# RTX64 Features by Release


IZ-DOC-X64-0089-R10 | March 2017

# Operating System Support

| | RTX64 2013 | RTX64 2013 with SP1 | RTX64 2014 | RTX64 2014 with SP1 | RTX64 2014 with SP2 | RTX64 3.0 | RTX64 3.1 |
|---|---|---|---|---|---|---|---|
| **Windows 10** | No | No | No | No | No | Yes (Anniversary Update Version 1607) | Yes (Anniversary Update Version 1607) |
| **Windows 10 IoT** | No | No | No | No | No | No | Yes (Enterprise LTSB Version 1607) |
| **Windows 8.1** | No | No | No | No | Yes (8. 1 with Update) | Yes (8. 1 with Update) | Yes (8. 1 with Update) |
| **Windows 8** | No | Yes | Yes | Yes | No | No | No |
| **Windows 7** | Yes (SP1) | Yes (SP1) | Yes (SP1) | Yes (SP1) | Yes (SP1) | Yes (SP1) | Yes (SP1) |
| **Windows Embedded Standard 8** | No | Yes | Yes | Yes | Yes (8.1) | Yes (8.1) | Yes (8.1) |
| **Windows Embedded Standard 7** | Yes (SP1) | Yes (SP1) | Yes (SP1) | Yes (SP1) | Yes (SP1) | Yes (SP1) | Yes (SP1) |
| **Windows Vista** | No | No | No | No | No | No | No |
| **Windows XP Professional** | No | No | No | No | No | No | No |

**NOTE: Windows Home Editions are not tested.**

# Visual Studio Support

| | RTX64 2013 | RTX64 2013 with SP1 | RTX64 2014 | RTX64 2014 with SP1 | RTX64 2014 with SP2 | RTX64 3.0 | RTX64 3.1 |
|---|---|---|---|---|---|---|---|
| **Visual Studio 2017** | No | No | No | No | No | No | Yes (BETA support; only RC2 was tested) |
| **Visual Studio 2015** | No | No | No | No | Yes (requires RTX64 Visual Studio 2015 support) | Yes (Update 2; Ultimate, Premium, Pro, and Community editions supported) | Yes (Update 3; Ultimate, Premium, Pro, and Community editions supported) |
| **Visual Studio 2013** | No | No | No | Yes | Yes | Yes (Update 3; Ultimate, Premium, Pro, and Community editions supported) | *Deprecated*. Support will be removed in a future release. |
| **Visual Studio 2012** | No | Yes (Building only, no debugging) | Yes (Building only, no debugging) | Yes | Yes | Yes (Update 1 or greater) | *Deprecated*. Support will be removed in a future release. |
| **Visual Studio 2010** | Yes | Yes | Yes | No (Existing projects can be built and run) | No (Existing projects can be built and run) | No (Existing projects can be built and run) | No (Existing projects can be built and run) |

# Support for Key Features

| | RTX64 2013 | RTX64 2013 with SP1 | RTX64 2014 | RTX64 2014 with SP1 | RTX64 2014 with SP2 | RTX64 3.0 | RTX64 3.1 |
|---|---|---|---|---|---|---|---|
| **Application Wizard** | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| **Microsoft C Runtime** | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| **Local Debugging** | Yes (VS 2010) | Yes (VS 2010) | Yes (VS 2010) | Yes (VS 2013 and 2012) | Yes (VS 2013 and 2012) | Yes (VS 2015, 2013, and 2012) | Yes (VS 2017, 2015); Available but deprecated (VS 2013 and 2012) |
| **Remote Debugging** | No | No | No | Yes (VS 2013 and 2012) | Yes (VS 2013 and 2012) | Yes (VS 2015, 2013, and 2012) | Yes (VS 2017, 2015); Available but deprecated (VS 2013 and 2012) |
| **Debugger Support for Launch** | Yes (VS 2010) | Yes (VS 2010) | Yes (VS 2010) | Yes (VS 2013 and 2012) | Yes (VS 2013 and 2012) | Yes (VS 2015, 2013, and 2012) | Yes (VS 2017, 2015); Available but deprecated (VS 2013 and 2012) |
| **Debugger Support for Attach** | No | No | No | No | No | No | Yes (Local running RTSS processes) |
| **Using Windows groups for Security** | Yes | Yes | Yes (The Debuggers group was removed) | Yes | Yes | Yes | Yes |
| **Dedicated Mode (up to 63 processors for RTX64)** | Yes | Yes | Yes | Yes | Yes | Yes | Yes |

| | RTX64 2013 | RTX64 2013 with SP1 | RTX64 2014 | RTX64 2014 with SP1 | RTX64 2014 with SP2 | RTX64 3.0 | RTX64 3.1 |
|---|---|---|---|---|---|---|---|
| **Structured Exception Handling** | Yes | Yes | Yes | Yes (Global on/off, not by feature) | Yes (Global on/off, not by feature) | Yes (Global on/off, not by feature) | Yes (Global on/off, not by feature) |
| **Floating Point Support** | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| **Supports MMX, SSE/SSE2/SSE3/SSE4** | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| **Supports AVX** | Yes | Yes | Yes (2.0) | Yes (2.0) | Yes (2.0) | Yes (2.0) | Yes (2.0) |
| **Deterministic Memory Allocation** | No | Yes | Yes | Yes | Yes | Yes | Yes |
| **Plug and Play Device Support** | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| **Line-based Interrupt Support** | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| **Message-based & Extended Message-based Interrupt Support** | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| **Managed Code Interface to Configure Subsystem** | Yes | Yes | Yes | Yes | Yes | Yes | Yes |

| | RTX64 2013 | RTX64 2013 with SP1 | RTX64 2014 | RTX64 2014 with SP1 | RTX64 2014 with SP2 | RTX64 3.0 | RTX64 3.1 |
|---|---|---|---|---|---|---|---|
| **Control Panel** | No | Yes | Yes | Yes | Yes | Yes | Yes |
| **Tool for Activation and Configuration** | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| **Tool to Start and Stop Processes** | Yes (RtssRun, RtssKill) | Yes (RtssRun, RtssKill) | Yes (RtssRun, RtssKill, Task Manager) | Yes (RtssRun, RtssKill, Task Manager) | Yes (RtssRun, RtssKill, Task Manager) | Yes (RtssRun, RtssKill, Task Manager) | Yes (RtssRun, RtssKill, Task Manager) |
| **Console to Display Process Output** | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| **Tool to Evaluate Performance** | No | No | Yes (Latency View) | Yes (Latency View) | Yes (Latency View) | Yes (Latency View) | Yes (Latency View) |
| **Tool to Trace Process Behavior** | No | No | Yes (Monitor) | Yes (Monitor) | Yes (Monitor) | Yes (Monitor) | Yes (Monitor) |
| **Tool to Display Process Behavior** | No | No | No | No | No | Text file (Runtime)<br>Tracealyzer (SDK) | Text file (Runtime)<br>Tracealyzer (SDK) |
| **Tool to Show CPU Usage** | No | No | No | No | Yes (command line) | Yes (command line) | Yes (command line) |
| **Tool to Show Object State** | No | No | No | No | Yes (command line) | Yes (command line) | Yes (command line) |
| **Command Line Tool to Gather System Information** | Yes (RTX64 Analyzer) | Yes (RTX64 Analyzer) | Yes (RTX64 Analyzer) | Yes (RTX64 Analyzer) | Yes (RTX64 Analyzer) | Yes (RTX64 Analyzer) | Yes (RTX64 Analyzer) |

# Network Support

| | RTX64 2013 | RTX64 2013 with SP1 | RTX64 2014 | RTX64 2014 with SP1 | RTX64 2014 with SP2 | RTX64 3.0 | RTX64 3.1 |
|---|---|---|---|---|---|---|---|
| **IPv4** | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| **IPv6** | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| **UDP** | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| **TCP** | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| **ICMPv4** | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| **ICMPv6** | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| **ARP** | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| **Ethernet** | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| **Multicast** | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| **Raw Socket Support** | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| **Jumbo Packets** | No | No | Yes (driver dependent) | Yes (driver dependent) | Yes (driver dependent) | Yes (driver dependent) | Yes (driver dependent) |
| **MAC Layer Filter Support** | Yes (single layer) | Yes (single layer) | Yes (single layer) | Yes (single layer) | Yes (single layer) | Yes (single layer) | Yes (single layer) |
| **Basic Winsock 2.0 Support** | Yes | Yes | Yes | Yes | Yes | Yes | Yes |

| | RTX64 2013 | RTX64 2013 with SP1 | RTX64 2014 | RTX64 2014 with SP1 | RTX64 2014 with SP2 | RTX64 3.0 | RTX64 3.1 |
|---|---|---|---|---|---|---|---|
| **Virtual Network** | No | No | Yes (single network) | Yes (single network) | Yes (single network) | Yes (single network) | Yes (single network) |
| **Networking Tools** | Yes (RtssArp, RtssIpConfig, RtssPing) | Yes (RtssArp, RtssIpConfig, RtssPing) | Yes (RtssArp, RtssIpConfig, RtssPing, RtssRoute) | Yes (RtssArp, RtssIpConfig, RtssPing, RtssRoute) | Yes (RtssArp, RtssIpConfig, RtssPing, RtssRoute) | Yes (RtssArp, RtssIpConfig, RtssPing, RtssRoute) | Yes (RtssArp, RtssIpConfig, RtssPing, RtssRoute) |

# Key Features by Release

## RTX64 3.1

Key Features

- Subsystem
  - Improved Subsystem performance by a range of 16%-33%, depending on system cache architecture. (987)
- Tools and Utilities
  - Improved monitoring event capture to gather information about processes running before monitoring is started. (5043)
  - Implemented these improvements in the RTX64 Analyzer:
    - Analyzer output now contains a listing of the installed versions of .NET (5284)
    - Analyzer output now contains a listing of the RTX64 DLLs in the Global Assembly Cache (GAC). (5284)
    - Analyzer output now contains the contents of internal Registry keys. (5285)
- Application Development and Debugging
  - Added the feature Attach to RTSS Process, which allows you to attach the Visual Studio 2015 debugger to any running RTSS process on the local machine.
  - Added support for Intel Compiler 17.0.1 ( as shipped with Intel Parallel Studio XE 2017 Update 1). (5317)
  - Added a new debugging property in Visual Studio that allows you to allocate memory from the Windows memory pool, which uses non-deterministically allocated memory. Note that this only applies when the default memory allocation behavior is set to use Local memory. (4257)
  - Added BETA support for Visual Studio 2017 (RC2 was tested).
- SDK
  - Added new Real-Time Network functions:
    - RtnAttachProcessExitHandler registers an application's networking exit handler to allow an RTSS application to perform custom socket code cleanup when an application exits.
    - RtnReleaseProcessExitHandler removes an application's networking exit handler registered by the function RtnAttachProcessExitHandler.

    (5355)

- Added support for these Interlocked functions in real-time applications:
  - InterlockedAdd
  - InterlockedAdd64
  - InterlockedAnd
  - InterlockedAnd8
  - InterlockedAnd16
  - InterlockedAnd64
  - InterlockedCompare64Exchange128
  - InterlockedCompareExchange16
  - InterlockedCompareExchange64
  - InterlockedCompareExchange128
  - InterlockedCompareExchangePointer
  - InterlockedDecrement16
  - InterlockedDecrement64
  - InterlockedExchange8
  - InterlockedExchange16
  - InterlockedExchange64
  - InterlockedExchangePointer
  - InterlockedExchangeSubtract
  - InterlockedExchangeAdd64
  - InterlockedIncrement16
  - InterlockedIncrement64
  - InterlockedOr
  - InterlockedOr8
  - InterlockedOr16
  - InterlockedOr64
  - InterlockedXor
  - InterlockedXor8
  - InterlockedXor16
  - InterlockedXor64

- Added a new Real-Time API RtIsDebuggerPresent, which determines whether a local real-time process is attached to the IntervalZero Real-Time Debugger. (5417)
- Added support for these C Runtime functions:
  - errno
  - fwrite
  - fflush
  - ferror
  - feof
  - clearerr
- See the *Matrix of C Library Supported Functions* in the RTX64 Help for the full list. (5280)

# RTX64 3.0

Key Features

- General
  - Added support for Windows 10 Anniversary Update Version 1607.
  - Added support for Windows 10 Version 1511 (Feb 2016). (4598)
  - Added user notification pop-ups when a IntervalZero-dongle is plugged in and is available for use (4599)
- RT-TCP/IP Stack and Drivers
  - Added support for the Intel x540 10Gb/s network adapter (0x1528) through the RTX64 RtI10GB driver. This driver also includes untested support for several other Intel network adapters in the same family. See the RTX64 Supported NICs document, available at https://www.intervalzero.com/technical-support/guides-and-minitutorials/, for a list of supported devices. (4567)
  - Incorporated the RtBCM driver, which had previously been released as a standalone. See the RTX64 Supported NICs document, available at https://www.intervalzero.com/technical-support/guides-and-minitutorials/, for a list of supported devices. (4566)
  - Resolved performance issues that occurred when socket applications were run on gigabit connections and on at least two different cores. (4663)
  - Added support for the Intel® I210 Flash-less Copper-only Ethernet Controller (device ID 0x157B) to the RtIGB driver. (4531)
  - Added support for the Intel® PRO/1000 PF Dual Port Server Adapter (device ID 0x105F) to the RtE1000 driver. (4880)

- Tools and Utilities
  - Added Tracealyzer for RTX64, a diagnostics tool from Percepio for viewing monitoring session data, to the SDK. (4692)
  - New Monitoring features:
    - Improved monitor event MonitorEventSemaphoreRelease to include the handle of the semaphore. (4314)
    - Improved monitor events MonitorEventWFSOReturned and MonitorEventWFMOReturned to return a pointer to the objects they acted upon. (4315)
    - Expanded monitor events to include handles for Release events. (4312)
    - Improved Critical Section monitor events to include the mutex handle of the critical section object. (4701)
    - Added these new monitor events: (4904, 4432)
      - File Object Create – Generated when a file has been opened by the subsystem.
      - File Object Destroy – Generated when a file has been destroyed by the subsystem.
      - SRI to Windows Event – Generated when an SRI over to Windows has been initiated.
      - SRI to Windows Return Event – Generated when an SRI over to Windows has returned.
      - Thread Terminate – Generated when a thread terminates.
  - Improved the Latency View tool to allow each graph view to be scaled independently and to log the duration of the test run. (4414, 1668, 3548)
  - Added the following information to RTX64 Analyzer output:
    - System boot configurations. (4798)

      **NOTE: In order to display boot configuration data when running RTX64 Analyzer from the Start Menu shortcut, you must right-click and choose Run as administrator.**

    - Monitor configuration information (4792)
    - VSIX Extensions installed on the machine (4784)
    - On Windows 8.1 systems: whether the supported Windows 8.1 version (Windows 8.1 with Update) is installed. (4793)
- SDK
  - Integrated standalone Visual Studio 2015 with Update 2 build and debug support, including (4844, 4530):
    - Visual Studio templates for creating a RTX64 Application and RTDLL.
    - Support for statically linked Debug and Release versions of the Microsoft Visual Studio C Runtime.
    - Added Snippets for some key RTAPI function calls.
    - Added debugging support through launch.

        - Added support for Start Without Debugging within the Visual Studio debugger.
        - Added support for launching a RTSS process on a remote target system for debugging.
    - Added support for Intel Compiler 16.0. (4396, 4406)
    - Added custom error codes that can be returned by Real-Time APIs. (4601, 4459)
    - Added a new Real-Time function RtGetModuleFileNameEx, which retrieves the fully-qualified path for the file that contains the specified module. (4470)
    - Added new RTK API functions that verify whether a specified RTX64 Runtime or RT-TCP/IP Stack version is installed and licensed (4660):
        - RtkIsRuntimeLicensed – Verifies that a specified RTX64 Runtime is installed and has a valid license.
        - RtkIsTcpStackLicensed – Verifies that a specified RTX64 RT-TCP/IP Stack component is installed and has a valid license.
    - Added new API functions that return whether the provided RTSS application binary can be run, which means it has been built to run with the provided license feature and that there is a valid license for the feature on the system. (4643)
        - RtIsAppRunnable
        - RtkIsAppRunnable
    - Added new API functions that return the version of the installed RTX64 Runtime (3151):
        - RtGetRuntimeVersionEx
        - RtkGetRuntimeVersionEx
        - RT_VERSION_INFO
    - Added support for the Windows-supported API function GetModuleHandleEx, which retrieves a module handle for the specified module in a specified process. (3176)
    - Added support for these C-Runtime functions: (4541)
        - _beginthread, _beginthreadex
        - _endthread, _endthreadex
    - Added new Real-Time API functions for setting and retrieving the flush TLB tick mod of a RTSS process. By default, a processor's TLB cache is flushed when the processor is idle (4690):
        - RtSetFlushTLBTickMod
        - RtGetFlushTLBTickMod
    - Added new Real-Time API functions for setting and retrieving the time quantum value for a specified thread, in microseconds (4714):
        - RtSetThreadTimeQuantumEx
        - RtGetThreadTimeQuantumEx

    - Added support for Winsock function WSAEnumNetworkEvents, which discovers occurrences of network events for the indicated socket. Additionally, support was added for structure WSANETWORKEVENTS, used to store a socket's internal information about network events for WSAEnumNetworkEvents. (4703)
    - Improved performance in the Real-Time Network Device function RtndFrameTransmit. (4688)
    - Added a new Real-Time API function, RtEnumProcessEx, which retrieves the process identifier for each RTSS process object using data specified by new structure RTPROCESS_INFORMATION. (4775)
- Samples
    - Added a Start Menu entry that points directly to the location to which Samples are installed by default: %public%\Documents\IntervalZero\RTX64 SDK\\Samples. You can also access RTX64 program samples directly from the RTX64 SDK Welcome Screen. (3579, 4564)
    - Updated the RTKIPC sample, which now requires Visual Studio 2015 and WDK 10 to build. The resulting binary will run on all Windows versions supported by RTX64 (7, 8.1, and 10). As part of this update, Visual Studio 2012 and Visual Studio 2013 versions of the sample are no longer provided. (4877, 4885)
    - Added an Advanced Installer sample to the Merge Modules installation showing how to use the RTX64 Runtime merge modules. (4966, 4705)

# RTX64 2014 with Service Pack 2

Key Features

- General
    - Added support for Windows 8.1 with Update (3220, 3221).
- Activation and Configuration
    - Added support for smaller form factor dongles that can hold license files. (3394)
    - Added functionality to provide dongle information through command line utilities and API calls. (3880)
- Subsystem
    - You can now configure a Search Path to allow RTX64 to load an RTSS application or RTDLL by base filename only, assuming one of the directories in the search path contains the file you want to load. (3895, 3709, 2027)
- Tools and Utilities

- Added a new RTX64 CPU Usage utility, which displays CPU usage information for all RTSS cores on the system. (3782)
- Added a new RTX64 Objects utility, which displays information about RTSS processes and their associated objects, such as events, semaphores, and loaded RTDLLs. (3454)

- Monitoring
  - Provided the ability to associate a monitor event with a trigger to start monitoring (2971)
  - Improved the Timer Set monitoring event to include fields for Expiration Period and Interval Period. (3302)
  - Redesigned the Monitor utility and added support for the optional association of monitor events with triggers. (3376)
  - Added new memory free failure events to the Monitor utility (3410):
    - Local Memory Free Fail – Represents failed freeing of local memory.
    - TLS Free Fail – Represents a failed call to TlsFree by a user application.
    - Heap Free Fail – Represents failed freeing of memory via HeapFree().
    - Contiguous Memory Free Fail – Represents failed freeing of contiguous memory
    - Windows Memory Free Fail – Represents failed freeing of memory to Windows.
- RT-TCP/IP Stack and Drivers
  - Added support for the Intel Gigabit ET2 Quad-Port driver. (3323)
  - Added support for the Intel i217 and i218 Network Interface Cards (NICs). (3711, 3228, 3100)
- Debugging
  - Added new debugging properties to application Property Pages in Visual Studio (3848). You can:
    - Choose to override default subsystem behavior to allocate memory from the RTX64 local memory pool, which uses deterministically allocated memory.
    - Set the ideal processor on which the main thread of the debugged process will run.
    - Set the affinity mask that specifies the processor(s) on which the debugged process will run.
- SDK
  - Added new API functions that list the serial number for each dongle connected to the machine. (2676)
    - RtGetDongles
    - GetDonglesInfo (available from the Managed Code Framework)
  - Added two new API functions that retrieve information about licenses installed on the system. (3034)
    - RtGetLicenses
    - RtkGetLicenses
  - Added new API functions that verify whether a specified RTX64 Runtime or RT-TCP/IP Stack version is installed and licensed:

        - RtIsRuntimeLicensed – Verifies that a specified RTX64 Runtime is installed and has a valid license.
        - RtIsTcpStackLicensed – Verifies that a specified RTX64 RT-TCP/IP Stack component is installed and has a valid license.
    - Added support for an IP Helper  function and structures (3536):
        - GetAdaptersAddresses, which retrieves the addresses associated with the adapters on the local computer.
        - IP_ADAPTER_ADDRESSES, which is the header node for a linked list of addresses for a particular adapter.
        - IP_ADAPTER_UNICAST_ADDRESS, which stores a single unicast IP address in a linked list of IP addresses for a particular adapter.
    - Expanded the enumeration RTX64_MONITOR_CONTROL_OP to include operations for setting and resetting triggers for monitoring events (4139):
        - MONITOR_CONTROL_SET_EVENT_TRIGGERS – Deterministically sets triggers for monitoring events.
        - MONITOR_CONTROL_RESET_EVENT_TRIGGERS – Deterministically resets (i.e., turns off) triggers for monitoring events.
        - MONITOR_CONTROL_SET_CUSTOM_EVENT_TRIGGERS – Deterministically sets triggers for custom monitoring events.
        - MONITOR_CONTROL_RESET_CUSTOM_EVENT_TRIGGERS – Deterministically resets (i.e., turns off) triggers for custom monitoring events.
    - Added Seek functionality to the Monitoring API. (3819)
- Samples
    - Added a new SimpleProducerConsumer sample that demonstrates performance impact caused by CPU cache. The sample builds two applications: a consumer and a producer. (2774)
    - Added a new RTX64Config sample that contains a subset of the source code for RTX64Config, the command-line tool for configuration and control of the RTX64 Runtime component.  This sample demonstrates how to use the RTX64 Managed Code Framework (IntervalZero.RTX64.dll), which provides programmatic access to all RTX64 configuration and control operations. (2834)

# RTX64 2014 with Service Pack 1

Key Features

- Tools and Utilities
    - Added support to log Windows custom events with support for RtGenerateEvent under Windows. (2972)
    - Added optimizations to reduce generation of monitoring events to no more than 20 instructions in real-time code paths. (2870)

- RT-TCP/IP Stack and Drivers
  - Increased the size of datagram packets used by the RT-TCP/IP Stack from 8k to 64k. (3480)
  - Added support for IGMPv3, MLDv2, and IPV6_MULTICAST_LOOP in the RT-TCP/IP Stack. (3573)
  - Incorporated Intel 82579 controller support into this release of RTX64. (3436)
- Debugging
  - Added support to Visual Studio 2012 and 2013 for local and remote debugging of Real-time applications in Visual Studio:
    - Added local debugging support through launch. (1577, 3211)
    - Added support for Start without Debugging within the Visual Studio debugger. (2643)
    - Added support for launching a RTSS process on a remote target system for debugging. (3213)
- Software Development Kit
  - Added development support for Visual Studio 2013.
  - Added new Visual Studio templates for creating a RTSS Application or Real-Time DLLs.
  - Added the API call RtEnumProxyProcesses, which enumerates proxy processes associated with Windows processes linked to RTAPI.
  - Added a collection of code snippets for Real-Time API calls that can be inserted where you need them in your code in Visual Studio.
  - Added two new Winsock API calls:
    - inet_ntop – Converts an IPv4 or IPv6 Internet network address into a string in Internet standard format.
    - inet_pton – Converts an IPv4 or IPv6 Internet network address in its standard text presentation form into its numeric binary form.
  - Added support for additional socket options in the RTX64 Winsock API. For the complete list of supported options, see getsockopt and setsockopt. (3554)

# RTX64 2014

Key Features

- Real-time Subsystem
  - Added monitoring infrastructure that allows you to profile real-time application behavior.
  - Added AVX 2.0 support.

    - Improved thread priority mapping between Windows processes and how they interact with the real-time subsystem. You now have complete control over how Window proxy threads interact with RTSS thread and objects.
- Tools and Utilities
    - Improved the control panel to allow for modification of Network interface friendly names.
    - Improved error messages provided by tools when RTSS binaries are incorrectly stamped or user does not have the proper permissions.
    - The control panel now provides the ability to disable the logic used by the RTX64 subsystem to prevent windows power management of processor speeds.
    - Added new tools:
        - Task Manager – view active Real-time processes (.rtss) and windows processes linked to RTX64 (.exe). You can start new tasks and terminate currently running tasks.
        - Monitor – start and stop monitoring sessions, and generate log files of monitoring results.
    - Analyzer now includes information about the status of Virtual Network components.
- RT-TCP/IP Stack and Drivers
    - Added a Virtual Network that provides a virtual point-to-point connection between Windows and RTX64. It emulates a local area network connection between Windows and the Real-time Subsystem with no additional hardware required.
- Debugging
    - Added the RTX64 WinDbg Extension, which extends Microsoft's 64-bit version of WinDbg and provides a way to analyze and interpret the state of RTSS applications and the RTX64 Subsystem.
- Software Development Kit
    - The managed code framework now provides functionality to disable the logic used by the RTX64 subsystem to prevent windows power management of processor speeds.
    - The Managed code interface now provides functionality to allow Windows applications to enumerate RTSS processes.
    - Added support for multiple RTX64 SDKs on the same system through common tools and versioned build environments.
    - New RTAPI calls were added to support getting and setting of priorities for Windows proxy threads. New API calls are as follows:
        - RtSetProxyThreadPriority - sets the priority of a RTSS proxy thread from a Windows application.
        - RtGetProxyThreadPriority - gets the priority of a RTSS proxy thread from a Windows application.
        - RtkSetProxyThreadPriority - sets the priority of a RTSS proxy thread from a Windows kernel driver.
        - RtkGetProxyThreadPriority - gets the priority of a RTSS proxy thread from a Windows kernel driver.
    - Added an API call RtGetEnabledXStateFeature which allows developers to determine the capabilities of the system processor.

- Added new API calls to support Monitoring functionality:
  - RtGenerateEvent – allows you to generate user defined events within an RTSS process.
  - RtMonitorControl – allows you to control monitoring within an RTSS process programmatically.
- Added new API calls to support link status monitoring:
  - RtnIsDeviceOnline – gets the online status of a network device for link status monitoring.
  - RtnIsStackOnline – gets the online status of the RT-TCP/IP stack for link status monitoring.
- Added support for the Windows API call GetModuleFileName, which retrieves a handle for each module in a specified process.
- Added support for the Windows API call TryEnterCriticalSection, which attempts to enter a critical section without blocking. If the call is successful, the calling thread takes ownership of the critical section.

# RTX64 2013 with Service Pack 1 Update 1

Key Features

- Incorporated the updated RtIGB driver, which supports the Intel® i350 controller.

# RTX64 2013 with Service Pack 1

Key Features

- Added support for the Windows 8 64-bit operating system on non 2xAPIC systems.
- Added support for Windows Kernel driver to communicate with RTSS applications through a Real-Time Kernel API (RTKAPI) functions.
- Activation and Configuration
  - Added the -v option to RTX64ActivationUtil.exe that validates whether a activation was successful. This parameter will also display the current license(s). (VAN-1338)
- Real-time Subsystem
  - Streamlined the process for converting a Windows PCI/PCIe device to RTX64 control, and vice-versa, using Windows Device Manager for conversion.

    - Properly implemented local memory. Initial implementation did not work, and customers were instructed not to use it.
- Tools and Utilities
    - RTX64 Control Panel – provides a number of settings that let you determine how the Subsystem, RTSS applications, and RT-TCP/IP Stack (if purchased) behave. It also allows you to view information specific to your version, access product documentation, and launch the RTX64 Analyzer utility.
    - RTX64 Latency View – display a visual comparison of latency between Windows and RTX64 cores.
    - Added a new /default flag to RTX64Config that resets the local memory expansion size to the default value.
    - Modified the RTX64 Server to support writing to a log file.
- RT-TCP/IP Stack and Drivers
    - Added the RtIGB Real-time Network Driver that supports Intel i210 and i350 NIC cards.
- Software Development Kit
    - Added support for Windows Kernel driver to communicate with RTSS applications through a Real-Time Kernel API (RTKAPI) functions. For more information on the available API functions, see the section Windows Driver IPC API (RTKAPI) Reference in the RTX64 Help.
    - Added new Real-time API functions:
        - RtWaitForSingleObjectEx – allows a thread to wait on an object to be signaled with 100 nanosecond time-out interval.
        - RtWaitForMultipleObjectsEx – allows a thread to wait on one of multiple objects to be signaled with 100 nanosecond time-out interval.
        - RtQueryPerformanceCounter – retrieves the current value of the high-resolution performance counter (based on the processor's time-stamp counter, TSC).
        - RtQueryPerformanceFrequency – retrieves the frequency of the high-resolution performance counter (based on the processor's time-stamp counter, TSC).
        - RtShrinkLocalMemory – shrinks the RTSS local memory pool up to the size specified.
        - RtAllocateLocalMemoryEx – allocates memory from a pre-allocated RTSS local memory pool to avoid SRI activity if allocating memory from the Windows memory pool.
    - Added new Real-time Network Device (RTND) API functions:
        - RtndReceiveFilterEx – called when an incoming Ethernet frame is received from the NIC driver. This function, if implemented, will be called instead of RtndReceiveFilter. It supplies an additional parameter when the filter routine is called. This additional parameter is a pointer to the network interface in question.

        - RtndTransmitFilterEx – called when an outgoing Ethernet frame is received from the stack. This function, if implemented, will be called instead of RtndTransmitFilter. It supplies an additional parameter when the filter routine is called. This additional parameter is a pointer to the network interface in question.
    - Added Visual Studio 2012 build support, including:
        - RTX64 Application wizard
        - Support for statically linked Debug and Release versions of the Microsoft Visual Studio C Runtime support.
    - Additional samples to show RTX64 functionality:
        - RTK IPC – Real-time kernel driver sample that shows inter-process communication with an RTSS application.
        - Simple Ping Pong – multiple thread sample that shows synchronization between 2 threads using an event.
        - Simple Data Exchange – multiple process sample that shows synchronization and communication between 2 processes using events and shared memory.
        - Mailbox – sample to show how customers could implement their own mailboxes to use within RTSS processes.
        - Windows RTX64 using STL – sample to show how one can develop RTSS applications using the Microsoft Standard Template libraries.

# RTX64 2013 Update 1

Key Features

- Improved Product and Component Activation
    - Support for dongle-based licensing. You can activate RTX64 components to an IntervalZero-provided dongle, which you can then connect to different machines to activate installed components.
    - Optionally create a proxy connection from the RTX64 Activation and Configuration dialog when a network connection cannot be established.
- The RTX64 Software Development Kit provides a new Real-Time function, RtGetLicenseFeatureStatusEx, which obtains the status of the license for a given product feature.

# RTX64 2013

Key Features

- Real-time Subsystem
  - Scalable from 1 to 63 real-time processors
  - SMP aware scheduler utilizes both priority driven and pre-emptive algorithms to ensure critical thread context switches; and yields to threads of high priority occur in the sub-microsecond range.
  - Real-time Win32 like API
  - Configurable thread and interrupt affinity
  - Configurable timer period
  - Ability to attach to line-based and message-based (MSI/MSI-X) interrupts
  - Shutdown handling on Windows STOP or shutdown
  - Deterministic memory
  - Access to Windows file system and registry
  - Dynamic-link library support through RTDLLs, which can be loaded implicitly or explicitly
- Real-time Inter Process Communication between Windows user processes and real-time processes
  - Native and managed interface for 32-bit or 64-bit Windows processes
  - Objects available: events, mutexes, and semaphore
  - Data sharing through shared memory
- Windows user groups for limiting access to RTX64 features (RTX64Debuggers, RTX64Administrators, and RTX64Users)
- Tools and utilities
  - RTX64 Configuration utility – configuring the Real-time Subsystem
  - RTX64 Server console – display print messages from RTSS processes
  - SRTM – provide information on Timer latency
- Software Development Kit
  - Headers and libraries for application development (Windows 32-bit and 64-bit application support)
  - Microsoft Visual Studio 2010 Support
  - Wizard for application development
  - Microsoft Visual Studio C-Runtime support

    - Debugger that allows for:
        - launching a process for debugging in Visual Studio
        - support for the following types of breakpoints:
            - function breakpoints which cause the program to break when execution reaches a specified location within a specified function
            - file breakpoints which cause the program to break when execution reaches a specified location within a specified file
            - address breakpoints which cause the program to break when execution reaches a specified memory address
            - data breakpoints which cause the program to break when the value of a variable changes
    - Sample source to show basic concepts
- Product Documentation
    - Getting Started Guide
    - Real-time features and concepts
    - MiniTutorials
    - API Reference