

IntervalZero

RTX 2011 Runtime for Embedded

Frequently Asked Questions

This document provides answers to questions frequently asked about the RTX 2011 Runtime for Embedded environment.

How do I create an RTX embedded target image?

After installing Microsoft Windows Embedded Studio and RTX 2011 Runtime for Embedded, use the Microsoft Component Database Manager to import the System Level Definition (.SLD) files found in the RTX installation directory. The next time you open the Microsoft Windows Embedded Studio Target Designer, the RTX components will appear in the components list.

How can I change RTX Settings?

During the creation of your image, RTX settings can be changed through the RTX Kernel Core component's Settings. If the RTX Control panel is added to the embedded image, you can use that to change subsystem configurations after installation.

How do I change the RT-TCP/IP Stack settings?

The RT-TCP/IP Stack is configured through the RtxTcplp.ini file located in the SystemRoot directory of your target system. This file is generated during the First Boot Agent, based on information entered on the RT-TCP/IP Runtime component's Settings page. This page allows you to configure all the standard fields of the INI file. See "How do I add custom fields in my RtxTcplp.ini file?" for information on custom fields.

How do I ensure that I have the correct HAL for my target system?

After the RTX Kernel Core has been selected in the Microsoft Target Designer, highlight RTX Kernel Core settings to show and select the General HAL Type. Ensure that the HAL type selected matches the HAL type of your target system. If they don't match, the embedded system will not successfully complete the first boot agent (FBA). Supported HAL types are listed in the *Installation Guide for RTX 2011 Runtime for Embedded*.

How do I convert a device to RTX during image creation?

The need to convert devices is the result of providing Plug and Play (PnP) and Power Management support within RTX. If you plan to use a PnP device within an RTX application, that device must be converted to RTX. With an embedded image, the best time to convert a device to RTX is during the running of the First Boot Agent (FBA). To convert a device during FBA you must use an INF file to install your device.

Steps for creating INF support and adding them to your target image:

1. Make a copy of one of the `.inf` files that comes with RTX Runtime for Embedded (for example, `RtE1000.inf`).
2. Rename and edit the file by replacing the hardware IDs and description names to match your device. If the device is *not* a Network adapter, you must change all references of `RtxPnPNet` to `RtxPnp`.
3. Open **Component Designer**.
4. From the *File* menu, select **Import**.
5. From the *Files of type* menu, Select **Setup information files (*.inf)**, browse to the INF file that you created in steps 1 and 2, and click **Open**. An *INF Processing Options* dialog box appears
6. Select Automatic and click **OK**.
7. In the *Import File* dialog box that appears, click **Start**. This may take a couple of minutes.
8. Once Status says Completed processing file [import succeeded], click **Close**.

You should now have a new SLD with the same name as your imported INF file and a component for every different device type. Under the Resources section of each component, you will have a Service Data resource referring to the RTX Plug and Play driver and PnPID resources for each different hardware ID that was supported for this device type in your INF file.

If you want the INF file to be added to the target, to support conversion of new devices added after the FBA is run, you will need to create a repository that contains the INF and add the repository to the SLD and the file to the component.

9. Release all the components in the SLD and close **Component Designer**.
10. Open the **Component Database Manager**.
11. On the *Database* tab, click **Import**.
12. In the *Import SLD* dialog box, browse to your SLD and click **Import**.
13. Once the SLD has been imported, in the *Import SLD* dialog box, click **Close** and then, in the *Component Database Manager* window, click **Close**.
14. Open **Target Designer** and select your component, usually located in **Hardware > Others**.

What if the RTX PCI device has IRQ conflicts with other devices?

If the RTX Properties control panel component is included, you can use it to modify device resources. If the control panel is not included, there is a command line utility called **RTSSModify** that is included with the RTX Runtime component to display all RTX devices as well as change an RTX device's IRQ or shareable setting. To use RTSSModify, you must have permission to edit the Windows registry (HKEY_LOCAL_MACHINE\System\CurrentControlSet\Enum). The default for this area is **Read only**.

If you do not want to include the RTX Runtime component, you can edit registry keys directly. Go to the enumerated instance of your device in the registry. Enumerated PCI devices are located under:

```
HKLM\System\CurrentControlSet\Enum\PCI
```

Find your device and open the **Device Parameters** key. Edit the **IRQNumberPending** value to an available IRQ. You will most likely have to change your permissions for the Device Parameters key. To do so, select the **Device Parameters** key, then from the *Security* menu, choose **Permissions** and give yourself full control). You must reboot the system for the IRQ changes to take effect.

Why does each supported RT-TCP/IP NIC card require two components?

Each RT-TCP/IP Supported NIC card requires two components to be added to the target image, the RT-TCP/IP NIC Driver and RT-TCP/IP NIC Card Support (where the NIC is a given card type). The RT-TCP/IP NIC Driver component provides and registers the RTDLL driver for a given card, whereas the RT-TCP/IP NIC Card Support component converts the NIC to RTX.

The RT-TCP/IP NIC Card Support components are provided so you do not have to create custom components to support default cards. However, there may be cases where the PNPIDs supplied in the RT-TCP/IP NIC Card Support components are too general. This may cause problems when trying to include multiple NIC cards of the same type, where not all will be controlled by RTX. In cases where the RT-TCP/IP NIC Card Support components are too general, you can include the RT-TCP/IP NIC Driver component, but then create a custom component to convert only specific devices to RTX (see [How do I convert a device to RTX during image creation?](#)).

How do I add custom fields in my RtxTcplp.ini file?

The `RtxTcpIp.ini` file used by the RtxTcplp Protocol Stack is generated during the First Boot Agent (FBA) from information stored in the registry. This allows for target specific information to be entered in the RT-TCP/IP Runtime component's Settings page, rather than editing the `RtxTcplp.ini` file for each individual target once they are up and running (although this may still be required).

To add custom fields not included in the RT-TCP/IP Runtime component's Settings page, you can add a registry value to your component that will be incorporated into the `RtxTcplp.ini` file. The registry value name should be the field name and the data should be of type REG_SZ and the value of the custom field. For example, a registry value of MemoryInk with a value of 256 will generate the line `MemoryInk=256` within the `RtxTcplp.ini`.

Section Name	Registry Location
[TCP/IP]	HKLM\System\CurrentControlSet\Control\RT_TCP/IP
[RTND0]	HKLM\System\CurrentControlSet\Control\RT_TCP/IP\RTND0
[RTND1]	HKLM\System\CurrentControlSet\Control\RT_TCP/IP\RTND1
[RTND2]	HKLM\System\CurrentControlSet\Control\RT_TCP/IP\RTND2
[RTND3]	HKLM\System\CurrentControlSet\Control\RT_TCP/IP\RTND3
[DNS]	HKLM\System\CurrentControlSet\Control\RT_TCP/IP\DNS

Is Windows Embedded Standard 7 Supported?

Yes. RTX can run in a Windows Embedded Standard 7 environment. However, Windows Embedded Standard 7 does not use SLD files. For information on including the RTX Silent Installer as a component to a Windows Embedded Standard 7 distribution share using the Image Configuration Editor (ICE) see the RTX MiniTutorial *Including RTX Runtime with Windows Embedded Standard 7*.