MaxRT eRTOS 1.0 Runtime

USER GUIDE

IntervalZero

MaxRT eRTOS

MaxRT eRTOS 1.0 Runtime User Guide

IZ-DOC-MaxRT-eRTOS-0003

IntervalZero

200 Fifth Avenue, FL 6, STE 6020
Waltham, MA 02451
Phone: 781-996-4481
www.intervalzero.com

Contents

Overview

In this guide, we introduce key eRTOS Runtime features and show you how to configure and control Runtime components. The **eRTOS Runtime** includes a Real-time HAL (RTHAL) and a Real-time Kernel (RTKernel) that are necessary to run eRTOS executables: Real-time applications (.ertos) or Real-time DLLs (.edll).

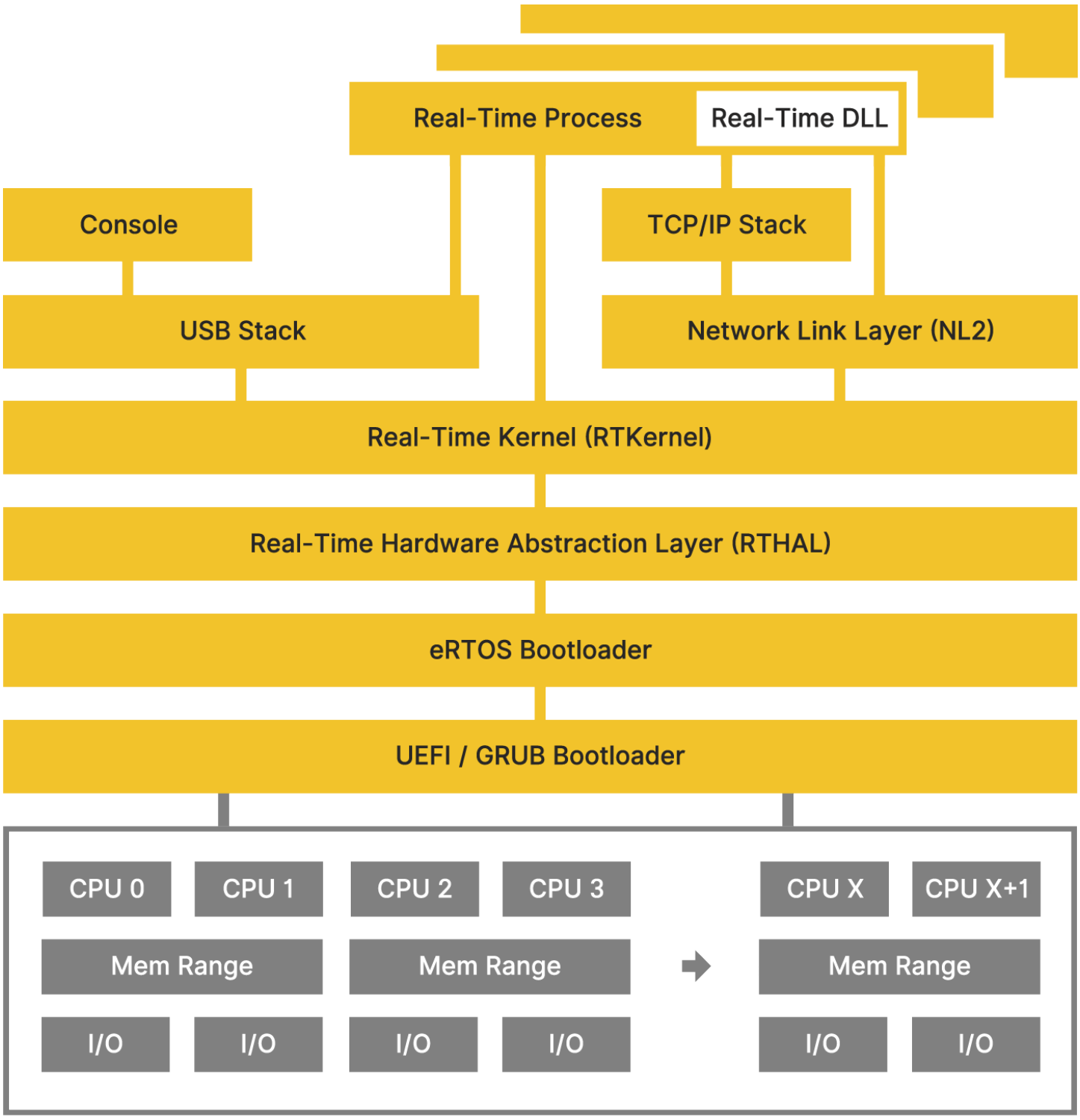
> **Note:** This guide assumes you have installed the eRTOS Runtime. For more information on installation, see the *eRTOS Runtime Install Guide*.

1

eRTOS Architecture

eRTOS components are loaded in the following order:

1. The system boots up through UEFI / GRUB Bootloader, displaying *GNU GRUB version 2.XX* on the monitor screen.
2. The Real-time Bootloader provides MaxRT eRTOS boot options.
3. The Real-time Bootloader loads the Real-time Hardware Abstract Layer (RTHAL).
4. The RTHAL initializes and loads the Real-time Kernel (RTKernel).
5. RTKernel initializes and loads the USB stack, which is an optional feature and is on by default. RTKernel starts the Network Link Layer (NL2) and TCP/IP Stack as configured in `RtKrnlConfig.ini` file.
6. The RTKernel then loads any Real-time processes (RTProcesses) as configured to startup at boot time in `RtKrnlConfig.ini` and `AutoStart.bat`.


Real-Time Process
Real-Time DLL
Console
TCP/IP Stack
USB Stack
Network Link Layer (NL2)
Real-Time Kernel (RTKernel)
Real-Time Hardware Abstraction Layer (RTHAL)
eRTOS Bootloader
UEFI / GRUB Bootloader
CPU 0
CPU 1
CPU 2
CPU 3
CPU X
CPU X+1
Mem Range
Mem Range
Mem Range
I/O
I/O
I/O
I/O
I/O
I/O

Component	Description
UEFI / GRUB Bootloader	Widely used operating system bootloader, referring to *Multiboot2 Specification version 2.0*.
eRTOS Bootloader	Reads Real-time HAL (RTHAL) configuration parameters, then loads RTHAL.
Real-Time Hardware Abstraction Layer (RTHAL)	• Processor initialization and management • Address space management • I/O device and bus management • Interrupt and exception handling • High resolution clock and timer • Power and thermal management
Real-Time Kernel (RTKernel)	• Thread, process, object, and handle management • MSpace memory management • Process/RTDLL loader • Structured exception handling • File system • Registry database • Basic C Runtime support • Console I/O logging
Network	• Network Interface Drivers • Network Link Layer (NL2) • TCP/IP protocol stack
USB stack	eRTOS Console

Note: eRTOS allocates drive letters differently than Windows.

2

Configuring eRTOS Runtime

The eRTOS Real-time HAL, Real-time Kernel, and USB Settings are configurable.

- To configure the **Real-time HAL**, edit the `grub.cfg` file at `<GRUB Drive>\boot\grub\`. See Configuring the Real-Time HAL for a full list of configuration settings.

  When the system boots into the configuration selection screen, select the appropriate IntervalZero MaxRT eRTOS entry. Once selected, you can optionally edit the entry's command by pressing the **E** key. The command contains multiple options separated by white space.

- To configure the **Real-time Kernel (RTKernel)**, edit the `RtKrnlConfig.ini` file, which is read during Real-time Kernel initialization. You can access this file at `<InstallDrive>\MaxRT\eRTOS\`. See Configuring the Real-Time Kernel for a full list of Real-time Kernel configuration settings.

- Edit the `RtConfig.reg` file to configure the **MaxRT eRTOS USB Settings** through the USB Host Stack, **NL2 settings**, and the **TCP/IP Stack settings**.

IN THIS SECTION:

eRTOS Boot Configurations

The table below lists the different eRTOS boot configurations from GRUB entries.

Boot configuration	Description
Standard Mode	The default configuration has a 100 us HAL timer period.
Fastest Mode	Fastest configuration with 20 us HAL timer period.
Tick Compensation	Adjust timer ticks based on the Time Stamp Counter (TSC).
SMI Mitigation	Disable Advanced Configuration and Power Interface (ACPI) power management.
Priority-Based CAT/MBA	Enable cache allocation and memory bus throttling based on thread priority.
Debug Mode	Using serial port to display and send shell commands.

Note: When the system boots into the Configuration Selection screen, select the appropriate IntervalZero eRTOS entry. Once selected, you can edit the entry's command by pressing **E** key. The command contains multiple options separated by white space.

Configuring the Real-Time HAL

You can configure the Real-time HAL by editing the `grub.cfg` file accessible from `<GRUB Drive>\boot\grub\` on the GRUB installed drive. The table below lists the configurable Real-time HAL settings.

Setting	Default value	Description
NumProc	64	Sets the maximum number of processors/cores (x) to run. This value must be a integer in the range 1 to 64.
TickPeriod	100	Sets HAL Timer Period to x microseconds. **Note:** Use caution when setting the HAL timer period below 20 microseconds. If the HAL timer period is set too low, it can inversely impact performance or cause your system to hang.
TickCompd	0	Select **1** to enable adjusting timer ticks based on CPU time stamp counter (TSC). Select **0** without adjustment.
AcpiPwMgr	0	Select **1** to enable SMI mitigation by disabling ACPI power management. Select **0** without disabling ACPI power management.
MultiIntLvl	1	Select **0** to use single interrupt level. Select **1** to use multiple interrupt levels. Select **2** to use two interrupt levels.

Setting	Default value	Description
DisableRdt	0	Select **1** to ignore Intel RDT capability. Select **0** to use Intel RDT capability.
PriorityCat	0	Select **1** to enable cache allocation based on thread's priority. Select **0** for flat cache allocation.
PriorityMba	0	Select **1** to enable memory bus throttling based on thread's priority. Select **0** without memory bus throttling.
RemoveMemory	0	Removes memory (x MBs) from the total available memory that RTHAL/RTKernel can use.
TruncateMemory	0xFFFFFFFFFFFFFFFF	Limits the amount of physical memory available to RTHAL/RTKernel. When you use this option, RTHAL/RTKernel ignores all memory at or above the specified physical address. Specify the address in bytes.
uart	port@0x3F8	Select **port@** for serial port debugging. Select **bdf@** for PCI serial device debugging. Select **mmio@** for MMIO device debugging.

Configuring the Real-Time Kernel

You can configure the Real-time Kernel by editing the `RtKrnlConfig.ini` file accessible from `<InstallDrive>\MaxRT\eRTOS\`.

The table below lists the configurable Kernel settings.

Setting	Type	Default value	Description
StarvationBehavior	DWORD	00000000	Sets the behavior after WatchDog time-out of a running thread. • 0 = Disabled • 1 = Freeze all active processes
StarvationTimeout	DWORD	004c4b40	Sets the starvation timeout value (in microseconds). • 0 = No time-out
SystemExtMSpacePoolCommit	DWORD	00000000	Determines whether to allocate the system process external MSpace at RTKernel start up. • 0 = No • 1 = Yes
SystemExtMSpacePoolMinThreshold	DWORD	00100000	Sets the minimum size (in bytes) of the system process external MSpace.

Setting	Type	Default value	Description
SystemIntMSpacePoolMinThreshold	DWORD	00400000	Sets the minimum size (in bytes) of the system process internal MSpace.
DisableSystemMSpacePoolExpand	DWORD	00000000	Determines whether the system process MSpaces expand when depleted. • 0 = Expandable • 1 = Not expandable
SystemMSpacePoolExpandSize	DWORD	00100000	Sets the minimum expand size (in bytes) for the system process MSpaces at each expansion.
DisableSystemMSpacePoolShrink	DWORD	00000000	Determines whether the system process MSpaces are auto shrink when memory frees. • 0 = Auto shrink • 1 = Not auto shrink
IntMSpacePoolMinThreshold	DWORD	00100000	Sets the minimum size (in bytes) of the user process internal MSpace.

Setting	Type	Default value	Description
ExtMSpacePoolMinThreshold	DWORD	00100000	Sets the minimum size (in bytes) of the user process external MSpace.
DisableMSpacePoolExpand	DWORD	00000000	Determines whether the user process MSpaces expand when depleted. • 0 = Expandable • 1 = Not expandable
MSpacePoolExpandSize	DWORD	00100000	Sets the minimum expand size (in bytes) for user process MSpaces at each expansion.
DisableMSpacePoolShrink	DWORD	00000000	Determines whether user process MSpaces auto shrink when memory frees. • 0 = Auto shrink • 1 = Not auto shrink
ZeroMemoryAtAllocation	DWORD	00000001	Determines whether to initialize memory to zero at allocation. • 0 = memory is not initialized • 1 = memory is initialized to zero

Setting	Type	Default value	Description
MSpaceFootPrintLimit	QWORD	0000000000000000	Sets a limit (in bytes) for the total amount of memory that can be allocated from an MSpace. **Note:** This is a 64-bit value. • 0 = No limit
PoolCacheTrimThreshold	DWORD	00200000	Sets the maximum amount (in bytes) of unused top-most memory to keep in Pool Cache. • ffffffff = Poll Cache trim is disabled
PoolCacheGranularity	DWORD	00010000	Sets the granularity size (in bytes) of allocations from the Pool Cache to Local Pool.
PoolCacheMmapThreshold	DWORD	ffffffff	Sets the minimum size (in bytes) allowed for allocations directly from the Local Pool. • ffffffff = Disable direct allocation

Setting	Type	Default value	Description
FreezeProcessOnMSpaceExhaust	DWORD	00000000	Determines whether to freeze the process when its non-expandable MSpace is exhausted. • 0 = Don't freeze • 1 = Freeze
DefaultTimeQuantum	DWORD	0000000000000000	Sets the default thread time quantum value (in micro-seconds). **Note:** This is a 64-bit value.
HardwareExceptionHandling	DWORD	00000000	Determines how to handle hardware exceptions. • 0 = Structure Exception Handling • 1 = Freeze the process, and send an error message to the console

Setting	Type	Default value	Description
EnterDebuggerOnException	DWORD	00000000	Determines the action following an unhandled exception. • 0 = Freeze the process, and send an error message to the console • 1 = Enter debugger
FreeStackOnTerminateThread	DWORD	00000000	Determines whether to free the thread stack after a thread is terminated. • 0 = Save the thread stack • 1 = Free the thread stack

Setting	Type	Default value	Description
StopAtFirstException	DWORD	00000000	Determines whether to stop at first exception to avoid extra exception during exception handling. **Note:** This is a bitmap of exception indexes. • 0 = Don't stop at first exception • 1 = Stop at first exception.
BehaviourStopAtFirstException	DWORD	00000000	Determines the behavior when stopping at first exception. • 0 = Freeze process and send an error message to the console • 1 = In addition to freeze process and display error message, write extra exception analysis information into log file.

Setting	Type	Default value	Description
PriorityInversion	DWORD	00000001	Enables or disables Mutex priority inversion. • 0 = Disable Mutex priority inversion • 1 = Enable Mutex priority inversion
SearchPath	String	eRTOS bin directory	Sets the search path for loading process and RTDLL files. The maximum length allowed for the search path is 266 characters. **Note:** Quotation marks inside strings are not permitted.
LogOutputOnFile	DWORD	00000001	Determines whether RtPrintf outputs to a log file (`RtLogFile.txt`). • 0 = Disable RtPrintf logging • 1 = Enable RtPrintf logging
NL2AutoStart	DWORD	00000000	Determines whether the NL2 should start automatically.

Setting	Type	Default value	Description
TCPIPAutoStart	DWORD	00000000	Determines whether the TCPIP Stack should start automatically. • 0 = Don't start the TCPIP Stack • 1 = Start the TCPIP Stack **Note:** *NL2AutoStart* needs to be enabled first if you want to start *TCPIPAutoStart*.
USBAutoStart	DWORD	00000001	Determines whether the USB Stack should be enabled. • 0 = Don't start the USB Stack • 1 = Start the USB Stack **Note:** If the USB Stack is not enabled, the eRTOS Console will not be available.

Setting	Type	Default value	Description
EnableRtssJig	DWORD	00000001	Determines whether to run the test jig at RTKernel startup. • 0 = Don't run test jig • 1 = Run test jig
TimeZone	String	EST5EDT	Sets the TimeZone environment variable to establish the relationship between local time and UTC time.
DaylightState	DWORD	00000000	Sets whether the system is operating in the STANDARD time zone or DAYLIGHT time zone. • 0 = STANDARD time zone • 1 = DAYLIGHT time zone

Configuring eRTOS Components

eRTOS Runtime includes the following optional components:

- Network Link Layer (NL2)
- TCP/IP Stack (This feature requires a separate license)
- USB Stack

You can configure eRTOS components and store custom user values for applications by editing the `RtConfig.rtreg` configuration file accessible from <InstallDrive>\MaxRT\eRTOS\. The `RtConfig.rtreg` file is written in human readable plain text, which makes it easy to read and modify.

Note: This `RtConfig.rtreg` file is read during startup of each component. If you modify the `RtConfig.rtreg` file after a component is started, the changes will not take place until you restart eRTOS.

Note: eRTOS only recognizes one `RtConfig.rtreg` file and it has a specific syntax.

Syntax

```
[SectionPath1]
“DataItemName1”=DataType1:DataValue1
“DataItemName2”=DataType2:DataValue2
[SectionPath2]
“DataItemName3”=DataType3:DataValue3
```

Parameters

`SectionPathx`

The unique path that separates each hierarchy. For example: **`HKEY_LOCAL_MACHINE\SOFTWARE\INTERVALZERO\eRTOS`**. The *SearchPath* must be surrounded by brackets ([ ]). `RtConfig.rtreg` can contain several unique paths. Each path is case sensitive and can contain one or many data items.

The following *SearchPaths* locations should not be modified since they are reserved for eRTOS components:

- `HKEY_LOCAL_MACHINE\SOFTWARE\INTERVALZERO\eRTOS` – general eRTOS settings.
- `HKEY_LOCAL_MACHINE\SOFTWARE\INTERVALZERO\eRTOS\NL2` – general NL2 settings.
- `HKEY_LOCAL_MACHINE\SOFTWARE\INTERVALZERO\eRTOS\TCPIP` – general TCP/IP Stack settings
- `HKEY_LOCAL_MACHINE\SOFTWARE\INTERVALZERO\eRTOS\RtUSB` – general USB settings
- `HKEY_LOCAL_MACHINE\SOFTWARE\INTERVALZERO\eRTOS\NL2Interfaces` - NL2 Interfaces settings. Each NL2 interface should have a subkey whose name is the name of the NL2 interface. The NL2 will load the interfaces in the order they appear in the `RtConfig.rtreg` file.

Below are examples of possible unique paths where a user could store custom data:

- `HKEY_LOCAL_MACHINE\MyData\MyApplication`
- `HKEY_USERS\MyData\MyApplication`

`DataItemNamex`

The name of a unique data item. An equal sign (=) immediately follows the name of the data item. For example, if you wanted to represent the data item **MyDword** with a value of **99** you would write *"MyDword"=dword:99*. If you wanted to represent the data item **MyQword** with a value of **9999999999** you would write *"MyQword"=qword:9999999999*. If you wanted to represent a string for a data item **MyString** that contains a string value of **This is a string!**, you would write "*MyString*"=" *This is a string!*". No data type is needed for strings. The data item name is case-sensitive.

> **Note:** You can enter several *DataItemNamex* lines for the same section path.

`DataTypex`

A keyword that specifies the type of data value assigned to a *DataItemNamex* data item. A string is the default data type, so it does not need to be provided. For all other data types, the type must follow the equal sign (=) without spaces. A colon (:) must immediately follow the data type.

Typical data types:

Data type	In RtConfig.rtreg
dword	Data represented by a number that is 4 bytes long (a 32-bit unsigned integer). The number must be written in decimal notation (not hex) and be in the range 0..4294967295.
qword	Data represented by a number that is a 64-bit unsigned integer. The number must be written in decimal notation (not hex) and be in the range 0..18446744073709551615.

`DataValuex`

Immediately follows the colon (or the equal sign with string value) and must be in the appropriate format (for example, string or decimal).

Registry Types

The `RtConfig.rtreg` file supports these registry types:

Registry type	Format example
REG_DWORD	`"MyDword"=dword:123`
REG_QWORD	`"MyQword"=qword:977348343935`
REG_SZ	`"MyString"="This is a string!"`

Configuring the Network and Network Interfaces

eRTOS provides several options for configuring the network (Network Link Layer (NL2) and optional TCP/IP Stack) and for managing network interfaces through the `RtConfig.rtreg` file.

IN THIS SECTION:

Network Interface Cards

To use a Network Interface Card (NIC) with the NL2, the associated driver must be present on the eRTOS drive and be referenced in the NL2 Interface configuration (see below). eRTOS provides several drivers that allow you to use many NICs out-of-the-box. You can also see eRTOS SDK Templates to see how to configure NICs for different divers.

Each NIC card that is an eRTOS device requires its own interrupt, whether it's line-based, MSI, or MSI-X.

Configuring and Controlling the Network

The eRTOS `RtConfig.rtreg` file provides several options for configuring network behavior and performance. This file and the `RtKrnlConfig.ini` file are available from `<InstallDrive>\MaxRT\eRTOS\`.

Configuring and Controlling the Network Link Layer (NL2)

Controlling NL2 Startup Behavior

The status of the *NL2AutoStart* parameter in the `RtKrnlConfig.ini` file determines whether the NL2 starts automatically when the kernel starts.

To start the NL2 with the kernel:

Set *NL2AutoStart* to 1 to start the NL2 automatically when the kernel starts. The default value is 0 (do not start the NL2 with the kernel).

To start the NL2 independent of the kernel:

Set *NL2AutoStart* to 0 to start the NL2 independent from the kernel. This is the default behavior.

NL2 Configuration Settings

The `RtConfig.rtreg` configuration file contains several options for configuring the Network Link Layer (NL2). You can find these by Section Path `[HKEY_LOCAL_MACHINE\SOFTWARE\INTERVALZERO\eRTOS\NL2]`

Setting	Type	Recommended value	Description
Verbose	DWORD	1	Allows the NL2 to be run in verbose mode. • 1 (Enable verbose logging) • 0 (Disable verbose logging)
IdealProcessor	DWORD	0	The processor number on which the NL2's main thread will run. **Note:** Processor numbers are zero based. Processor 0 is the first processor.
MainThreadPriority	DWORD	60	The priority of the NL2's main thread. This value must be within the range 0 to 127.

Setting	Type	Recommended value	Description
ExtMSpacePoolMinimumSize	DWORD	1048576	The minimum size of the NL2 process's external MSpace in bytes. The minimum is 65536 bytes. The best value to use depends on the number of enabled interfaces and queues.
MSpacePoolExpandable	DWORD	1	Controls whether the NL2 process will request additional memory when its MSpace is exhausted. • 1 (Expand) • 0 (Do not expand)
MSpacePoolExpandSize	DWORD	1048576	The size of memory requested if the NL2 expands its MSpace.

Configuring and Controlling the TCP/IP Stack

Note: These settings require a valid TCP/IP Stack license.

Note: eRTOS does not validate or prevent the configuration of multiple enabled interfaces with the same IP address.

Controlling TCP/IP Stack Startup Behavior

The status of the *TCPIPAutoStart* parameter in the `RtKrnlConfig.ini` file determines whether the TCP/IP Stack starts automatically.

To start the TCP/IP Stack with the NL2:

Set the *TCPIPAutoStart* setting to 1 to start the TCP/IP Stack automatically.

Note: If *TCPIPAutoStart* is enabled (*TCPIPAutoStart=1*), the *NL2AutoStart* must be set to 1 as well.

TCP/IP Stack Configuration Settings

The `RtConfig.rtreg` configuration file contains several options for configuring the TCP/IP Stack. You can find these settings under Section Path `[HKEY_LOCAL_MACHINE\SOFTWARE\INTERVALZERO\eRTOS\TCPIP]`

Setting	Type	Recommended value	Description
Memory	DWORD	4096	The amount of memory, in kilobytes, allocated to the TCP/IP stack.

Setting	Type	Recommended value	Description
MaxConcurrency	DWORD	0	The number of threads allowed to run concurrently within the TCP/IP Stack. The range is 0 to 10340. The TCP/IP Stack needs to initialize certain attributes when started to allow a certain number of threads to run and provide services for each client that requests services. For example, running a client and a server application will require 1 concurrency each. The TCP/IP Stack running by itself requires 1 concurrency to run the Loopback service. Loading and managing interfaces requires an average of 3 threads, thus a concurrency of 3 for each interface. **Note:** We recommend that this value be calculated automatically (0, default). **Note:** If the TCP/IP Stack requires more threads than the initialization process prepared for, the Stack will crash.

Setting	Type	Recommended value	Description
IdealProcessor	DWORD	0	The processor number on which the TCP/IP Stack will run. **Note:** Processor numbers are zero-based. Processor 0 is the first processor.
MaxSockets	DWORD	64	The maximum number of sockets. The TCP/IP Stack allocates actual socket memory when it creates a socket, so it must know the maximum number of sockets it must create. The specified value must be in the range of 1 to 32765. The default value is 64. **Note:** In the running system, the socket range is 0 to the maximum number of sockets. For example, if the maximum number of sockets is set to 64, the range is 0 to 63.

Setting	Type	Recommended value	Description
TickInterval	DWORD	100	The Stack timer tick interval. The Stack timer is an internal timer used for all internal synchronization. The TCP/IP Stack requires a fixed-time notification for every Stack timer interval to update its elapsed time counters. Since several protocols are implemented within the Stack, dealing with individual timers would be cumbersome. Therefore, the TCP/IP Stack is optimized to use a single notification for how much time has elapsed. The Stack timer system manages all the different timers used within the TCP/IP Stack. The specified value must be within the range of 1 to 1000 milliseconds.
TimerIdealProcessor	DWORD	0	The ideal processor for the TCP/IP Stack Timer. **Note:** Processor numbers are zero based. Processor 0 is the first processor.

Setting	Type	Recommended value	Description
TimerPriority	DWORD	66	The priority of the TCP/IP Stack's first real-time timer thread, which updates timer variables. This value must be within the range 1-127, where 1 is the lowest priority and 127 is the highest priority.
IPReassemblyTimeout	DWORD	60	The time-out interval on IP reassembly. **Note:** We recommend that you decrease the IP reassembly time-out value so that it is less than the wrap-around time in an IP ID field.

Setting	Type	Recommended value	Description
ExtMSpacePoolMinimumSize	DWORD	6422528	The minimum size of the TCP/IP process's external MSpace, in bytes. The minimum is 65536 bytes. This value needs to be large enough to support TCP/IP Stack heap allocation and must be at least the TCP/IP Stack heap allocation size plus 2176 kilobytes. You can use the following formula to determine the amount of memory you need to allocate: *Minimum size for external MSpace = StackHeap(k) + 2176k + (64k * numberOfEnabledInterfaces)*
MSpacePoolExpandable	DWORD	1	Controls whether the TCP/IP process will request additional memory when its MSpace is exhausted. • 1 (Expand) • 0 (Do not expand)
MSpacePoolExpandSize	DWORD	1048576	Sets the size of memory requested if the TCP/IP Stack expands its MSpace.

Setting	Type	Recommended value	Description
MaxArpEntries	DWORD		Sets the maximum number of ARP entries allowed by the TCP/IP Stack. Each ARP cache entry is 100 bytes. It is recommended that the maximum ARP cache entries supported be greater than the total number of devices with which the interface communicates. **Note:** If the value is too small, the ARP cache can overflow. The potential for an overflow increases when the most network devices are offline. **Note:** When an overflow occurs, the TCP/IP Stack presents the warning message *tfRtClone: ARP cache full*, which indicates that the maximum number of entries supported should be increased.

Setting	Type	Recommended value	Description
TimerExecutePriority	DWORD	60	The priority of the TCP/IP Stack's second real-time timer thread, which executes functions for expired timers. This value must be in the range 1-127, where 1 is the lowest priority and 127 is the highest priority. This value must be less than or equal to the value set for Stack timer priority.

Managing Network Interfaces

You can add, delete, set properties for, and associate filters with eRTOS network interfaces using the `RtConfig.rtreg` configuration file accessible from `<InstallDrive>\MaxRT\eRTOS\`.

Topics:

Adding, Modifying, and Deleting Interfaces

Before you can manage interfaces, you must first add the eRTOS network interface to the `RtConfig.rtreg` configuration file. See Configuring eRTOS Components for `RtConfig.rtreg` syntax. eRTOS SDK also includes templates for different divers to configure NICs.

To add an interface:

1. Open the `RtConfig.rtreg` configuration file and add the unique **SectionPath** surrounded by brackets [].

   For example:

   `[HKEY_LOCAL_MACHINE\SOFTWARE\INTERVALZERO\eRTOS\NL2Interfaces\XXX]`

   where XXX is the name you want to give to this network interface. The name cannot exceed 64 characters and cannot include spaces.

   > **Note:** The NL2 starts the network interfaces in the order they appear in the `RtConfig.rtreg` file.

2. Create the structure that will host all the NIC properties. Under the newly created section, add the following sub-sections:

   `[HKEY_LOCAL_MACHINE\SOFTWARE\INTERVALZERO\eRTOS\NL2Interfaces\XXX\RxQueues\0]`

   `[HKEY_LOCAL_MACHINE\SOFTWARE\INTERVALZERO\eRTOS\NL2Interfaces\XXX\RxQueues\1]`

   ...

[HKEY_LOCAL_MACHINE\SOFTWARE\INTERVALZERO\eRTOS\NL2Interfaces\XXX\RxQueues*NB_RX_QUEUES-1*]

[HKEY_LOCAL_MACHINE\SOFTWARE\INTERVALZERO\eRTOS\NL2Interfaces\XXX\TxQueues\0]

[HKEY_LOCAL_MACHINE\SOFTWARE\INTERVALZERO\eRTOS\NL2Interfaces\XXX\TxQueues\1]

…

[HKEY_LOCAL_MACHINE\SOFTWARE\INTERVALZERO\eRTOS\NL2Interfaces\XXX\TxQueues*NB_RX_QUEUES-1*]

[HKEY_LOCAL_MACHINE\SOFTWARE\INTERVALZERO\eRTOS\NL2Interfaces\XXX\MsixMessages\0]

[HKEY_LOCAL_MACHINE\SOFTWARE\INTERVALZERO\eRTOS\NL2Interfaces\XXX\MsixMessages\1]

…

[HKEY_LOCAL_MACHINE\SOFTWARE\INTERVALZERO\eRTOS\NL2Interfaces\XXX\MsixMessages*NB_MSIX_MESSAGES-1*]

[HKEY_LOCAL_MACHINE\SOFTWARE\INTERVALZERO\eRTOS\NL2Interfaces\XXX\Protocols\TCPIP]

Note:
- If you are unsure how many Receive Queues your NIC supports, or if you want to use one Receive Queue only, create the "RxQueues\0" sub-section (do as if *NB_RX_QUEUES=1*)
- If you are unsure how many Transmit Queues your NIC supports, or if you want to use one Transmit Queue only, create the "TxQueues\0" sub-section (do as if *NB_TX_QUEUES=1*).
- If you are unsure how many MSI-X Messages your NIC supports, or if you don't want to use MSI-X Multi vector interrupt mode, create the "MsixMessages\0" sub-section (do as if *NB_MSIX_MESSAGES=1*).

3. Create the required data items for this NIC.

   a. Add a **PciLocation** String item under the NIC's root section to specify the PCI bus Location of the NIC in the form of three semicolon-separated integers.

      For example:

      "PciLocation"="2;0;0"

You can run the PciScanBus.ertos sample binary to get location information. See the eRTOS SDK Help for information on this sample.

b. Add a **DriverPathname** String item under the NIC's root section to specify the eRTOS Driver to use for this NIC.

For example:

```
"DriverPathname"="RtndIpch.edll"
```

See eRTOS Supported NICs in the eRTOS Help for a list of NIC drivers which are available out of the box.

4. Create the optional data items for this NIC. See Network Link Layer (NL2) Properties and TCP/IP Stack Properties below for the exhaustive list of required and optional data items.

To change interface properties:

To change an interface property, follow the steps below.

1. Open the `RtConfig.rtreg` configuration file and find the interface whose properties you want to view or edit under `[HKEY_LOCAL_MACHINE\SOFTWARE\INTERVALZERO\eRTOS\NL2Interfaces\XXX]` section.
2. Edit the properties you want to change. See Network Link Layer (NL2) Properties and TCP/IP Stack Properties below.
3. You must restart the machine running eRTOS.

To delete an interface:

1. Open the `RtConfig.rtreg` configuration file and find the interface you want to delete under `[HKEY_LOCAL_MACHINE\SOFTWARE\INTERVALZERO\eRTOS\NL2Interfaces\XXX]` section.
2. Delete the section and all its sub-sections.

Network Link Layer (NL2) Interface Properties

To change NL2-specific interface properties for a specific interface, select the interface you want to edit from the `RtConfig.rtreg` configuration file. You can find these settings under the `[HKEY_LOCAL_MACHINE\SOFTWARE\INTERVALZERO\eRTOS\NL2Interfaces\XXX]` section.

eRTOS SDK contains the following NIC-specific configuration templates. You can use these templates as a starting point when configuring interface properties.

- RtConfigi210.rtreg
- RtConfigi219.rtreg
- RtConfigi225.rtreg
- RtConfigi350.rtreg
- RtConfigRealtek.rtreg
- RtConfigx550.rtreg

NL2 interface-level properties

These are the NL2 properties that apply to the whole network interface. They are all represented by data items under the `[HKEY_LOCAL_MACHINE\SOFTWARE\INTERVALZERO\eRTOS\NL2Interfaces\XXX]` section.

Setting	Type	Recom-mended value	Description
Enable	DWORD	1	Enable/disable the interface.

Setting	Type	Recom-mended value	Description
PciLocation	String	-	The PCI bus location of the network interface card for the interface in the form of three semicolon-separated integers. **Note:** You can run the `PciScanBus.ertos` sample binary to get location information. See the eRTOS SDK Help for more information.
DriverPathname	String	-	The eRTOS driver for the NIC.
JumboEnabled	DWORD	0	Enables Jumbo frames. • 1 (Enable Jumbo) • 0 (Disable Jumbo)
JumboMaxPacketSize	DWORD	1514	The maximum packet size allowed by the interface, excluding the FCS, in bytes. **Note:** This property is ignored if *JumboEnabled=0*.
AutoNegotiationDisabled	DWORD	0	Disables Auto-negotiation. • 1 (Disable auto-negotiation) • 0 (Enable auto-negotiation)

Setting	Type	Recom-mended value	Description
ForcedFullDuplex	DWORD	-	Forced duplex mode, in case *AutoNegotiationDisabled=1*. • 1 (Full-duplex) • 0 (Half-duplex)
ForcedSpeedMbps	DWORD	-	Forced speed, in Mbps, in case *AutoNegotiationDisabled=1*.
RestrictAdvertisedDuplex	DWORD	0	Advertise a subset of the supported duplex modes. • 1 (Advertise only the duplex mode selected by **AdvertisedFullDuplex**) • 0 (Advertise all supported duplex modes)
AdvertisedDuplexMode	DWORD	-	The single duplex mode to advertise in case *RestrictAdvertisedDuplex=1*. • 1 (Advertise Full-duplex only) • 0 (Advertise Half-duplex only)

Setting	Type	Recom-mended value	Description
RestrictAdvertisedSpeed	DWORD	0	Advertise a subset of the supported speeds. • 1 (Advertise only the speed selected by **AdvertisedSpeedMbps**) • 0 (Advertise all supported speeds)
AdvertisedSpeedMbps	DWORD	-	The single speed to advertise in case *RestrictAdvertisedSpeed=1*.
FlowControlSetup	DWORD	0	Use non-hardware-default settings for hardware flow control. • 1 (Use non-hardware-default settings) • 0 (Use hardware-default settings)
FlowControlFrameTransmissionDisabled	DWORD	-	Disable the transmission of Flow Control frames. • 1 (Prohibit the hardware from transmitting Flow Control frames) • 0 (Allow the hardware to transmit Flow Control frames)

Setting	Type	Recommended value	Description
FlowControlFrameReceptionDisabled	DWORD	-	Disable the processing of received Flow Control frames. • 1 (Prohibit the hardware from processing the received Flow Control frames) • 0 (Allow the hardware to process the received Flow Control frames)
FlowControlDownshiftEnabled	DWORD	-	Allow the driver to downshift the above **FlowControlFrameTransmissionDisabled** and **FlowControlFrameReceptionDisabled** settings depending on the results of auto-negotiation. • 1 (Allow downshift) • 0 (Disallow downshift)

Setting	Type	Recom-mended value	Description
HardwareTimestampingEnabled	DWORD	-	Globally enable the hardware timestamping logic in the NIC. • 1 (Enable hardware timestamping) • 0 (Disable hardware timestamping) **Note:** Hardware timestamping is not supported by all NICs.
IngressTimestampingRule	DWORD	-	Determines the type of incoming packets to timestamp, in case *HardwareTimestampingEnabled=1*. • 0 (Timestamp all incoming frames) • 1 (Timestamp incoming PTPv1 over UDP event frames) • 2 (Timestamp incoming PTPv2 over UDP event frames) • 3 (Timestamp incoming PTPv2 over ETH event frames) • 4 (Timestamp incoming PTPv2 event frames) **Note:** Not all options are supported by all NICs.

Setting	Type	Recom-mended value	Description
IngressTimestampingUdpPort	DWORD	319	The UDP port used to recognize UDP event frames.
IngressTimestampingEtherType	DWORD	35063 (0x88F7)	The **EtherType** used to recognize ETH event frames.
InterruptType	DWORD	2	The type of interrupt that will be used: • 0 (Line-based) • 1 (MSI) • 2 (MSI-X) **Note:** For improved performance, we recommend you use MSI-X when possible.
NonMsixIstIdealProcessor	DWORD	0	The ideal processor for the thread servicing the interface interrupts, in case **InterruptType** is not equal to 2 (MSI-X). This value must be a valid eRTOS processor.
NonMsixPriority	DWORD	64	The interface's interrupt priority, in case **InterruptType** is not equal to 2 (MSI-X). This value must be a valid eRTOS priority within the range 1-127.
MsixNonQueueMessageId	DWORD	-	The Message ID to use for non-queue interrupts, in case **InterruptType** is equal to 2(MSI-X).

NL2 Receive Queue-level properties

These are the NL2 properties that apply to a specific Receive Queue of the network interface. They are all represented by data items under the `[HKEY_LOCAL_MACHINE\SOFTWARE\INTERVALZERO\eRTOS\NL2Interfaces\XXX\RxQueues\N]` section, where *N* is the index of the Receive Queue to configure.

Setting	Type	Recommended value	Description
Enable	DWORD	1	Enable/disable the Receive Queue.
NbBuffers	DWORD	256	The number of NL2 Buffers that the driver must allocate for this Receive Queue at startup.
TimestampingEnabled	DWORD	-	Allows NL2 applications to request timestamping of incoming frames on this Receive Queue. • 0 (don't allow ingress timestamping on this Receive Queue) • 1 (allow ingress timestamping on this Receive Queue)

Setting	Type	Recommended value	Description
ManagementThreadIdealProcessor	DWORD	0	The processor number on which this Receive Queue's Management thread will run. **Note:** Processor numbers are zero based. Processor 0 is the first processor.
ManagementThreadPriority	DWORD	62	The priority of this Receive Queue's Management thread. This value must be within the range 0 to 127.
MsixQueueMessageId	DWORD	-	The Message ID to use for interrupts associated with this Receive Queue, in case **InterruptType** is equal to 2(MSI-X).

NL2 Transmit Queue-level properties

These are the NL2 properties that apply to a specific Transmit Queue of the network interface. They are all represented by data items under the `[HKEY_LOCAL_MACHINE\SOFTWARE\INTERVALZERO\eRTOS\NL2Interfaces\XXX\TxQueues\N` section, where *N* is the index of the Transmit Queue to configure.

Setting	Type	Recommended value	Description
Enable	DWORD	1	Enable/disable the Transmit Queue.
NbBuffers	DWORD	256	The number of NL2 Buffers the driver must allocate for this Transmit Queue at startup.

Setting	Type	Recommended value	Description
TimestampingEnabled	DWORD	-	Allows NL2 applications to request timestamping of outgoing frames on this Transmit Queue. • 0 (don't allow egress timestamping on this Transmit Queue) • 1 (allow egress timestamping on this Transmit Queue) **Note:** Most drivers don't allow egress timestamping on multiple Transmit Queues at the same time.

Setting	Type	Recommended value	Description
ManagementThreadIdealProcessor	DWORD	0	The processor number on which this Transmit Queue's Management thread will run. **Note:** Processor numbers are zero based. Processor 0 is the first processor.
ManagementThreadPriority	DWORD	62	The priority of this Transmit Queue's Management thread. This value must be within the range 0 to 127.
MsixQueueMessageId	DWORD	-	The Message ID to use for interrupts associated with this Transmit Queue, in case **InterruptType** is equal to 2(MSI-X).

NL2 MSI-X Message-level properties

These are the NL2 properties that apply to a specific MSI-X Message of the network interface. They are all represented by data items under the `[HKEY_LOCAL_MACHINE\SOFTWARE\INTERVALZERO\eRTOS\NL2Interfaces\XXX\MsixMessages\N` section, where *N* is the index of the MSI-X Message to configure.

Note: These properties apply only if **InterruptType** is set to 2(MSI-X).

Setting	Type	Recommended value	Description
Enabled	DWORD	1	Enable/disable the MSI-X Message.
IstIdealProcessor	DWORD	0	The ideal processor for the thread servicing this MSI-X interrupt. This value must be a valid eRTOS processor. **Note:** Processor numbers are zero based. Processor 0 is the first processor.
IstPriority	DWORD	64	The interrupt priority for this MSI-X Message. This value must be a valid eRTOS priority within the range 1-127.

TCP/IP Stack Properties

Note: eRTOS does not validate or prevent the configuration of multiple enabled interfaces with the same IP address.

Setting	Type	Recommended value	Description
Enabled	DWORD	1	Enable/disable TCP/IP support for this NIC.
IPAddr	String	192.168.100.50	The IPv4 address of the interface in dotted-quad notation.
Netmask	String	255.255.255.0	The IPv4 subnet mask of the interface in dotted-quad notation.
Gateway	String	0.0.0.0	The gateway for this interface in dotted-quad notation. Value 0.0.0.0 means No Gateway.
IPv6Addr	String	Auto	The IPv6 address. If “Auto”, the interface uses the Link-Local IPv6 address only.
IPv6Prefix	DWORD	64	The IPv6 network prefix for IPv6Addr.
LinkStatus	DWORD	1	Enable/disable Link Status monitoring. • 0 (disable Link Status monitoring) • 1 (Enable Link Status monitoring)

Setting	Type	Recommended value	Description
LinkStatusIdealProcessor	DWORD	0	The ideal processor for the Link Status thread. This value must be a valid eRTOS processor. **Note:** Processor numbers are zero based. Processor 0 is the first processor.
LinkStatusPriority	DWORD	0	The Link status thread priority. This value must be a valid eRTOS priority within the range 0-127
ReceiveIdealProcessor	DWORD	0	The ideal processor for the Receive thread. This value must be a valid eRTOS processor. **Note:** Processor numbers are zero based. Processor 0 is the first processor.
ReceivePriority	DWORD	0	The Receive thread priority. This value must be a valid eRTOS priority within the range 0-127.
RxQueueIndex	DWORD	0	Index of the Receive Queue that the TCP/IP Stack receives frames from.

Setting	Type	Recommended value	Description
RxBufferCount	DWORD	512	Maximum number of inflight Ethernet frames from the NIC to the TCP/IP Stack. **Note:** This is the value of the **BufferCount** parameter passed by the Stack to the **Rtnl2CreateLogicalRxQueue()** function when it creates the Logical Rx Queue for this NIC at startup.
TxQueueIndex	DWORD	0	Index of the Transmit Queue that the TCP/IP Stack transmits frames to.
TxBufferCount	DWORD	512	Maximum number of inflight Ethernet frames from the TCP/IP Stack to the NIC. Note: this is the value of the **BufferCount** parameter passed by the Stack to the **Rtnl2CreateLogicalTxQueue()** function when it creates the Logical Tx Queue for this NIC at startup.
FilterState	DWORD	0	Enable/disable an Ethernet Filter driver on this NIC. • 0 (don'use a filter driver) • 1 (use a filter driver)

Setting	Type	Recommended value	Description
Filter	String	-	Pahtname of the filter driver to use, if *FilterState=1*.

RtConfig.rtreg Example

In this example configuration, the NL2 and TCP/IP Stack support one network interface card which uses the RtndIpch driver:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\INTERVALZERO\eRTOS\NL2Interfaces\I219]
"Enabled"=dword:1
"PciLocation"="0;31;6"
"DriverPathname"="RtndIpch.edll"
"InterruptType"=dword:1
"JumboEnabled"=dword:0
"JumboMaxPacketSize"=dword:0
"AutoNegotiationDisabled"=dword:0
"ForcedFullDuplex"=dword:0
"ForcedSpeedMbps"=dword:0
"RestrictAdvertisedDuplex"=dword:0
"AdvertisedFullDuplex"=dword:0
"RestrictAdvertisedSpeed"=dword:0
"AdvertisedSpeedMbps"=dword:0
"FlowControlSetup"=dword:0
"FlowControlFrameTransmissionDisabled"=dword:0
"FlowControlFrameReceptionDisabled"=dword:0
"FlowControlDownshiftEnabled"=dword:0
"HardwareTimestampingEnanled"=dword:0
"IngressTimestampingRule"=dword:0
"IngressTimestampingUdpPort"=dword:0
"IngressTimestampingEtherType"=dword:0
"NonMsixIstIdealProcessor"=dword:0
"NonMsixIstPriority"=dword:0
"MsixNonQueueMessageId"=dword:0
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\INTERVALZERO\eRTOS\NL2Interfaces\I219\RxQueues\0]
"Enabled"=dword:1
"NbBuffers"=dword:256
"TimestampingEnabled"=dword:0
"ManagementThreadIdealProcessor"=dword:0
"ManagementThreadPriority"=dword:62
"MsixQueueMessageId"=dword:0

[HKEY_LOCAL_MACHINE\SOFTWARE\INTERVALZERO\eRTOS\NL2Interfaces\I219\TxQueues\0]
"Enabled"=dword:1
"NbBuffers"=dword:256
"TimestampingEnabled"=dword:0
"ManagementThreadIdealProcessor"=dword:0
"ManagementThreadPriority"=dword:62
"MsixQueueMessageId"=dword:0
[HKEY_LOCAL_MACHINE\SOFTWARE\INTERVALZERO\eRTOS\NL2Interfaces\I219\MsixMessages\0]
"Enabled"=dword:0
"IstIdealProcessor"=dword:0
"IstPriority"=dword:0

[HKEY_LOCAL_MACHINE\SOFTWARE\INTERVALZERO\eRTOS\NL2Interfaces\I219\Protocols\TCPIP
"Enabled"=dword:1
"IPAddr"="192.168.100.50"
"Netmask"="255.255.255.0"
"IPv6Addr"="Auto"
"IPv6Prefix"=dword:64
"LinkStatus"=dword:1
"LinkStatusIdealProcessor"=dword:0
"LinkStatusPriority"=dword:0
"ReceiveIdealProcessor"=dword:0
"ReceivePriority"=dword:63
"RxQueueIndex"=dword:0
"RxBufferCount"=dword:512
"TxQueueIndex"=dword:0
"TxBufferCOunt"=dword:0
"FilterState"=dword:0
"Filter"=""
```

Device-Specific Interface Values

This topic lists the device-specific interface defaults and value ranges for all eRTOS supported devices. This information is useful when adding or modifying interfaces in the `RtConfig.rtreg` configuration file accessible from `<InstallDrive>\MaxRT\eRTOS\`.

See eRTOS Supported NICs in the eRTOS Help for an up-to-date list of supported devices.

> **Note:** Some devices may have more restrictive limits. To see the most accurate limits, consult the appropriate specification guide for your device.

Properties:

- Maximum packet size
- Number of receive buffers
- Number of transmit buffers
- Default receive queue
- Number of receive queues
- Number of transmit queues
- Link configuration
- Interrupt type

Maximum packet size

Setting	Description	Recommended value	Type	NL2 limits
MaxPacketSize	The maximum packet size allowed by the interface.	1514	DWORD	1514-16380

Driver	Devices	Driver limits	Device limits	Notes
RtndIgc	All	1514-9212	1514-9212	
RtndIgb	All	1514-9716	1514-9716	
RtndIpch	All	1514-9014	1514-9212	
RtndI10gb	All	1514-15868	1514-15868	

Driver	Devices	Driver limits	Device limits	Notes
RtndRtl	All	1514-9212	1514-9212	Maximum packet size for specific hardware revisions:

Revision	Driver Limits	Device Limits	Tested
0x8110	1514	1514	No
0x8168C, 0x8111C, 0x8168CP, 0x8111CP	6140	6140	No
0x8168C-SPIN2, 0x8111C-SPIN2	6140	6140	Yes
0x8168B, 0x8110S, 0x8169	7436	7436	No
0x8168D, 0x8168DP, 0x8168G, 0x8168EP	9212	9212	No
0x8168EVL, 0x8111EVL, 0x8168H, 0x8111H, 0x8168E, 0x8111E, 0x8168F, 0x8111F	9212	9212	Yes

Number of receive buffers

Setting	Description	Recommended value	Type	NL2 limits
NumRecvBuffers	The number of receive buffers used by the interface. Must be greater than or equal to 64.	256	DWORD	8-16384

Driver	Devices	Driver limits	Device limits	Notes
RtndIgc	All	80-16384	8-16384	Multiple of 8, rounded up.
RtndIgb	All	80-16384	8-16384	Multiple of 8, rounded up.
RtndIpch	All	80-16384	8-16384	Multiple of 8, rounded up.
RtndI10gb	All	80-16384	8-16384	Multiple of 8, rounded up.
RtndRtl	All	64-1024	8-1024	Multiple of 8, rounded up.

Number of transmit buffers

Setting	Description	Recommended value	Type	NL2 limits
NumXmitBuffers	The number of transmit buffers used by the interface. Must be greater than or equal to 64.	256	DWORD	8-16384

Driver	Devices	Driver limits	Device limits	Notes
RtndIgc	All	80-16384	8-16384	Multiple of 8, rounded up.
RtndIgb	All	80-16384	8-16384	Multiple of 8, rounded up.
RtndIpch	All	80-16384	8-16384	Multiple of 8, rounded up.
RtndI10gb	All	80-16384	8-16384	Multiple of 8, rounded up.
RtndRtl	All	64-1024	8-1024	Multiple of 8, rounded up.

Default receive queue

Setting	Description	Recommended value	Type	NL2 limits
DefaultQueue	The number of the queue that receives unfiltered data, starting with queue 0.	0	DWORD	0 (Number of receive queues -1)

Driver	Devices	Driver limits	Device limits
RtndIgc	All	0 (Number of receive queues -1)	0 (Number of receive queues -1)
RtndIgb	All	0 (Number of receive queues -1)	0 (Number of receive queues -1)
RtndIpch	All	0	0
RtndI10gb	All	0	0
RtndRtl	All	0	0

Number of receive queues

Setting	Description	Default	Type	NL2 limits
MaxNumberOfRecvQueues	The number of receive queues to configure for the interface.	1	DWORD	1-128

Driver	Devices	Driver limits	Device limits
RtndIgc	All	1-1	1-4
RtndIgb	0x10C9, 0x10E6, 0x10E7, 0x10A7, 0x1526	1-16	1-16
RtndIgb	0x1521,0x150E, 0x150F, 0x1510, 0x1511, 0x1516	1-8	1-8
RtndIgb	0x1533, 0x1534, 0x1535, 0x157B	1-4	1-4
RtndIgb	0x1539	1-2	1-2
RtndIpch	All	1-1	1-2
RtndI10gb	All	1-16	1-128
RtndRtl	All	1-1	1-1

Number of transmit queues

Setting	Description	Recommended value	Type	NL2 limits
MaxNumberOfXmitQueues	The number of transmit queues to configure for the interface.	1	DWORD	1-128

Driver	Devices	Driver limits	Device limits
RtndIgc	All	1-1	1-4
RtndIgb	0x10C9, 0x10E6, 0x10E7, 0x10A7, 0x1526	1-16	1-16
RtndIgb	0x1521,0x150E, 0x150F, 0x1510, 0x1511, 0x1516	1-8	1-8

Driver	Devices	Driver limits	Device limits
RtndIgb	0x1533, 0x1534, 0x1535, 0x157B	1-4	1-4
RtndIgb	0x1539	1-2	1-2
RtndIpch	All	1-2	1-2
RtndI10gb	All	1-16	1-128
RtndRtl	All	1-1	1-1

Link configuration

Setting	Description	Recommended value	Type	NL2 limits
LinkStatus	The method used to establish the Ethernet link.	1 (Auto Negotiate)	DWORD	Auto Negotiate, 10 Mbps Half Duplex , 10 Mbps Full Duplex, 100 Mbps Half Duplex, 100 Mbps Full Duplex

Driver	Devices	Driver limits	Device limits
Rtndlgc	All	Auto Negotiate, 10 Mbps Half Duplex , 10 Mbps Full Duplex, 100 Mbps Half Duplex, 100 Mbps Full Duplex	Auto Negotiate, 10 Mbps Half Duplex , 10 Mbps Full Duplex, 100 Mbps Half Duplex, 100 Mbps Full Duplex
Rtndlgb	All	Auto Negotiate, 10 Mbps Half Duplex , 10 Mbps Full Duplex, 100 Mbps Half Duplex, 100 Mbps Full Duplex	Auto Negotiate, 10 Mbps Half Duplex , 10 Mbps Full Duplex, 100 Mbps Half Duplex, 100 Mbps Full Duplex
Rtndlpch	All	Auto Negotiate, 10 Mbps Half Duplex , 10 Mbps Full Duplex, 100 Mbps Half Duplex, 100 Mbps Full Duplex	Auto Negotiate, 10 Mbps Half Duplex , 10 Mbps Full Duplex, 100 Mbps Half Duplex, 100 Mbps Full Duplex

Driver	Devices	Driver limits	Device limits
RtndI10gb	All	Auto-Negotiate	Auto Negotiate, 10 Mbps Half Duplex , 10 Mbps Full Duplex, 100 Mbps Half Duplex, 100 Mbps Full Duplex
RtndRtl	All	Auto-Negotiate	Auto Negotiate, 10 Mbps Half Duplex , 10 Mbps Full Duplex, 100 Mbps Half Duplex, 100 Mbps Full Duplex

Interrupt type

Setting	Description	Recommended value	Type	NL2 limits
InterruptType	The type of interrupt used by an eRTOS network interface.	2 (MSI-X Single Vector when available, otherwise MSI)	DWORD	Line-Based, MSI, MSI-X Single Vector, MSI-X Multi Vector

Driver	Devices	Driver limits	Device limits
RtndIgc	All	Line-Based, MSI, MSI-X Single Vector, MSI-X Multi Vector	Line-Based, MSI, MSI-X Single Vector, MSI-X Multi Vector
RtndIgb	0x1520, 0x1521, 0x1522, 0x1523, 0x1524, 0x152F	Line-Based, MSI, MSI-X Single Vector, MSI-X Multi Vector	Line-Based, MSI, MSI-X Single Vector, MSI-X Multi Vector
RtndIgb	All others	Line-Based, MSI, MSI-X Single Vector	Line-Based, MSI, MSI-X Single Vector, MSI-X Multi Vector
RtndIpch	All	Line-Based, MSI, MSI-X Single Vector	Line-Based, MSI, MSI-X Single Vector, MSI-X Multi Vector
RtndI10gb	All	Line-Based, MSI, MSI-X Single Vector	Line-Based, MSI, MSI-X Single Vector, MSI-X Multi Vector
RtndRtl	All	Line-Based, MSI, MSI-X Single Vector	Line-Based, MSI, MSI-X Single Vector, MSI-X Multi Vector

Configuring Jumbo Frames

To send/receive Jumbo packets on a network interface, you need to enable Jumbo mode and to select the maximum allowed size for those Jumbo packets. Maximum Packet Size includes 14 bytes of Ethernet header. It does not include FCS. The IP MTU must be configured to be less than or equal to (Maximum Packet Size – 14).

The TCP/IP Stack supports Path MTU Discovery. eRTOS devices will successfully communicate over TCP with another device that supports Path MTU Discovery, even if MTUs are configured to different sizes.

> **Note:** All tested drivers support Jumbo Packets. To view a list of supported network cards, see Supported NICs.

Using an Ethernet Filter

The eRTOS Ethernet Filter is an extensible interface between the TCP/IP stack and a network interface. You can use this interface to filter all frames at the data link layer and send Ethernet frames directly to the RTND driver.

The filter layer is created as a Real-Time DLL (RTDLL). The stack loads the filter, which is associated with a given interface. Since it is an RTDLL, it can use all supported RTAPI calls.

Loading the Ethernet Filter

The Ethernet Filter is loaded when the TCP/IP stack loads. Once you create a custom filter, you can associate it with an interface and configure it to be loaded by the stack using the RtConfig.rtreg configuration file.

Tips, Tricks and Configuration Hints

This topic contains helpful tips, tricks, and configuration recommendations for the network (Network Link Layer (NL2) and TCP/IP Stack).

Potential Causes of Driver Sub-optimal Performances

If the Interrupt Thread Priority for a driver is configured to a value that is less than the priority of the application thread that sends data, it can result in a sub-optimal performances of the device interface.

Configuring eRTOS USB

eRTOS provides USB capabilities through a USB Host Stack, which can be optionally loaded when the eRTOS system boots.

> **Note:** If the eRTOS USB is not enabled, you will not be able to use the eRTOS Console.

You can configure eRTOS USB through the `RtConfig.rtreg` file under Section Path `[HKEY_LOCAL_MACHINE\SOFTWARE\INTERVALZERO\eRTOS\RtUSB]`

Setting	Type	Default value	Description
IdealProcessor	DWORD	0	DWORD Sets the processor on which the USB Stack will run.

Setting Time Zone and Daylight Time

eRTOS uses the **TimeZone** environment variable to establish the relationship between local time (target time) and the system time (UTC).

The **TimeZone** environment variable is configurable via the RtKrnlConfig.ini file and is taken into RT Kernel at eRTOS start-up. In `RtKrnlConfig.ini` file, the **TimeZone** line configures the **TimeZone** environment variable. The line consists of two components: the name, which is "TimeZone", and the value, which is a string. The default string value is set to "**EST5EDT**".

Use the following syntax to configure the **TimeZone** line:

Syntax

```
TimeZone = "tzn[+|-]hh[:mm[:ss]][dzn]"
```

Parameters

`tzn`
: Three-letter time-zone name, such as EST. You must specify the correct offset from local time to UTC.

`hh`
: Difference in hours between UTC and local time. Sign (+) optional for positive values.

`mm (Optional)`
: Minutes. Separated from *hh* by a colon (:).

`ss (Optional)`
: Seconds. Separated from *mm* by a colon (:).

`dzn`
: Three-letter daylight-saving-time zone such as EDT. If daylight saving time is never in effect in the locality, set **TimeZone** without a value for *dzn*. eRTOS assumes adherence to the United States' regulations for implementing daylight saving time (DST) calculations.

For example, the default "**EST5EDT**" uses EST to denote Eastern standard time zone. It assumes that UTC is five hours ahead of Eastern standard time, and that the United States adheres to daylight-saving time.

When the current time zone observes daylight-saving time (i.e., **TimeZone** contains three-letter of daylight-saving-time zone name, *dzn*), you need to specify whether the system is operating in the range of standard time or daylight time when eRTOS starts. eRTOS uses a Daylight state variable to take your specification. In the `RtKrnlConfig.ini` file, the **DaylightState** line configures the Daylight state variable. In that line, the name is **DaylightState**, and the value is a dword (0 for standard time, 1 for daylight time). The default value is 0.

eRTOS does not support automatic transition from standard time to daylight time, nor from daylight to standard time. However, you can manually trigger these transitions with the **setdst** shell command, or by calling **SetDaylightState** RTAPI at runtime.

3

Using eRTOS Runtime

eRTOS Runtime includes an eRTOS Application batch file (`AutoStart.bat`) and a keyboard-based console that can be used to run programming samples and real-time applications, as well as run simple commands to query information about the system.

TOPICS:

About the Batch File

The eRTOS Runtime includes an eRTOS Application batch file, `AutoStart.bat`, which is located in `<InstallDrive>\MaxRT\eRTOS\`. This file contains a list of Run commands for all program samples included with eRTOS Runtime. It can also be used to run real-time applications.

Using eRTOS Console

The eRTOS Runtime includes eRTOS Console, a console that supports USB keyboard input. eRTOS Console supports the following commands (case-insensitive). You can find full usage information for each of these commands in the eRTOS Help:

Command	Description
`Run <app_path>`	Runs the eRTOS application found at the given path.
`Kill`	When no parameter is given, this command lists the currently running eRTOS processes.

Command	Description
`Kill <PID>`	When a valid Process ID is given, that process is terminated.
`dir`	Lists all files and folders found in the current directory.
`DisplayVolumes`	Lists all of the FAT32 volumes on the system found by eRTOS, as well as what type of media they are on (SATA or USB).
`cd <folder_path>`	Changes the current directory of the console to the given path.
`cls`	Clears the console of all output history.
`getdst`	Retrieves whether the system is operating in the STANDARD time zone or DAYLIGHT time zone.
`setdst`	Sets the current time zone to STANDARD time zone or DAYLIGHT time zone. • 0 = STANDARD time zone • 1 = DAYLIGHT time zone

eRTOS Console also supports special keys that provide additional functionality to the console:

Key(s)	Description
Page Up/Down	Scrolls up and down through the console's output history.
Up/Down Arrow	Cycles through the history of previously entered commands.
Escape	Clears any input currently entered into the console.

Running Program Samples

eRTOS includes several program samples to help developers create real-time programs that will run in the eRTOS Runtime environment. eRTOS Runtime provides a set of .ertos executables in the `<InstallDrive>\MaxRT\eRTOS\bin` directory.

To run samples automatically after the Real-time Kernel completes initialization, the `AutoStart.bat` file needs to be edited. It is deployed in the `<InstallDrive>\MaxRT\eRTOS\` folder.

By default, AutoStart.bat will only run two instances of SRTM.ertos on the 1st and 2nd cores.

To configure RTKernel to run real-time applications:

1. Navigate to `<InstallDrive>\MaxRT\eRTOS\`.
2. Right-click **Edit** to modify the **RtKrnlConfig.ini** file.
3. Set the **EnableRtssJig** value to the following:
   `EnableRtssJig = 00000001 ;dword`
4. Reboot the eRTOS system.

To run eRTOS program samples:

1. Navigate to `<InstallDrive>\MaxRT\eRTOS\`.
2. Right-click **Edit** to modify the **RtKrnlConfig.ini** file.
3. Observe the Run commands for all samples included with eRTOS Runtime.

Note: For more information on Run, see the eRTOS SDK Help.

4. Find the Run command(s) for the desired sample(s).
5. Remove the comment characters (: :) to enable the Run command(s).
6. Reboot the system from a GRUB bootable USB drive or hard drive.
7. Select the desired GRUB boot configuration. See GRUB Boot Configurations for more information.
8. Upon system boot, the sample(s) will run automatically after the eRTOS Kernel startup.
9. Sample output will be displayed on the screen when the program ends.
10. Re-boot the system in its Windows Boot Configuration.
11. Navigate to `<InstallDrive>\MaxRT\eRTOS\`.
12. Open the RtLogFile.txt log file to view sample output.

Running User-Developed Real-Time Applications

You can use the eRTOS Application batch file, `AutoStart.bat`, or eRTOS Console to run real-time applications.

To configure the Real-time Kernel for running real-time applications:

1. Navigate to `<InstallDrive>\MaxRT\eRTOS\`.
2. Right-click **Edit** to modify the **RtKrnlConfig.ini** file.
3. Set the **EnableRtssJig** value to the following:
   `EnableRtssJig = 00000001 ;dword`

To run real-time applications:

1. Navigate to `<InstallDrive>\MaxRT\eRTOS\`.
2. Right-click `AutoStart.bat` and select **Edit**.
3. Add an Run command pointing to the desired application.
4. Reboot the system from a GRUB bootable USB drive or hard drive.
5. Select the desired GRUB boot configuration. See GRUB Boot Configurations for more information.
6. Upon system boot, the application(s) will run automatically after the Real-time Kernel start-up.
7. Application output will be displayed on the screen when the program ends.
8. Reboot the system in its Windows Boot Configuration.
9. Navigate to `<InstallDrive>\MaxRT\eRTOS\`.
10. Open the RtLogFile.txt log file to view application output.

To run applications using eRTOS Console:

1. Boot the system from a GRUB bootable USB drive or hard drive.
2. Select the desired GRUB boot configuration. See GRUB Boot Configurations for more information.
3. Once eRTOS has started, and the Console has initialized (indicated by a "`C:\>`" near the bottom of the screen), type `Run` followed by the path for the application to run.
4. Press **Enter**.

5. Output for the application will be displayed on the screen.
6. Output which exceeds the length of the screen can be inspected by either opening the log file in Windows as described above, or by scrolling through the output history with the Page Up and Page Down keys.

Support

For help with eRTOS, contact IntervalZero Technical Support by phone or access the online support resources available at https://www.intervalzero.com/en-support/en-customer-service/

Contacting Technical Support by Phone

> **Note:** If you purchased an IntervalZero product through a third-party reseller, please contact the reseller for support.

Location	Number	Hours
United States	1-781-996-4481 At the prompt, press 3 for Support.	Monday - Friday, 8:30 a.m. – 5:30 p.m. US Eastern Time (GMT-500), excluding holidays.
R.O.C. Taiwan	+ 886-2-2556-8117	Monday - Friday, 9:00 a.m. – 5:00 p.m. Taipei Standard Time (GMT+8), excluding holidays.

Before Calling Technical Support

Please have this information ready when you contact IntervalZero Technical Support:

- Your Support ID

  Customers who purchase direct support receive an e-mail address and password for accessing the IntervalZero Customer Support Portal.

- Your eRTOS version number

> **Note:** You must have a valid maintenance contract to receive product support.

Online Resources

Visit https://www.intervalzero.com/en-support/en-customer-service/ to log in to the Customer Support Portal (requires valid credentials), access online product Help, and view Support and Lifecycle policies and Product Release Notices.

Index