

## Product Release Notice

### RTX64 4.1

---

#### General Availability Release Date

April 8, 2021

#### Product Overview

RTX64 4.1 is the latest 64-bit version of IntervalZero's market-leading hard real-time software products. This release provides a number of new features, usability improvements, and resolved issues.

To see a full list of all new features and a detailed list of new APIs added and issues resolved, please refer to the [product release notes](#), found in the online RTX64 4.x Help.

#### Key Features

##### General Performance

- Improves the performance of RTSS-to-Windows communication on Windows 10 Feature Update Version 2004 and 2009 from the initial implementation in the RTX64 4.0.1 patch release. (9578)
- Implements Internal Subsystem Timer Compensation to account for SMI and other non-maskable system behaviors that impact latency and cause jitter. (8967)

##### RTX64 Console

- RTX64 Console offers an output display console window that can be configured to display for each real-time process or as a single instance for all real-time processes. This new RTX64 Console replaces the RTX64 Server Console of previous releases.
- Adds functionality to the Control Panel to support configuration of RTX64 Console (one or per process) and logging of real-time application output.

- Adds new Native Framework functions `RtfwSetConsoleConfigurationEx` and `RtfwGetConsoleConfigurationEx` along with structure `RTFW_CONSOLE_CONFIGURATION_EX` for configuring real-time application output consoles. (9239)
- Adds new properties and methods to Managed Framework class `IntervalZero.RTX64.Config.ServerConsole`. (9239)

## Remote Attach Application Debugging

- Adds support for attaching the Visual Studio debugger to an RTSS process running on a remote system, outside of the Visual Studio IDE. Visual Studio 2019, 2017 and 2015 are all supported.
- Adds functionality to the RTX64 Control Panel to support configuration of remote debugging connections.
- Adds new Native Framework functions `RtfwGetRemoteDebuggerConfiguration` and `RtfwSetRemoteDebuggerConfiguration` along with structure `RTFW_REMOTE_DEBUGGER_CONFIGURATION` for configuring remote debugging.
- Adds new properties and methods to Managed Framework class `IntervalZero.RTX64.Config.RemoteDebugger` for configuring remote debugging.

## Improvements

### Tools & Utilities

- Improves RTX64 Analyzer to display a more complete installation history. (7898)
- Adds functionality to `RtssRun` and RTX64 Task Manager that allows the user to specify the time-of-allocation for the process external MSpace. (9640)
- Adds the ability to reset default values for the memory profile settings Available system memory and Percentage expected to be used by Windows. (8844)

### Real-Time APIs

- Adds new Real-time APIs `RtSetTimeQuantum` and `RtGetTimeQuantum` for setting and retrieving a thread's time quantum value.
- Adds new Real-time API `RtQuerySharedMemory` that retrieve shared memory information.
- Adds attribute value `RT_PROC_THREAD_ATTRIBUTE_ALLOC_EXTERNAL_MSPACE_AT_PROCESS_START` to Real-

time function `RtUpdateProcThreadAttribute`. This value specifies the time-of-allocation for the process external MSpace. (9635)

- Adds logic to Real-time function `RtTerminateProcess` so that it now fails and sets last error to `ERROR_INVALID_HANDLE` when an invalid handle is specified for parameter `hProcess`. (8000)

## Native Framework APIs

- Adds new member `AllocateExtMSpaceAtStartup` to structure `RTFW_SCHEDULED_PROCESS` that specifies whether the process will allocate the process external MSpace at process start. (9224)

## Managed Framework APIs

- Adds new property `AllocateProcessExtMSpaceAtStartup` to Managed Framework class `IntervalZero.RTX64.Config.ScheduledProcess` that specifies whether the process will allocate the process external MSpace at process start. (9224)
- Adds new overload `Start(Dictionary)` to the Managed Framework `RTPProcess.Start` Method API which starts an RTSS process via a dictionary that uses the `StartParameters` enumeration as the dictionary's key type. (9224)

# Issues Resolved

## General

- Resolves an issue where RTX64 would sometimes change the system power scheme during Windows shutdown, resulting in a system hang. (9403)
- Resolves issues where some Help links in Tracealyzer targeted the wrong RTX64 Help system. (9574)
- Improves the warning messages that appears after a change is made to the RTSS boot configuration via command line. (9203)
- Resolves an issue where uninstalling the RTX64 SDK broke associations for Monitoring files (.monx and .mev), even when the Runtime was still installed. (9231)

## Subsystem

- Resolves an issue where FPU/SSE/PT states on the Intel® Celeron G3900E processor were not properly saved and restored on a thread context switch. (9680)

- Resolves an issue where the Subsystem could not be stopped following a call to a function in the Managed Framework Intervalzero.RTX64.RTAPI Namespace. (9306)

## Network and Drivers

- Resolves an issue where the X550T Network Interface Card would output error EEPROM Checksum Is Not Valid to the RTX64 Server console window. (9438)
- Resolves iPerf TCP transmit performance issues with the RtNall10GB driver. (9619)

## Tools and Utilities

- Resolves an issue where the RTX64 System Tray would not display a toast notification when the RTX64 boot configuration was not available. (7851)
- Resolves an issue where RtssRun did not display an error code when RtCreateProcess failed to launch a real-time application. (9583)
- Fixes an alignment issue and text truncation in the RTX64 Activation and Configuration utility. (9750)
- Resolves these issues with RTX64 Task Manager:
  - Resolves an issue where RTX64 Task Manager sometimes displays an unusually large value for CPU usage. (8625)
  - Resolves an issue where RTX64 Task Manager didn't accurately report CPU usage for system configurations where only one processor was assigned to Windows. (9921)
  - Resolves an issue where scheduled tasks added via the Add as scheduled task option in Task Manager did not include process arguments. (9266)

## Application Development and Debugging

- Resolves an issue where the RTX64 application template hard-coded character-set types in pre-processor definitions. This required a user to manually change the hard-coded value in order to change the character set once the project was created. (8470)
- Resolves build errors with STL code in Visual Studio 2019 version 16.6.x. (9386)

## Windows-Supported APIs

- Resolves an issue where function RegQueryValueEx returned garbage data when called within a Multi-Byte project. (9127)

## Real-Time APIs

- Resolves an issue where a Blue Screen resulted when parameter `IpApplicationName` in Real-time function `RtCreateProcess` was set to a pathname that contained spaces surrounding double-quotation marks. (8116)
- Resolves an issue where `RtCloseHandle` incorrectly succeeded in Windows build configurations when it was passed an invalid handle. (7993)
- Resolves an issue where `RtPrintf` did not fail if a null parameter was passed in. (8087)
- Resolves an issue where `RtIsAppRunnable` returned an unexpected error when given invalid parameters in a Windows build configuration. (8042)
- Resolves issues with `RtGetProcessIdealProcessor` where it would pass unexpectedly in RTSS build configurations and return the wrong error code in Windows build configurations. (7997)
- Resolves an issue where `RtCreateSharedMemory` crashed when the asking size was larger than the existing shared memory region size. (9125)
- Resolves an issue where Real-time function `RtFreeLockedMemory` returned true when its parameter was NULL. (9160)

## Native Framework APIs

- Resolves an issue where `RtfwStartTCPIPStack` did not always clear the shared memory it created. (9394)

## Managed Framework APIs

- Resolves an issue where Managed Framework API `RtProcess.WaitForExit(Int32)` returned False if the real-time process exited before calling `WaitForExit`. (8114)

## Samples

- Resolves an issue where the `MulticoreSystemResponseTimerMeasurement (mSRTM)` sample incorrectly interpreted processor indexing set by its `/p` parameter. (9270)
- Removes unnecessary build configurations for ARM and ARM64 from the RTKIPC sample project. (8282)
- Resolves an issue where the source code for samples `ManagedCodeFrameworkClientCPlusPlus` and `ManagedCodeFrameworkClientCSharp` referenced an API that was not called by the code. (9559)

- Resolves an issue where the FastSemaphore sample did not work on Virtual Machines. (8206)

## Activation & Licensing

The IntervalZero product licensing system allows for flexibility in how features are activated and deployed. Please [click here](#) for an overview of IntervalZero product licensing.

For additional information on deployment, please refer to the [RTX64 Deployment Guide](#).

## Availability

RTX64 4.1 is available beginning April 8, 2021 through Partners and by contacting Sales: [sales@intervalzero.com](mailto:sales@intervalzero.com).

We look forward to any comments and feedback. If you have any recommendations, or wish to suggest any product enhancements, please contact Product Management at: [productmanagement@intervalzero.com](mailto:productmanagement@intervalzero.com).