

ETND – ETNE TECHNICAL NOTE

ETND – ETNE are processes which allow multiple connections on a single hardware. This is useful, when it is necessary to spy device activity while custom application is running.
ETND – ETNE use TCP-IP connection. This thus allows connecting to hardware on a host present on network.

There are 4 ways to use ETND – ETNE processes:


1. ETND METHOD

With this method, customer application AND spying process (ComET for example), both connect to hardware via ETND process.

Advantage: Customer application AND spying process use the same url.

Disadvantage: Customer application will be slowed down due to TCP-IP communication overhead.

Use:

- Define a port.properties file which define the wanted url. This file MUST be in the same directory as ETND.
- Run ETND
- Use following url in the customer application:
`dsa_open_u (drv, "etb:etn://<ip>:<accept port>:<axis>");`
 where:
 - <ip>: is the ip of the host where ETND is running
 - <accept port>: is the port defined in port.properties corresponding block
- Run customer application
- Run ComET, display the list of available host using icon , select the desired host. After that, ComET will display the list of host's available connections. Select the wanted connection. ComET is now connected to the hardware.

Example:


The customer application will connect to axis 0 of the UltimET PCI, plugged in the PC running the customer application.

- Create a port.properties file in the same directory as the ETN40D process, with following content:

```
#number of ports defined
port.number=1

#parameters of port 0
port.0.name=UltimET
port.0.driver=ULTIMET
port.0.protocol=ETB
port.0.accept.0.driver=ETN:1140
port.0.start=auto
```

- run ETN40D present in the same directory as port.properties
- Use following url in the customer application:
`dsa_open_u (drv, "etb:etn://localhost:1140:0);`
- Run customer application


- Run ComET, display the list of available host using icon , select ETND of corresponding IP. In the available connections of the corresponding IP, you will find UltimET. Select it.

The customer application will connect to axis 1 of the UltimET PCI, plugged in a host PC with IP 172.22.10.20

- On the host PC, create a port.properties file in the same directory as the ETN40D process, with following content:

```
#number of ports defined
port.number=1

#parameters of port 0
port.0.name=UltimET
port.0.driver=ULTIMET
port.0.protocol=ETB
port.0.accept.0.driver=ETN:1140
port.0.start=auto
```

- on the host, run ETN40D present in the same directory as port.properties
- Use following url in the customer application:
dsa_open_u (drv, "etb:etn://172.22.10.20:1140:0);
- Run customer application on any PC present on the network
dsa_open_u (drv, "etb:etn://172.22.10.20:1140:0);
- Run ComET on any PC present on the network, display the list of available host using icon , select ETND of corresponding IP. In the available connections of the corresponding IP, you will find UltimET. Select it.

2. ETNE METHOD 1

With this method, customer application connect to hardware directly. The hardware must thus be plugged into the customer application PC.

Spying process (ComET for example), connects to hardware via ETND process.


Port.properties file contains the customer application used port.

Advantage: Customer application use direct connection, and thus, is not slowed down because of the TCP-IP communication overhead. The only overhead is due to the spying process communication.

Disadvantage: Customer application must be started before spying process.

Use:

- Use following url in the customer application:
dsa_open_u (drv, "etb:<url>:<axis>);
where:
 <url>: is the url used for the direct connection
- Allow host connection in the customer application, by adding the following line:
etne_start_custom (etb, <accept_port>, "Customer application");
where:
 etb is the communication bus got using dsa_get_etb_bus function.
 <accept_port>: is the port which will be used by ETND
- Run customer application.
- Define a port.properties file which define the wanted url. This file MUST be in the same directory as ETND.
- run ETN40D.

- Run ComET, display the list of available host using icon , select the desired host. After that, ComET will display the list of host's available connections. Select the wanted connection. ComET is now connected to the hardware.


Example:

The customer application will connect to axis 0 of the UltimET PCI, plugged in the PC running the customer application. The used accept port is the port 1140.

- Use following url in the customer application:
dsa_open_u (drv, "etb:ULTIMET:0);
- Allow host connection in the customer application, by adding the following line:
etne_start_custom (etb, 1140, "Customer application");
- Create a port.properties file in the same directory as the ETN40D process, with following content:

```
#number of ports defined
port.number=1

#parameters of port 0
port.0.name=UltimET
port.0.driver=ULTIMET
port.0.protocol=ETB
port.0.accept.0.driver=ETN:1140
port.0.start=auto
```

- run ETN40D present in the same directory as port.properties
- Run ComET, display the list of available host using icon , select ETND of corresponding IP. In the available connections of the corresponding IP, you will find UltimET. Select it.

3. ETNE METHOD 2

With this method, customer application connects to hardware directly. The hardware must thus be plugged into the customer application PC.

Spying process (ComET for example), connects to hardware via ETND process.


Port.properties file does not contain the customer application used port.

Advantage: Customer application use direct connection, and thus, is not slowed down because of the TCP-IP communication overhead. The only overhead is due to the spying process communication.

Disadvantage: None

Use:

- Use following url in the customer application:
dsa_open_u (drv, "etb:<url>:<axis>);
where:
 <url>: is the url used for the direct connection
- Allow host connection in the customer application, by adding the following line:
etne_start_custom (etb, <accept_port>, "Customer application");
where:
 etb is the communication bus got using dsa_get_etb_bus function.
 <accept_port>: is the port which will be used by ETND
- In the port.properties file present in the ETND directory, delete any block containing the url and/or the accept port.
- run ETN40D and customer application in any order.


- run ComET, display the list of available host using icon , select the desired host. After that, ComET will display the list of host's available connections. Select the wanted connection. ComET is now connected to the hardware.

Example:

The customer application will connect to axis 0 of the UltimET PCI, plugged in the PC running the customer application. The used port is the port 1140.

- Use following url in the customer application:
dsa_open_u (drv, "etb:ULTIMET:0);
- Allow host connection in the customer application, by adding the following line:
etne_start_custom (etb, 1140, "Customer application");
- Create a port.properties file in the same directory as the ETN40D process. This file should not contain the following block:

```
#parameters of port <n>
port.<n>.name=UltimET
port. <n>driver=ULTIMET
port. <n>protocol=ETB
port. <n>accept.0.driver=ETN:1140
port. <n>start=auto
```

- run ETN40D and customer application in any order.
- Run ComET, display the list of available host using icon , select ETND of corresponding IP. In the available connections of the corresponding IP, you will find UltimET. Select it.
-

4. ETNE METHOD 3

With this method, customer application connects to hardware directly. The hardware must thus be plugged into the customer application PC. The customer application starts an ETNE thread which will wait for TCP/IP connections.

Spying process (ComET for example), connects to hardware via the ETNE thread started by the customer application.

Port.properties file contains the customer application used port as accept port.

Advantage: Customer application use direct connection, and thus, is not slowed down because of the TCP-IP communication overhead. The only overhead is due to the spying process communication.

Disadvantage: ComET host list won't display the customer application host

Use:

- Use following url in the customer application:
dsa_open_u (drv, "etb:<url>:<axis>);
where:
 <url>: is the url used for the direct connection
- Allow host connection in the customer application, by adding the following line:
etne_start_custom (etb, <accept_port>, "Customer application");
where:
 etb is the communication bus got using dsa_get_etb_bus function.
 <accept_port>: is the port which will be used by ETND
- Define a port.properties file which define the wanted url as accept connection. This file MUST be in the same directory as COMET.
- run customer application.
- run ComET. In the available connections, you will find the accept connection added in port.properties. Select it. ComET is now connected to the hardware.

Example:

The customer application will connect to axis 0 of the UltimET PCI, plugged in the PC running the customer application. The used port is the port 1140.

- Use following url in the customer application:
dsa_open_u (drv, "etb:ULTIMET:0);
- Allow host connection in the customer application, by adding the following line:
etne_start_custom (etb, 1140, "Customer application");
- Create a port.properties file in the same directory as the ETN40D process. This file should contain the following block:

```
#parameters of port <n>
port.<n>.name="Customer application"
port. <n>driver=ETN:<localhost>:1140
port. <n>protocol=ETB
port. <n>accept.0.driver=ETN:1150
port. <n>start=auto
```

- run ComET, display the list of available connections. You will find a connection name "Customer application". Select it.

An example is provided with EDI package under <EDI-FOLDER>\samples\v40\c\etne

5. ETNE METHOD 3 BETWEEN RTX AND WINDOWS APPLICATION

With RTX application connected with UltimET Light PCI/PCIe, the PCI/PCIe board must be dedicated to RTX. Therefore, it is no more possible to communicate with it with a Windows application like ComET.

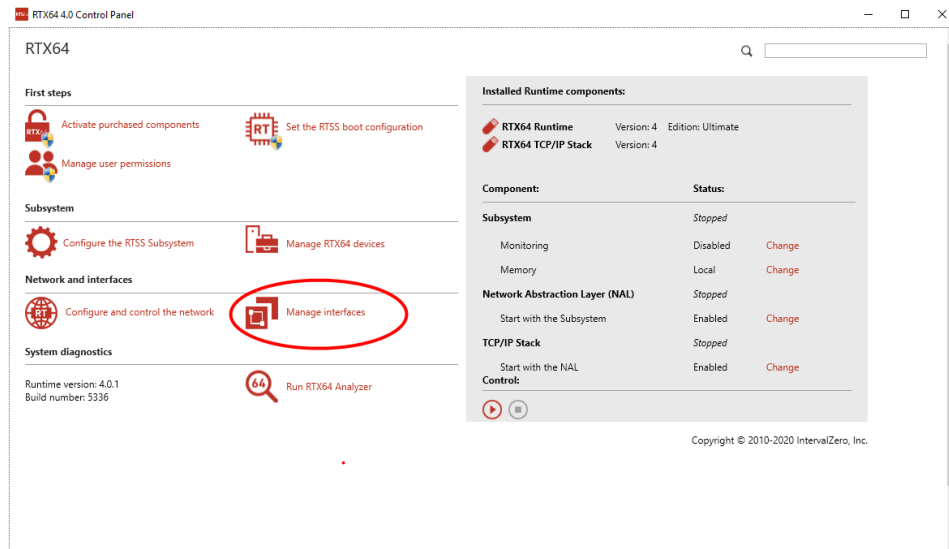
Using ETNE method 3 and RTX virtual network, a windows application will be able to communicate to your hardware.

With this method, the RTX customer application connects to hardware directly. The hardware is therefore dedicated to RTX. The RTX customer application starts an ETNE thread which will wait for TCP/IP connections.

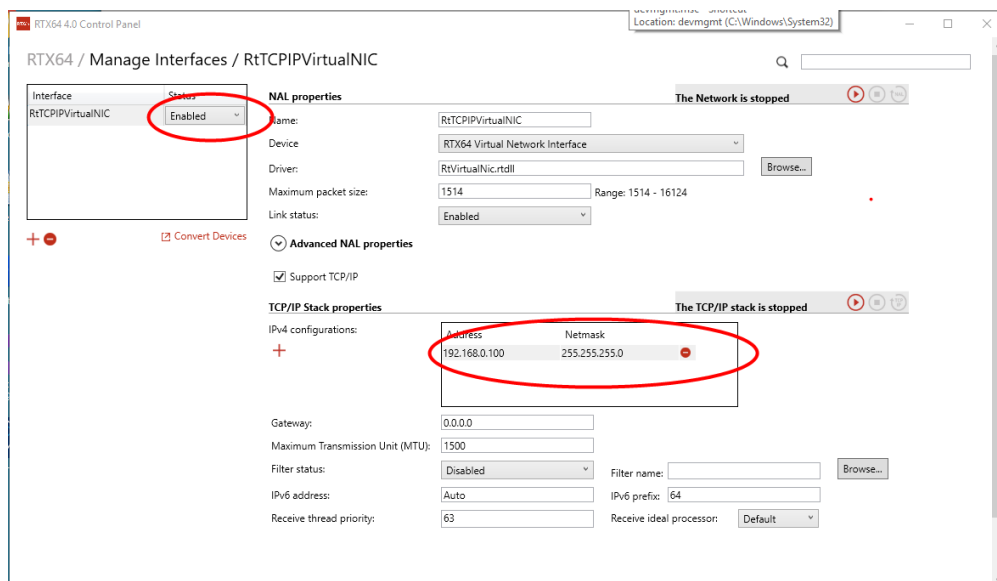
Spying process (ComET for example), connects to hardware via TCP/IP Virtual Networks and ETNE RTX thread. Port.properties file contains the customer application used port as accept port.

5.1. Configure the RTX 64 4.x Virtual network

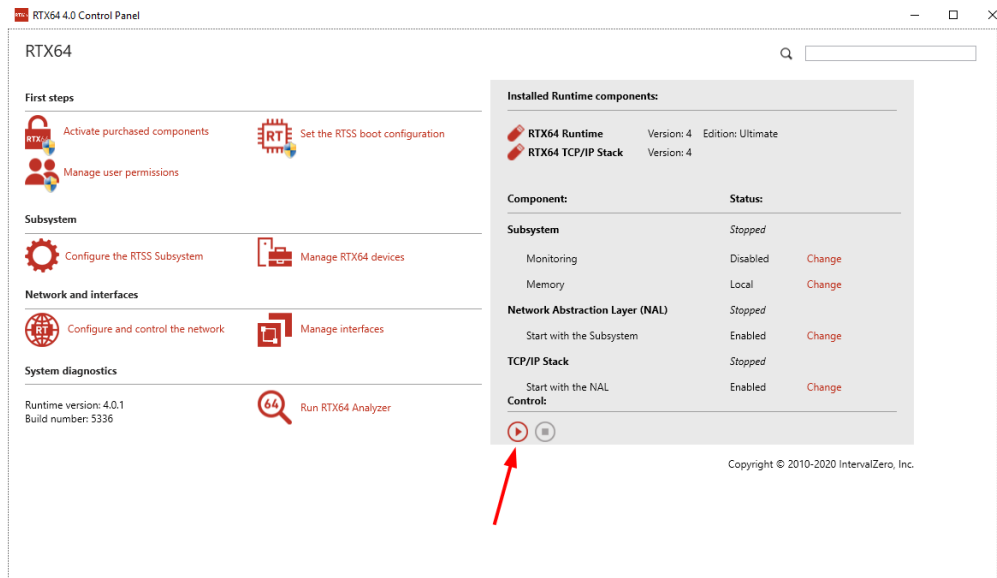
- Open RTX Control Panel to manage the interfaces



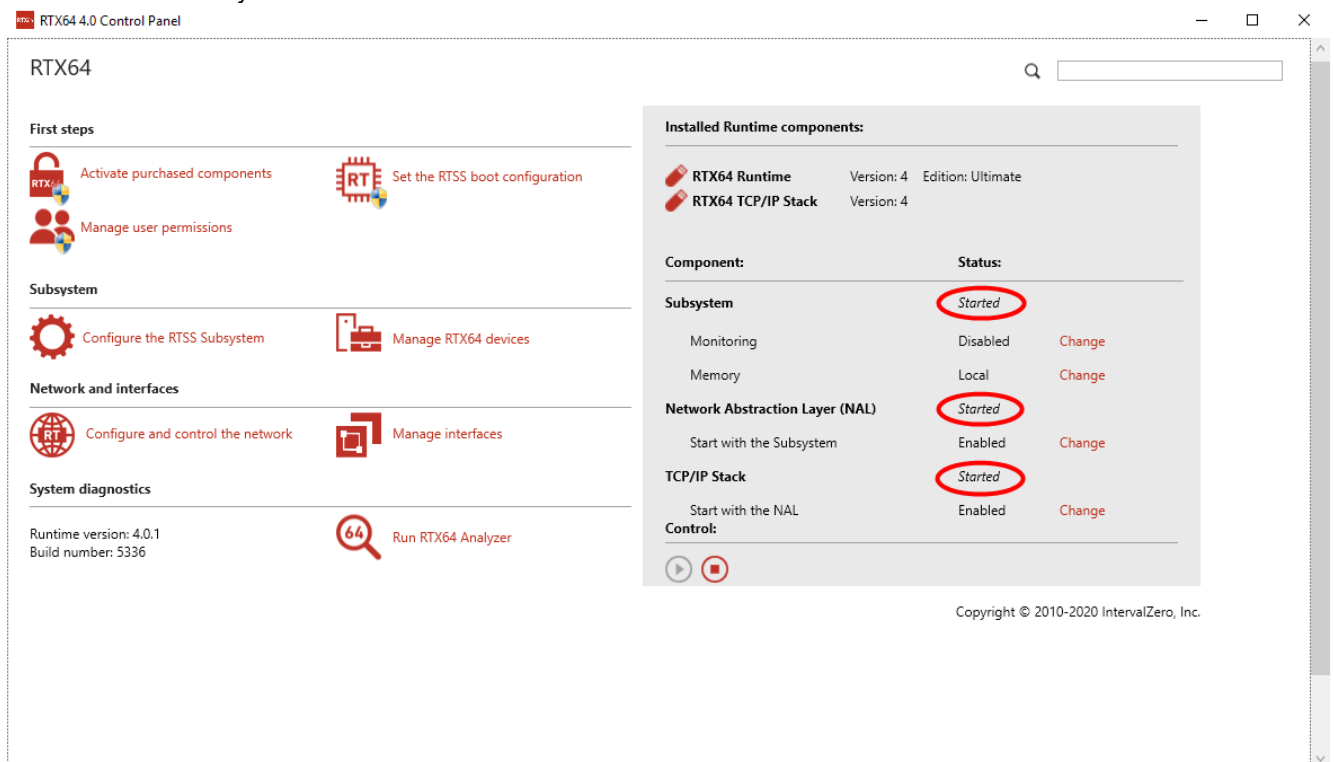
- Enable the RTXVirtualNic, and configure IP address and subnet mask:



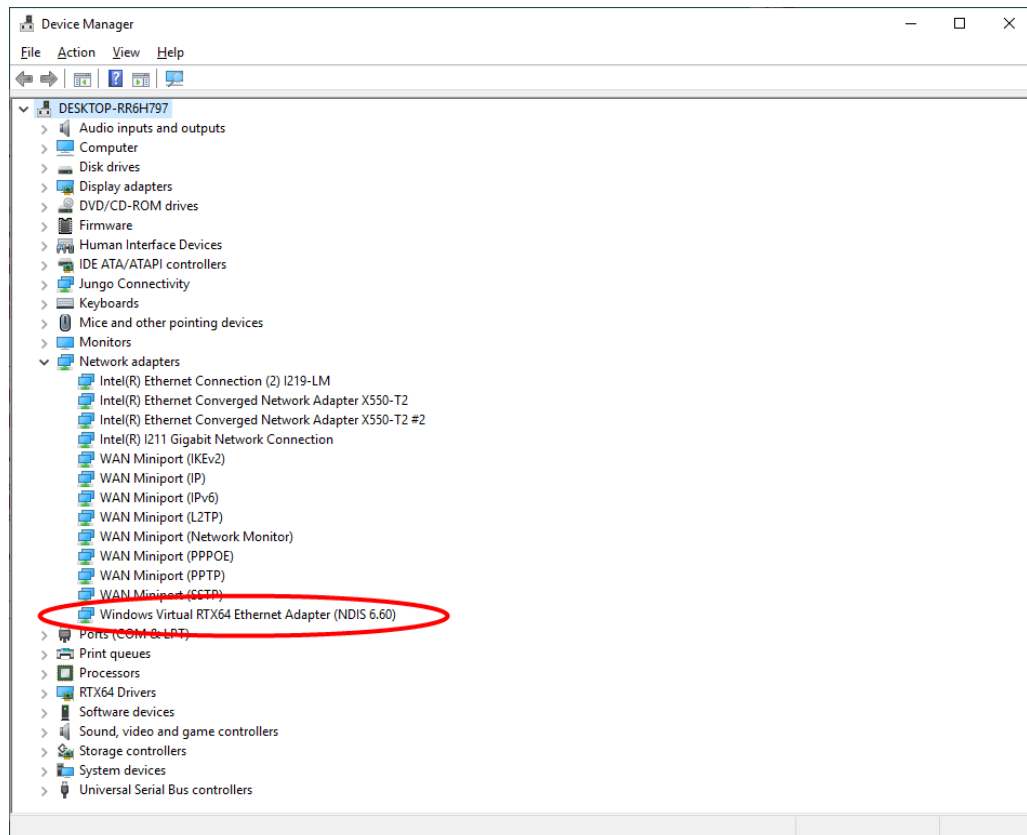
- Once done, start the TCP/IP stack. By default the TCP/IP stack is started with RTX Subsystem. Therefore, you can simply start the subsystem.



- Wait that RTX subsystem is started. This can take several seconds.

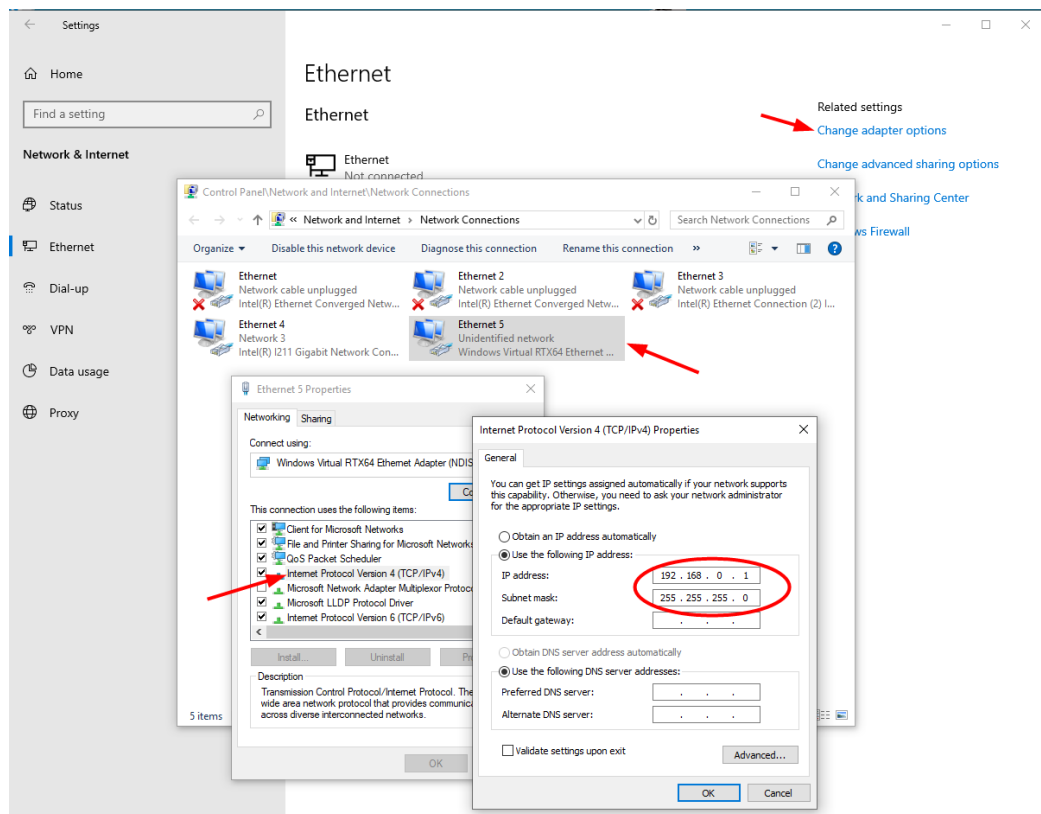


- As soon as the RTX TCP/IP stack is started, you can find a Windows Ethernet board. You can find it using the Windows Device Manager



- Check that the IP address and Subnet mask of the Virtual RTX Ethernet Adapter in Windows matches the RTX configuration. Configure it if needed:
 - o **Windows 10:**

Start > Control Panel > Network and Internet > Ethernet > Change adapter options > Virtual RTX Ethernet > Internet Protocol Version 4 (TCP/IPv4) > Properties > **Use the following IP address**



- You can now check that the connection is working by pinging the RTX Virtual Network board using a Windows console

```

Microsoft Windows [Version 10.0.17763.107]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\rtx>ping 192.168.0.1

Pinging 192.168.0.1 with 32 bytes of data:
Reply from 192.168.0.1: bytes=32 time<1ms TTL=128
Reply from 192.168.0.1: bytes=32 time<1ms TTL=128
Reply from 192.168.0.1: bytes=32 time<1ms TTL=128
Reply from 192.168.0.1: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\rtx>
    
```

5.2. Connection with EDI

Use:

- Use following url in the customer application:
 dsa_open_u (drv, "etb:<url>:<axis>");
 where:
 <url>: is the url used for the direct connection

- Allow host connection in the customer application, by adding the following line:
`etne_start_custom (etb, <accept_port>, "Windows Customer application");`
 where:
`etb` is the communication bus got using `dsa_get_etb_bus` function.
`<accept_port>`: is the port which will be defined in `port.properties`
- Define a `port.properties` file which define the wanted url as accept connection. This file MUST be in the same directory as COMET.
- run RTX customer application.
- run ComET. In the available connections, you will find the accept connection added in `port.properties`. Select it. ComET is now connected to the hardware.

Example:

The customer application will connect to axis 0 of the UltimET PCI, plugged in the PC running the customer application. The used port is the port 1140.

- Use following url in the customer application:
`dsa_open_u (drv, "etb:ULTIMET:0);`
- Allow host connection in the customer application, by adding the following line:
`etne_start_custom (etb, 1140, "Windows Customer application");`
- Create a `port.properties` file in the same directory as the ETN40D process. This file should contain the following block:

```
#parameters of port <n>
port.<n>.name="Windows Customer Application"
port. <n>driver=ETN:<IP address of RTXVirtualNic>:1140
port. <n>protocol=ETB
port. <n>accept.0.driver=ETN:0
port. <n>start=auto
```

<IP address of RTXVirtualNic> is the IP address configured in "Manage Interface and filter" menu of RTX Control Panel. (In the configuration example 192.168.0.100)

- run ComET, display the list of available connections. You will find a connection named "Windows Customer Application". Select it.

An example is provided with EDI package under <EDI-FOLDER>\samples\v40\c(Rtx)\etne

6. ANOTHER USE CASE

Compiling ETEL sequences is possible on Windows OS only. For Customer application running on another OS, it is thus not possible to compile and download ETEL sequences on the non-windows OS.

This can be solved using ETND – ETNE processes.

The user run its application on the non-windows OS. He then allows connection to its hardware by using one of the methods described above. After that, he can run ComET on a windows PC, connect to the hardware, compile and download sequence into the hardware.