IntervalZero

Product Release Notice

RTX64 4.0

General Availability Release Date

May 4, 2020

Product Overview

RTX64 4.0 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.

Please note: RTX64 4.0 does not work with previous RTX64 license versions and has limited binary compatibility with previous versions of RTX64.

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.

Key Features

RTX64 4.0 contains a number of key new features described below.

Memory

RTX64 4.0 Implements a completely new local memory architecture. There is no longer one global local memory pool but multiple separate internal and external memory allocation spaces (MSpaces).

- Improves the performance of allocations and de-allocations by up to 10 times previous releases.
- Improves fragmentation to optimize memory usage within a process.
- Locks down page frames to reduce cache misses and guarantees cache coherence and TLB. Windows tradable memory will not impact the Subsystem or real-time processes.

- Provides more flexible configuration and control over how memory is used within the Subsystem and by real-time processes.
    - Expands the Real-Time API (RTAPI) to include calls to query and configure process MSpaces.
    - Expands the Managed and Native framework APIs to support easy configuration of local memory.
- Provides tools to easily configure and view memory allocation and fragmentation within a real-time process.
    - Adds a new RTX64 MSpaces command line utility, which displays local memory allocation spaces (MSpaces) of all RTSS processes.
    - Adds settings to RtssRun and Task Manager that allow you to set an initial size (*/i*) and expand (*/e*) the process external MSpace for new and scheduled tasks. (8153)
    - Improves the Visual Studio Real-Time Debugger to accept initial and expand size values for the local pool of a process's external allocation space (MSpace). (8158)
    - Adds WinDbg extension commands for displaying local memory allocation space information.
- Adds a new sample, Dynamic MSpace Configuration, which provides sample code to avoid memory request errors when memory allocation spaces (MSpaces) are configured to not expand automatically when exhausted.

Network and Drivers

- Integrates the previously standalone Network Abstraction Layer (NAL). The NAL is a network layer that abstracts the network hardware and driver functions from the upper-level protocol stacks and provides management interfaces for those upper layers to easily query for and use available network assets.
- Layers the RT-TCP/IP Stack on top of the NAL to take advantage of the NAL’s ability to better handle NIC resources.
- Re-architected the Virtual Network to provide a more robust connection and better performance.
- Improves error handling on startup of the real-time networking components. (8297)
- Expands the Real-time Network (RTN) API, NIC Driver Real-time Network Device (RTND) API, and added the Real-time Network Abstraction Layer (RtNal) functions and structures.
- Adds a concept of concurrency to help set up Network Resources based on how you are using your Network. (7874)

Tools and Utilities

- Redesigned the Control Panel to support integration of Network Abstraction Layer (NAL) functionality, layering of the RT-TCP/IP Stack on top of the NAL, and expanded memory allocation (MSpace) functionality.

Development

- RTX64 4.0 SDK can co-exist on a system with previous RTX64 3.x SDKs.

Improvements

Network and Drivers

- RTSS applications can now call both a Network Abstraction Layer (NAL) API and an RT-TCP/IP Stack API in the same application. (7877)
- Increases the maximum number of sockets from 255 to 1024. (7879)
- Increases the device name length in the RT-TCP/IP Stack and drivers to 64 bytes. (8416)
- Adds support for the Intel i219 LM4 Ethernet Connection NIC to the RtNalIPCH driver. See the *RTX64 Supported NICs* document for details. (7083)
- Improves error handling on startup of the Real-time network components. (7751)
- Adds support for interrupt moderation to the RtNalIGB, RtNalIPCH, RtNalI10GB drivers. (5343)
- Improves the RtNalIPCH driver to receive time-stamped frames. (6214)

Tools and Utilities

- Improves the Control Panel to select the correct default interrupt type for new and/or modified network interfaces based on the NIC hardware. (2718)
- Improves informational messages in the RTX64 Task Manager when a function cannot be performed. (8364)
- Adds a */c* setting to RtssRun that checks compatibility of an RTSS and RTDLL binary with the target Runtime and lists any compatibility issues.

Application Development and Debugging

- Adds a new setting to the RTX64 Application and RTDLL project templates in Visual Studio to optionally include support for the Network Abstraction Layer (NAL) component. (7941)
- Adds a new setting to the RTX64 Application and RTDLL project templates in Visual Studio to optionally include RtVision support. Note that this requires the RtVision SDK to be installed. (8526)
- Adds support for WinDbg object string Mvector Interrupt, which specifies multiple interrupt vector attaching information. (5828)
- Adds a new WinDbg extension command, !rtrdtinfo, which displays the system's RDT capability and CLOS configuration (if not ignored by RTX64).

Real-Time APIs, Native and Managed Frameworks

- Adds new Real-time APIs for querying and starting Subsystem components. (8281)
- Adds and modifies the NAL interface structure and interface initialization APIs.
- Adds real-time functionality to get default Subsystem memory behavior.
- Adds Real-time APIs and changes existing APIs to support the new local memory architecture in RTX64.
- Adds Real-time Network (RTN) functions.
- Adds NIC Driver Real-time Network Device (RTND) functions and structures.
- Adds Real-time Network Abstraction Layer (RtNal) functions and structures.
- Adds Native framework enumerations, functions and structures for determining the status of an RTX64 component, and configuring the Subsystem and Network
- Adds properties and methods to the Managed Framework classes.

Issues Resolved

Licensing

- Improves messaging in license validation errors. (8603)
- Removes support for standard dongles to resolve 3rd-party library conflicts that resulted from supporting multiple dongles. This issue typically manifested itself through automatic start issues where licenses were not recognized or the system crashed. (6945)

Subsystem

- Resolves an issue where attempts to start the RTX64 Subsystem on a VirtualBox 6.1.2 virtual machine would result in a Blue Screen. (8946)

Network and Drivers

- Resolves an issue where calling Winsock function select from an application with greater than 64 threads for the same socket can cause the process to hang the utility RtssKill if used to terminate the process. (8100)
- Resolves various issues where RtssKill did not successfully terminate socket applications. (7954)
- Resolves an error state that occurred when the IntervalZero debugger terminated an application attached to the RT-TCP/IP Stack that used local memory. (7853)
- Resolves an issue where RtssIPConfig revealed identical subnet masks of all configured IPs even when they were different. (5812)
- Resolves an issue where the return buffer in the RTND_REQUEST Network Adapter Information structure used by RtndRequest did not update correctly, which caused a crash in some scenarios. (4421)
- Resolves an issue where the RtNalIGB would not receive interrupts in MSI-X mode. (6701)
- Resolves issues where MSI-X mode interrupts did not work correctly on Intel I350 Ethernet Controller. (4296)
- Resolves an issue where only one select() call returned when the socket was closed and multiple threads made a select() call for the same socket. (8090)
- Resolves an issue where RtNalIGB devices sometimes initialized with semaphore timeout errors. (6995)
- Resolves an issue where attempts to initialize an interface resulted in an error when more than 31 Windows cores were configured. (6688)
- Resolves an issue where the TCP/IP Stack fragmented and reassembled packets larger than 64K on a Raw IP socket. (6416)
- Resolves an issue where the same IP address could be used in multiple Network Interface Cards (NICs) configured for RTX64. (5813)
- Resolves an issue where timestamps did not work with 82576 devices supported by the RtNalIGB driver. (7500)
- Fixes erroneous COM port definitions in the file RTX64pnp.inf. (8685)

- Resolves an issue where application code needed to be recompiled when the number of sockets was increased via the RTX64 Control Panel. (5165)
- Resolves an issue where stopping the RT-TCP/IP Stack stopped NAL interfaces from transmitting in some scenarios. (7194)
- Resolves an issue where debug messages appeared when the RT-TCP/IP Stack was started with an i350 Quad NIC controlled by RTX64. (4293)

Tools and Utilities

- Resolves an issue where the RTX64 Analyzer displayed inconsistent information on RTX64-controlled NICs. (7934)
- Resolves an issue where RTX64 Analyzer output was missing information on the Virtual Network. (7310)
- Resolves an issue where RTX64 System Tray would prematurely display a message that the Subsystem wasn’t licensed. (8260)
- Resolves an issue where the Monitoring tool displayed incorrect progress during conversion of a session file. (8233)
- Resolves an issue where MSRTM defaulted to RtGetClockTime instead of QueryPerformanceCounter when the */c* flag was not specified. (9012)
- Resolves an issue where the Monitor utility incorrectly converted event timestamps. (8928)
- Improves the error message that appears when RtssKill is run with a non-existent order ID for a scheduled task. (8324)

Application Development and Debugging

- Resolves an issue where a Blue Screen occurred when *free* was called on memory allocated in an RTDLL built in the RTSSDebug configuration. (8088)
- Resolves an issue where the IntervalZero Real-Time Debugger did not display the call stack for some RTSS applications that included the C Runtime. (8169)
- Resolves an issue where static 8-bit arrays caused the IntervalZero Real-Time Debugger to freeze in Visual Studio 2019. (8473)

Real-Time API, Native and Managed Framework

- Resolves an issue where Real-time Network API RtnInstallStaticRoute did not support IPv6. (3841)

- Resolves an issue where the Shutdown function failed on UDP sockets, returning error WSAEOPNOTSUPP. (7768)
- Resolves an issue where the wrong error was returned when parameter pHardware in function RtNalGetDeviceSystemTimer was set to NULL. (6991)
- Adds logic to Real-time function RtIsAppRunnable to ensure it returns FALSE when passed a binary that is not an RTSS application. (7483)
- Resolves an issue where Real-time APIs RtMonitorControl and RtMonitorChangeState returned an incorrect error code when Monitoring was paused. (8328)
- Improves RtWaitForMultipleObjects to return the appropriate error when provided a NULL handle. (8587)
- Resolves an issue where Native Framework functions RtfwCreateNetworkInterface and RtfwModifyNetworkInterface did not reject NIC configurations that contained duplicate IPv6 addresses. (6405)
- Resolves an issue where IP validation logic erroneously rejected an IP address of 123.255.123.000. (8002)
- Resolves an issue where Managed Framework Class IntervalZero.RTX64.Config.Network allowed a client to create multiple instances with the same TCP/IP Stack instance ID, but with different network interfaces in the internal interface list. (3106)
- Resolves issues with garbage collection in the Managed Framework. (8306)

Samples

- Resolves an issue where the NalDataStream sample didn't print link status. (4468)
- Resolves an issue where the NalDataStream sample worked properly even when incorrect parameters were specified. (8734)
- Resolves an issue where the RtTcpipServer.rtss application didn't validate incorrect flags. (8739)

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.

Note: RTX64 4.0 components will require new license keys. Previous RTX64 version keys will not work with 4.0.

For additional information on deployment, please refer to the RTX64 Deployment Guide.

Availability

RTX64 4.0 is available beginning May 4, 2020 through Partners and by contacting Sales: 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.