate space modes have been merged into single framebuffer coordinate space mode
- Now the server automatically selects framebuffer coordinate space instead of viewport coordinate space when viewport is not accessible (e.g. if pure Direct3D8 device is in use)

- Improved application notification scheme during runtime profiles adjustment. Now runtime changes in On-Screen Display application profiles (e.g. On-Screen Display zoom ratio change on the fly) are reflected in affected application almost immediately instead of approximately 1 second delay in the previous versions
- Reduced runtime skin engine memory footprint due to dynamic skin storage unloading
- Improved race condition protection system for multithreaded 3D engines (e.g. Age of dragons : Origins and ArmA II)
- Improved race condition protection system for the systems running multiple On-Screen Display applications simultaneously (e.g. RivaTuner Statistics Server, FRAPS and STEAM In-Game Chat)
- Improved floating injection address hooking technology. Due to more advanced built-in code analyzer hooks can be injected deeper into 3D application code if stealth mode is enabled. These changes improve stealth mode compatibility with third party tools using similar stealth injection technologies (e.g. FRAPS v2.9.8 and newer)
- Now stealth mode cannot be toggled during 3D application runtime to reduce the risk of crashing 3D application
- Removed single screen capture per second limitation
- Updated profiles list

Version 3.6.0 (published on 11.11.2009 with MSI Afterburner v1.4.0)

- Improved desktop windows notification mechanism
- Added optional power user oriented task scheduler based startup way

Version 3.5.0 (published on 06.10.2009 with EVGA Precision v1.8.1)

- Floating injection address hooking technology, which has been introduced in the previous version, is now disabled by default and can be enabled via the *"Stealth mode"* option in the server's properties

Version 3.4.0 (published on 08.08.2009 with EVGA Precision v1.8.0)

- Added On-Screen Display rendering profile for "The Chronicles of Riddick: Assault on Dark Athena"
- Added floating injection address hooking technology aimed to reduce the risk of On-Screen Display detection by third party applications (e.g. anti-cheat systems)

Version 3.3.0 (published on 17.04.2009 with EVGA Precision v1.7.0)

- Updated profiles list

Version 3.2.1 (published on 01.04.2009 with EVGA Precision v1.6.1)

- Improved exception handling. The server no longer crashes on startup after closing the previous running instance of the server improperly (e.g. due to crash or due to forcible server process killing via the task manager)
- Various skin engine cleanups and optimizations

Version 3.2.0 (published on 26.03.2009 with EVGA Precision v1.6.0)

- New skin format v1.1 adding bitmap effects support to the skin compiler

Version 3.1.2 (published on 06.03.2009 with EVGA Precision v1.5.0)

- Updated profiles list

Version 3.1.1 (published on 25.02.2009 with RivaTuner v2.24)

- Fixed bug into startup daemon routine causing the server to be started and to reside in memory without displaying tray icon under certain conditions

Version 3.1.0 (published on 16.02.2009 with RivaTuner v2.23)

- Minor skin appearance changes, "Normal" application detection level has been renamed to "Low" for better understanding
- Improved hooking system injects code into OS kernel's LoadLibrary functions by precached offsets and improves compatibility with some protective systems, faking the real LoadLibrary function offsets (e.g. latest revisions of SecuRom coming with new games like Dead Space, Need for speed : Undercover etc). Now such games no longer crash on startup when the server is running and LoadLibrary hooking is enabled by selecting "High" application detection level
- Added profiles for 3D applications which require non-standard On-Screen Display coordinate space settings (e.g. Codename Panzers: Phase II and Alpha Prime).
- Updated profiles list

Version 3.0.0 (published on 22.12.2008 with EVGA Precision v1.4.0)

- Fully redesigned user interface:
    - Now the server supports user interface skins to conform requirements of AIB/AIC partners bundling the server to their products. User interface skinning is based upon proprietary engine, which was previously used only in commercial RivaTuner technology based products. User interface skinning usage approach allows RivaTuner fans community to change the server's appearance completely and to express themselves in new skins design. Skin format is completely open, skin compiler and decompiler are integrated in the server, so please visit techsupport forums to get more detailed information and help on the skin creation
    - Simplified beginner oriented profile settings. Many previously available power user oriented options, such as per-3D API hooking options, are now merged into more simple multi-buttons, e.g. *"Application detection level"* multi-button
    - Improved help system. Now context sensitive help system is activated by hovering mouse over the control, instead of traditional right-click and *"What's this?"* based access in the previous versions
    - On-Screen Display preview window greatly simplifies realtime On-Screen Display position, color and zoom adjustment
- Fully redesigned profiles architecture:
    - Now pre-created and user profiles are no longer stored into single file, so user profiles are no longer lost after the server reinstallation. Please take a note, that the user profiles are preserved only when the server is reinstalled in the same folder
    - Now it is no longer possible to remove critically important pre-created profiles and pre-created profile settings can be easily restored due to template based profiles architecture
    - Improved screen capture safety. The server no longer crashes on attempt to capture screenshot in DirectX9 applications without October 2006 or newer DirectX runtime libraries installed
- Improved product customization features:
    - In addition to skins and previously available product name customization feature, bundling partners can also customize product icon (including system tray icon) and define a link to own online product help page
    - Product customization information is no longer stored in editable text OEM file, now this information is encoded to protect the product from illegal modification and distribution by third parties

Version 2.8.0 (published on 27.10.2008 with EVGA Precision v1.3.3)

- Added new On-Screen Display rendering mode. Optional configurable On-Screen Display background improves On-Screen Display readability in some applications and prevents it from merging with solid color screen
- Updated profiles list

Version 2.7.0 (published on 17.07.2008 with EVGA Precision v1.3.0)

- New server uses more aggressive hooks unload routines aimed to minimize the situations which require system reboot during server uninstallation

Version 2.6.0 (published on 28.04.2008 with RivaTuner v2.09)

- Now On-Screen Display rendering routines are compatible with WPF based applications
- Now the server's manifest file specifies full administrative privileges requirement. If you're using Windows Vista with UAC enabled, it is recommended to create scheduled task launching the server simultaneously with RivaTuner to minimize amount of displayed UAC evaluation dialogs
- Changed server's own On-Screen Display rendering behavior, now the server doesn't display 3D API usage information in the On-Screen Display when framerate display is not requested by a client application
- Updated profiles list

Version 2.5.0 (published on 03.03.2008 with RivaTuner v2.07)

- Now the server supports %FRAMERATE% macro in the On-Screen Display text slot, allowing it to display application specific framerate and 3D API usage information instead of shared framerate and 3D API usage information directly sent to On-Screen Display by the client application
- Now the server is able to detect client specific framerate rendering requests and disable own statistics rendering in this case to prevent duplicating framerate info in the On-Screen Display
- Added Direct3D10 (for non-multisampled frame buffers only) screen capture support
- Extended list of supported screen capture formats. Now the server is able to capture screenshots in *.jpg and *.png formats for Direct3D9 and Direct3D10 applications. Please take a note that these formats can be also specified while capturing desktop, DirectDraw, Direct3D8 and OpenGL screenshots, but in fact it will result in storing .bmp file with *.jpg or *.png extension
- Now the server is able to capture multiple screenshots at once if multiple 3D applications are running simultaneously
- The server's startup routine is no longer trying to precache Direct3D10 runtime info under Windows XP to reduce startup time under this OS and avoid stealing the focus from other running applications
- Improved compatibility with applications using multithreaded 3D engines. Reduced risk of crashing the server when unloading it while such 3D application (e.g. Lost Planet : Extreme condition) is running
- Now the server is able to provide per-frame frametime statistics of any 3D application to automated third party benchmarking applications (e.g. TPUBench) via the named pipe
- Updated profiles list

Version 2.4.1 (published on 17.09.2007 with RivaTuner v2.04)

- Added On-Screen Display coordinates bounds checking to Direct3D10 On-Screen Display rendering code. This feature is provided as a workaround for some early Direct3D10 drivers not performing internal bounds checking of supplied data and tending to work improperly when On-Screen Display is rendered outside the screen

- Improved race condition preventing algorithm in the hooking system. New algorithm improves compatibility with multithreaded rendering engines (e.g. Bioshock) calling frame presentation routine concurrently from different threads

Version 2.4.0 (published on 22.08.2007 with HIS iTurbo v1.12)

- Added On-Screen Display rendering support for Direct3D10 applications
- Added On-Screen Display rendering support for Direct3D9 applications redefining render target in the end of frame rendering process (e.g. Overlord). Now the server is able to detect such conditions, bind render target to back buffer, render On-Screen Display then restore the previously set render target. This feature is implemented as compatibility flag and enabled by default for all applications, however power users may disable it via editing profile flags in the server's configuration file
- Improved 3D API usage detection for applications using multiple OpenGL / Direct3D renderers (e.g. Colin McRae: DIRT). Now the server stores information about the last used renderer instead of the first one
- Updated profiles list

Version 2.3.1 (published on 01.07.2007 with RivaTuner v2.02)

- Added framerate monitoring and On-Screen Display rendering support for Direct3D9 applications using IDirect3DSwapChain9 interfaces for page flipping under Vista (e.g. Tomb Raider: Legend)

Version 2.3.0 (published on 30.04.2007 with RivaTuner v2.01)

- Internal hook infrastructure has been seriously changed to provide support for Vista and future operating systems
- Added framerate monitoring for Direct3D10 applications. Please take a note that Direct3D10 On-Screen Display rendering features are currently not implemented in the server due to Direct3D10 API limitations
- Now most of the application profile's settings can be applied on the fly now. Changing profile's On-Screen Display position, visibility, size and color is immediately reflected in running 3D application now and no longer requires application restart to be applied
- Added On-Screen Display rendering workarounds for dynamically unloading OpenGL ICD drivers (e.g. ATI Catalyst OpenGL ICD driver under Vista)

Version 2.2.1 (published on 05.02.2007 with HIS iTurbo v1.11)

- Updated profiles list

Version 2.2.0 (published on 25.12.2006 with RivaTuner v2.0 Final Release)

- Added RivaTuner styled startup links checking, aimed to prevent conflicts between multiple simultaneously installed versions of the server. Now the server checks the startup registry link and offers you to correct it, it if is referring to other folder
- Added workaround for 3D applications, wasting CPU time into erroneous page flip calls (e.g. Company of Heroes during movies rendering). Now the server checks flip call status and throws failed page flip calls from the statistics to avoid registering abnormally high framerate in such applications
- Updated profiles list

Version 2.1.0 (published on 23.10.2006 with RivaTuner v2.0 Release Candidate 16.1)

- Added optional ability of hooking OS kernel's LoadLibrary function. This feature can be useful as a workaround for problems with missing framerate monitoring and On-Screen Display in some applications on certain systems. Normally this feature is not required for proper framerate monitoring and On-Screen Display rendering, however some vendors started bad practice of blocking third party hooks in some 3D applications, so try to enable it if the server's framerate monitoring and On-Screen Display rendering features don't work properly
- Improved Direct3D runtime info precaching technique. Now the server precaches offsets to hooked Direct3D runtime functions during the first startup and no longer performs Direct3D runtime reinitialization until DirectX reinstallation. New technique made the server's /F command line switch obsolete, now the server automatically uses fast load scenario after the first start and Direct3D runtime info precaching
- Added watchdog system for tracking 3D applications crashing events. Watchdog system allows the server to kick abnormally closed applications from the context of active 3D applications
- Added command line based On-Screen Display visibility enabling / disabling / toggling functions. Now the On-Screen Display can be enabled and disabled not only by hotkeys associated with explicit server loading and unloading, but without actually unloading the server. This function was added due to the server's 3D application runtime events tracking system, introduced in the previous version. Now you can disable On-Screen Display with hotkeys without unloading the server and losing its' 3D application runtime events tracking system. Please take a note that you can use newly introduced predefined applications list in regular launch item editor window for quick-creating launch items for On-Screen Display visibility toggling
- Added command line based screenshot capturing ability. Besides traditional hotkey based screenshot capturing, combining with RivaTuner's flexible monitoring thresholds programming technique, this function allows you to create a screenshot of desktop or 3D application when some system parameter gets critical (e.g. when CPU or GPU temperature reaches shutdown threshold). Please take a note that you can use newly introduced predefined applications list in regular launch item editor window for quick-creating launch item for screenshot capturing
- Vista compatible DirectX9 framerate monitoring and On-Screen Display rendering

Version 2.0.0 (published on 22.05.2006 with RivaTuner v2.0 Release Candidate 16)

- The server's hooking library internal architecture has been seriously changed for multitasking support. Now the server can load personal On-Screen Display profiles and collect independent framerate statistics / record benchmark results for up to 256 simultaneously running 3D applications
- The server's shared memory layout has been changed drastically to provide client applications access to framerate statistics of multiple simultaneously running applications. Third party software developers using the server's shared memory may refer to SDK and updated RTSSSharedMemorySample source code for details. Please take a note that new server's shared memory layout is not compatible with the previous one, so the applications designed to work with the previous shared memory layout must be recompiled to support new format
- The server's shared memory is no longer available during 3D application runtime only. Now the memory is available during whole server's life time
- Increased amount of available On-Screen Display slots. Now up to 8 applications can simultaneously capture and use independent On-Screen Display slots
- The server's *"Show own statistics in OSD when it is not in use by other application"* option has been replaced with *"Show own statistics in OSD"* option. Now the server's own statistics (instantaneous framerate or benchmark statistics) uses its' personal On-Screen Display text slot, which is not available to any other server's client applications, so the server's statistics can be displayed in On-Screen Display even if all 8 slots are in use
- Now the server redefines Direct3D application's viewport during On-Screen Display rendering when desktop / window On-Screen Display coordinate space is used. This allows the server to display On-Screen Display beyond the application's viewport

Version 1.4.0 (published on 01.12.2005 with RivaTuner v2.0 Release Candidate 15.8)

- The server's properties window is created in inactive state now to prevent the server from stealing input focus from certain 3D applications during the server's startup. This feature is useful when the server's launch is bound to a hotkey and it ensures that the server's runtime load will not cause 3D application minimization due to loss of input focus
- Now the server is automatically injected in foreground application during startup
- Added new option for On-Screen Display coordinates space selection. By default, the server uses 3D application's rendering viewport coordinate space. However certain applications (e.g. Codename Panzers: Phase II) can use multiple viewports during frame rendering. In this case On-Screen Display coordinate space origin will be located in the top-left corner f the last viewport, defined during frame rendering, which may not match with top-left corner of screen. In this case you may use new option to force different On-Screen Display coordinate space usage
- Added benchmark statistics recording ability. Please refer to FAQ to get details on benchmark statistics recording features. Now the server allows client applications to initiate framerate statistics recording either via command line or via shared memory and to stop statistics and flush it to file per client's request at desired moment of time. The statistics can be optionally displayed in the server's On-Screen Display, and it includes:
  - Total benchmarking time and total amount of rendered frames
  - Global average framerate (total amount of rendered frames divided by total benchmarking time)
  - Minimum instantaneous framerate (instantaneous framerate is measured once per second, minimum value is recorded)
  - Average instantaneous framerate (instantaneous framerate is measured once per second, then all the measurements are averaged)
  - Maximum instantaneous framerate (instantaneous framerate is measured once per second, minimum value is recorded)

Version 1.3.1 (published on 01.09.2005 with RivaTuner v2.0 Release Candidate 15.7)

- Added OEM name of the server (RTSS.exe) to the list of profiles

Version 1.3.0 (published on 23.08.2005 with HIS iTurbo v1.4.0)

- Added server window caption / tray icon tooltip customization possibility via OEM configuration file
- Changed server instance detection mechanism, allowing using multiple instances of customized servers on the same PC
- Added command line “/i” switch, allowing the server to install itself at Window startup automatically
- Added command line “/f” switch, allowing the server to use fast start algorithm and to skip 3D runtime initialization during startup. This command line switch can be used to load the server via RivaTuner's launcher directly during playing the game, without losing application's focus caused by 3D runtime initialization
- Added lowercase font to the server's On-Screen Display

Version 1.2 (published on 26.06.2005 with RivaTuner v2.0 Release Candidate 15.6)

- Polished GUI. Some controls have been realigned / renamed
- Added context sensitive help system for each option available in the server
- Some rarely used options like *"Enable Direct3D8/Direct3D9 thunk layer integration"* have been removed from GUI. Power users still can adjust these options manually via editing the RTSSHooks.cfg file
- Added *"Run at Windows startup"* option, allowing the sever to be automatically started with Windows

- Added *"Show own statistics in OSD when it is not in use by other application"*, controlling the server's own On-Screen Display rendering behavior. Take a note that now this option is disabled by default
- Added flicker free On-Screen Display rendering mode for DirectDraw applications and applications using DirectDraw derived Direct3D interfaces (Direct3D7 and older). Take a note than flicker free mode is now enabled by default, and it is a bit slower than the previously available On-Screen Display rendering mode. Power users may revert the previous mode via editing the RTSSHooks.cfg file
- Added support for Direct3D9 applications, performing page flipping via IDirect3D9SwapChain interface instead of IDirect3DDevice9 interface's page flipping method
- Added support for OpenGL applications, performing page flipping via wglSwapLayerBuffers
- Added 3 more text slots for the server's On-Screen Display. Now up to 4 applications can simultaneously connect to the server's On-Screen Display and use it to display independent text info. Take a note that RTSSSharedMemorySample sample code included in the SDK has been also updated to reflect these changes. Now you can start up to 3 instances of sample application simultaneously with the server to demonstrate multiple On-Screen Display text slots usage

Version 1.1 (published on 07.05.2005 with RivaTuner v2.0 Release Candidate 15.5)

- Added support for 3D applications, loading both Direct3D and OpenGL runtime libraries simultaneously (e.g. 3D Studio MAX 6 / 7, Warcraft 3)
- Added server customization dialog, allowing you to change the server's hooks injection behavior either globally or at per-application level. This feature can be used to disable the server's hooks injection for certain 3D applications in case of any incompatibilities between them. Some additional parameters can be also specified either globally or on per-application level via this dialog
- Added framerate statistics collection ability for windowed Direct3D8 / Direct3D9 applications. Take a note that this feature requires Direct3D runtime-level hooks injection opposing to more unified and safe OS thunk-level hooks injection, so now the server provides you both ways, which you can optionally choose via the server's customization dialog
- Added experimental framerate statistics collection ability for fullscreen DirectDraw applications and Direct3D applications using DirectDraw derived Direct3D interfaces (Direct3D7 and older interfaces). Take a note that this feature is disabled by default
- Added On-Screen Display for fullscreen DirectDraw and pre-Direct3D8 applications and fullscreen / windowed Direct3D8 / Direct3D9 / OpenGL applications. Kudos and credits goes to Beepa for his original On-Screen Display concepts, firstly used in FRAPS product. The server's on-screen display provides you the following functionality:
    - By default, the server uses On-Screen Display to display its' own framerate statistics, so now it can be used as a standalone application for framerate monitoring, if you want to see an instantaneous framerate only and don't need extended framerate graphs built by RivaTuner's hardware monitoring module
    - The server's On-Screen Display is a shared resource, and it can be used by any third party tool for displaying any text information. Using this technology, RivaTuner can optionally connect to the server's on-screen display and show current value, retrieved from any selected hardware monitoring module data source (e.g. core temperature). Take a note that now the SDK includes a sample, demonstrating On-Screen Display usage by third-party utility
    - The server's customization dialog allows altering On-Screen Display size, position and colors either globally or at per-application level. You can also separately disable On-Screen Display for any 3D API at application level
- Added original race condition preventing algorithm to the hooking system. Now the server analyzes hooked code and always tries to inject itself in the end of hooked chain, instead of injecting itself over other hook. Also, the server now detects when other applications' attempts to inject in the head of chain, and unloads itself from the memory when there is race condition threat

Version 1.0 (published on 04.03.2005 with RivaTuner v2.0 Release Candidate 15.4)

- Framerate statistics collecting for fullscreen Direct3D8 / Direct3D9 applications (for Windows XP only). RivaTuner Statistics Server uses original OS GDI thunk hooking approach, which is not used in any of currently existing framerate monitoring software and greatly simplifies DirectX framerate monitoring implementation
- Framerate statistics collecting for both windowed and fullscreen OpenGL applications. Known limitations: currently total FPS is monitored when multiple windowed OpenGL applications are running simultaneously
- Ability to unload currently loaded RivaTuner Statistics Server via launching new RivaTuner Statistics Server instance with *"-u"* command line switch
- Even considering that performance impact of RivaTuner Statistics Server is rather low (above 1%), there is still room for optimization left, so the performance impact can be and will be reduced in the future versions
- An ability of using RivaTuner Statistics Server's statistics in any other application via named shared memory. The sample code is available upon e-mail request, sample application will be included in SDK in the next version