

Application Note

HOW TO COMMUNICATE WITH R&S DEVICES USING VISA

R&S® VISA / NI VISA

Products:

- ▶ All R&S® Instruments supporting the described interface types

Peter Jahns | 1SL374 | Version 0e | 08.2021

<http://www.rohde-schwarz.com/appnote/1SL374>

ROHDE & SCHWARZ

Make ideas real



Contents

1	Overview.....	3
2	The different Interface Types.....	4
2.1	USB-VCP	4
2.1.1	Preparing the device	4
2.1.2	Installing the driver	4
2.1.3	Ensure correct installation of the driver	4
2.1.4	Test the communication in R&S VISA	5
2.1.5	Test the communication in NI VISA.....	6
2.2	USB-TMC.....	7
2.2.1	Preparing the device	7
2.2.2	Installing the driver	7
2.2.3	Ensure correct installation of the driver	7
2.2.4	Test the communication in R&S VISA	7
2.2.5	Test the communication in NI VISA.....	9
2.3	Other kinds of USB connections	10
2.3.1	Providing a Virtual Network Connection	10
2.3.2	NRP-Z Power sensors.....	11
2.4	Ethernet SCPI-Raw Socket (TCP Port 5025)	11
2.4.1	Preparing the device	11
2.4.2	Ping it	11
2.4.3	Test the communication in R&S VISA	11
2.4.4	Test the communication in NI VISA.....	13
2.5	Ethernet Standard Connections	14
2.5.1	Preparing the device	14
2.5.2	Ping it	14
2.5.3	Test the communication in R&S VISA	15
2.5.4	Test the communication in NI VISA.....	16
3	Termination of command lines.....	18
3.1	Explanation	18
3.2	Avoid Read Operation Failures in the R&S VISA Tester	18
3.3	Avoid Read Operation Failures in the NI VISA Test Panel.....	19
3.4	How to handle command line termination in applications.....	20
3.5	Controlling termination in your own programs	21
4	Literature.....	23

1 Overview

Using VISA (Virtual Instrument Software Architecture) several interfaces allow communication between your computer and instruments.

This FAQ describes the usage of both

- R&S® VISA

- NI VISA

running on Microsoft Windows 10™ with the following connection types:

- ▶ USB VCP (Virtual Com Port), also called USB CDC (Communications Device Class)
- ▶ USB TMC (Test and Measurement Class)
- ▶ Ethernet Raw Port Socket
- ▶ Ethernet in general e.g. VXI11 or Hislip

All the examples are shown using the R&S HMC8042 unless otherwise stated. There may be some difference when integrating other devices. As example some interface types are not available on every unit. This Application Sheet does not claim to be complete. All information has been compiled with care. However, errors can't be ruled out.

2 The different Interface Types

2.1 USB-VCP

This type of interface is more or less no longer contemporary, however it has some advantages. It simulates a COM port and so this way can also be used for testing with several terminal emulators. Many older applications allow to use COM ports.

2.1.1 Preparing the device

Enter the [SETUP] menu and change the „Interface“ to „USB VCP“ if not already set.

2.1.2 Installing the driver

For WIN10 the driver is distributed by Microsoft. This means, that if your computer has access to the internet, it will automatically request and install the correct VCP driver after the instrument is connected to the computer.

2.1.3 Ensure correct installation of the driver

After starting your device manager, you should find the instrument assigned to a COM port:

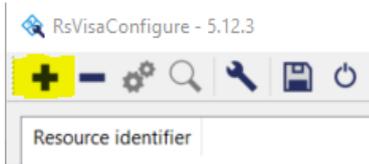


If this is not the case, check the device manager for the equipment. If your power supply can be found here, right click for properties and check the „Enhanced“ section, if VCP is enabled:

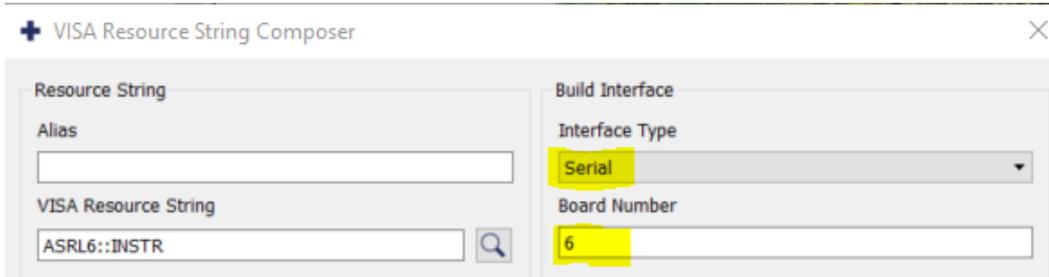


2.1.4 Test the communication in R&S VISA

Open RsVisaTester (e.g. START – R-Section – R&S VISA), click on the RsVisa Config section and add a new device using the "+" sign:



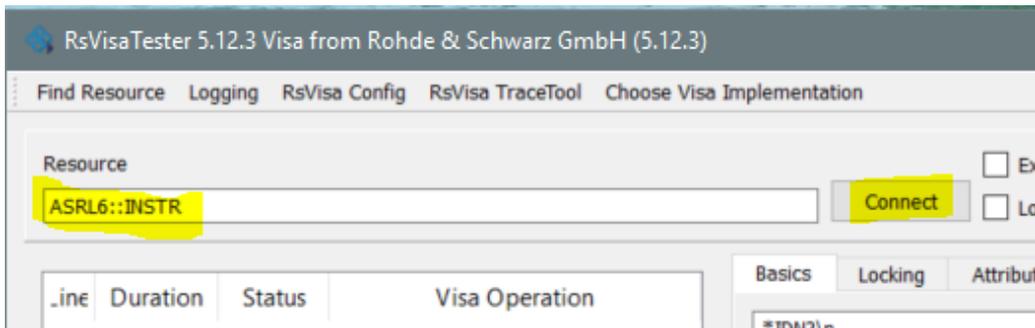
A new window appears where at least the interface type and the COM port number must be entered:



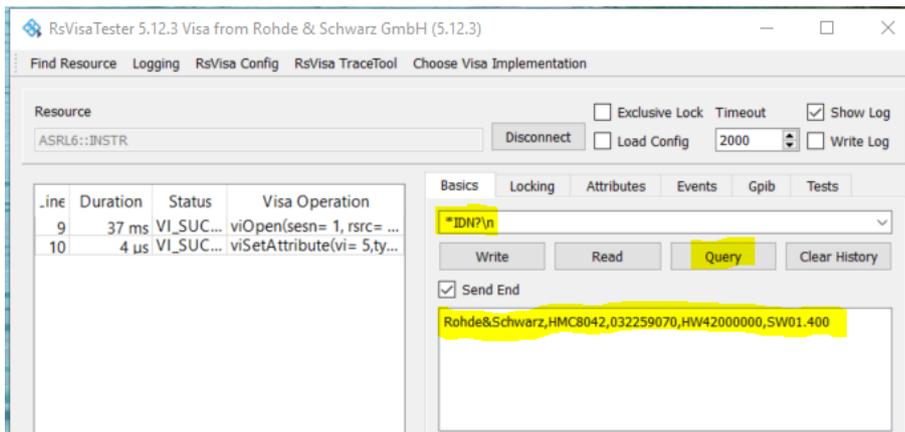
The VISA Resource String automatically appears and can easily be marked and copied into the buffer for the next step.

Leave the configure tool and do not forget to confirm the settings to be saved.

Being back in the tester just paste the Resource ID and press "Connect":



After successful connection query a "*IDN?". You should achieve the following result:



If you receive this answer, your device is configured correctly. If the device does not respond at all, it could have to do with the termination characters as described in Chapter 3.

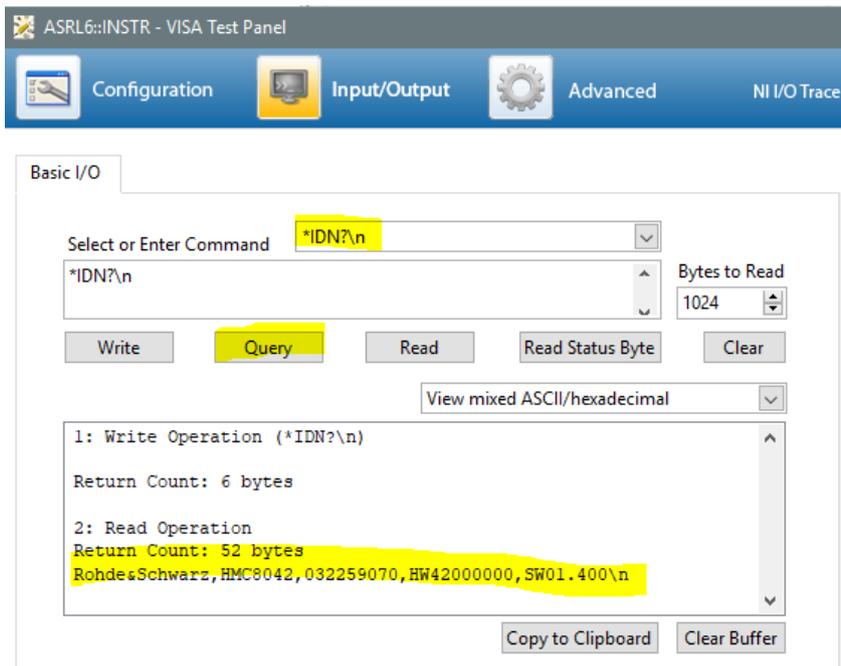
2.1.5 Test the communication in NI VISA

Open NI MAX (e.g. START – N-Section – NI MAX). You can see the COM-Port in the section „My system / Devices and Interfaces“:



If the according Port is still marked with the yellow warning, just press [F5], sometimes it takes a while before the port can be recognized by NI MAX.

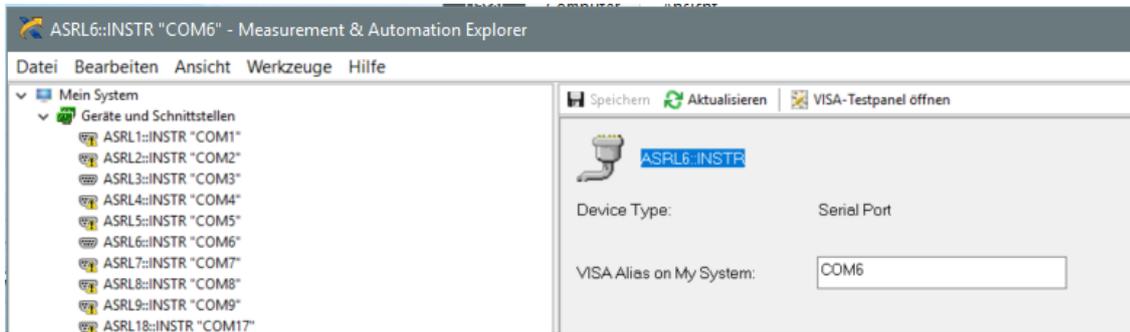
After the port shows the „unmarked view“, right click on it and choose „Open VISA Testpanel“. Change to the Input/ Output TAB and then perform a „*IDN?“ query:



If you receive this answer, your device is configured correctly.

If the device does not respond at all, it could have to do with the termination characters as described in Chapter 3.

The correct identifier string is to be found in the top line of the Test Panel, but the better way is to return to NI Max. and then mark and copy (use [CTRL]-[C]) the desired information:

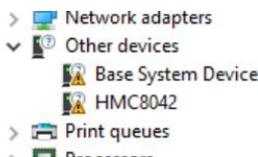


2.2 USB-TMC

USB TMC is the State-of-the-art USB interface. We recommend to use TMC to control measurement devices via USB.

2.2.1 Preparing the device

Enter the [SETUP] menu and change the „Interface“ to „USB “ if not already set.
If no driver is available on your system, it will appear in your device manager as follows:

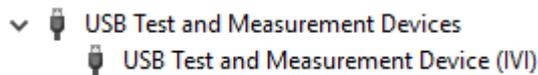


2.2.2 Installing the driver

Following the information in the manual, the TMC driver is automatically available after installing the R&S VISA or NI VISA package.

2.2.3 Ensure correct installation of the driver

In the device manager it will now appear in the „USB Test and Measurement Devices“:



2.2.4 Test the communication in R&S VISA

Open RsVisaTester 64 / 32bit (e.g. searching for rsvisa) and click on "Find Resource".
You can directly see the according USB device identifier:

Find and select resource

Resources

Resources	Details
Search...	
ASRL USB0::0x0AAD::0x0135::032259070	

If your measurement device is not automatically recognized by R&S VISA, RsVisaConfigure offers another way by adding the device manually. But where to find all these USB settings?

Resource String

Alias

VISA Resource String

SB0::0x0000::0x0000:::INSTR

Build Interface

Interface Type

USB

Board Number

0

USB

Vendor ID

0x0000

Product ID

0x0000

Device Serial No.

Those informations can be collected in the Windows device manager. All you have to do is to open the properties of the device and go to the Events tab. Every time stamp will tell you the desired values:

USB Test and Measurement Device (IVI)

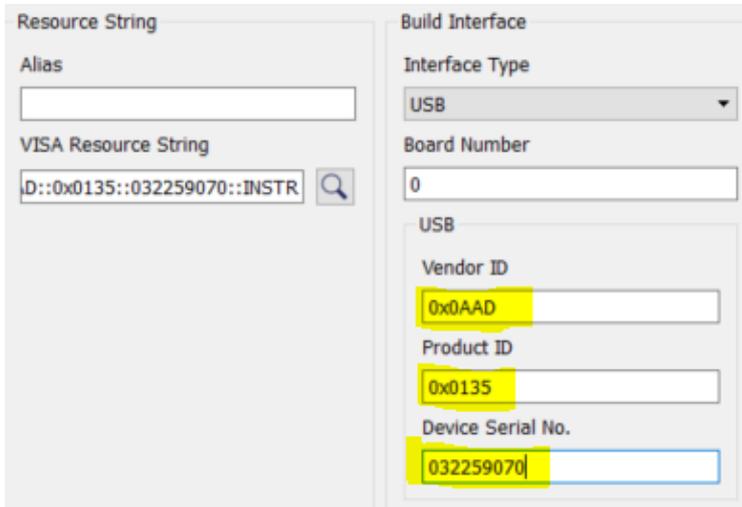
Ereignisse

Zeitstempel	Beschreibung
25.10.2019 13:35:35	Geräteinstallation angefordert
25.10.2019 13:35:35	Gerät migriert
16.03.2020 13:15:34	Gerät konfiguriert (oem166.inf)
16.03.2020 13:15:34	Gerät gestartet (Usbtmc)
21.07.2020 12:44:04	Geräteinstallation angefordert
21.07.2020 12:44:04	Treiberdienst hinzugefügt (usbser)

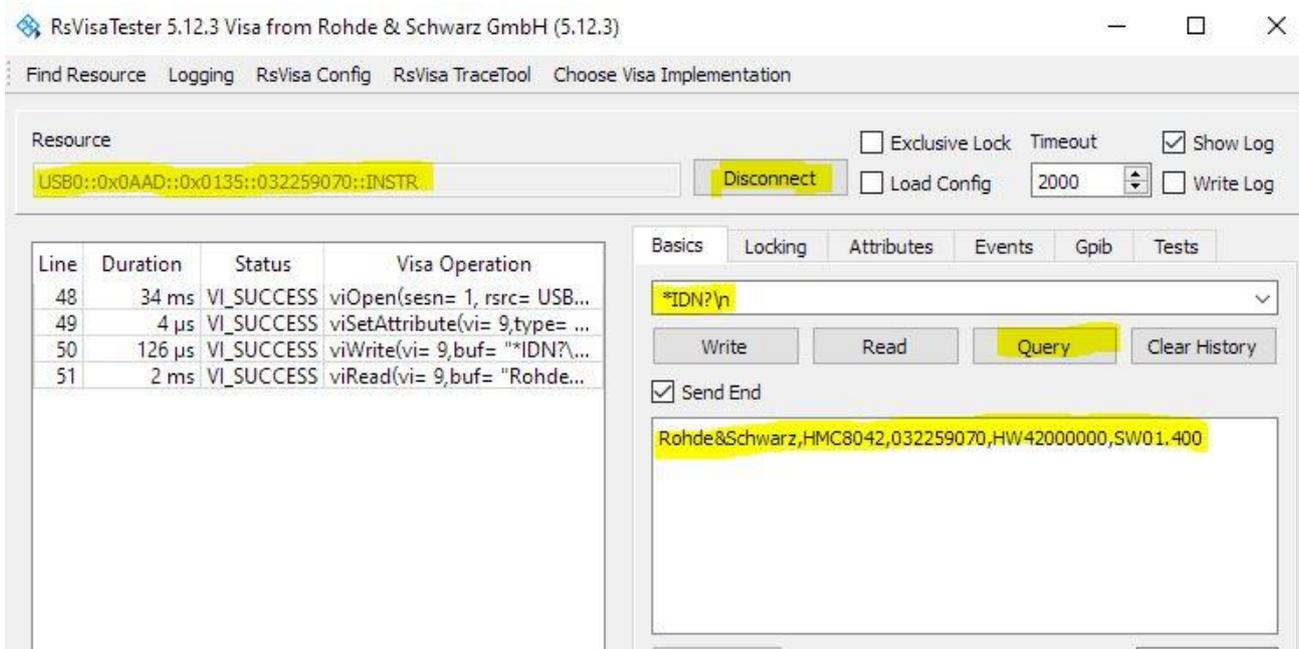
Informationen

Das Gerät "USB\VID_0AAD&PID_0135\032259070" erfordert weitere Installationen.

0AAD is the vendor ID, 0135 is the product ID, 032259070 corresponds to the serial number of the device. So, the correctly filled out form now should look like this:



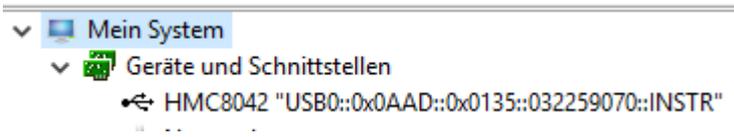
After the device is available in VISA, get back to the tester, connect the device and perform a query using the *IDN? string:



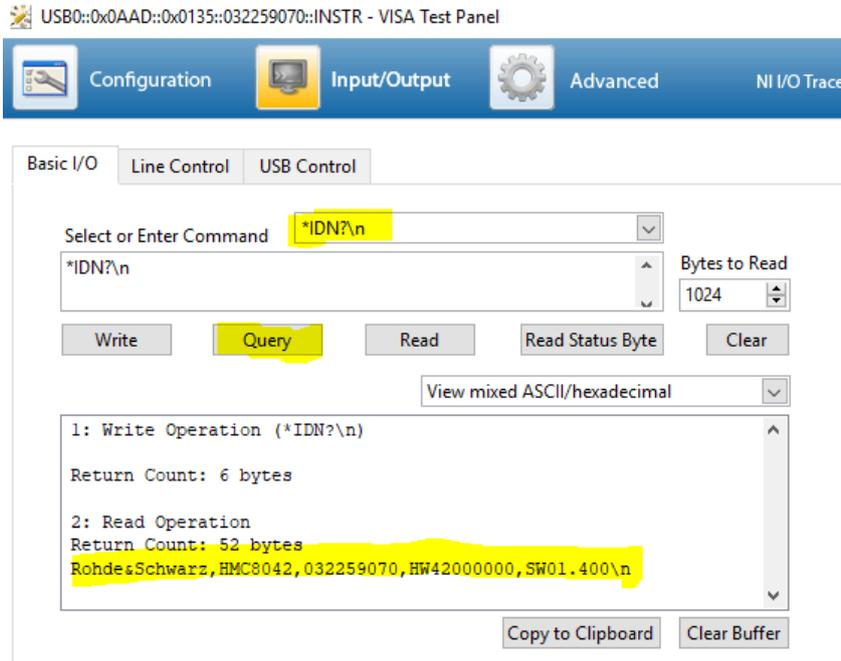
If your device responds like shown in the example, it is configured properly.

2.2.5 Test the communication in NI VISA

Open NI MAX (e.g. searching for NI MAX). You can see the USB-Port in the section „My system / Devices and Interfaces“:

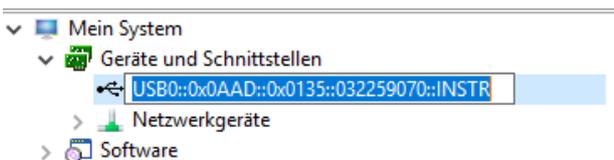


Now right click on it and choose „Open VISA Testpanel“.
 Change to the Input/ Output TAB and then perform a „*IDN?“ query:



If your device answers like described in the example, it is configured properly.

The correct identifier string is to be found in the top line of the Test Panel, but the better way is to return to NI Max. and then mark and copy (use [CTRL]-[C]) the desired information:



2.3 Other kinds of USB connections

2.3.1 Providing a Virtual Network Connection

Some devices like the FPC series or some handheld analyzers use the USB connection to provide an external network card. In this case the USB connection is used like a regular network connection. The corresponding IP address is described in the manual of the equipment.

2.3.2 NRP-Z Power sensors

Because these sensors do not support SCPI as standard, an additional software called "NRP VISA Passport" builds some kind of a bridge in the communication way when using NI VISA.

These sensors sometimes do not appear with the correct identifier string.

The typical USB string will work in the VISA tester but not from your application when it appears as "USB::0x0aad::0x0023::102800::INSTR".

One possible way to fix that unwanted behavior is to use "RSNRP::0x0023::102800::INSTR" instead.

The R&S VISA has the support for NRP-Zxx already integrated, no additional software installation needed.

2.4 Ethernet SCPI-Raw Socket (TCP Port 5025)

Some devices (e.g. the HMP series) only offer this way for communication over the network. If your device provides alternatives, raw port socket should not be your first choice for communication. Since Raw Socket really means raw communication, it does e.g. not support Device Locking or Service Request Handling. It does not correspond to the IEEE 488.2 protocol.

This connection type requires absolutely clean programming, otherwise it can lead to undesirable effects such as "mixed up answers".

2.4.1 Preparing the device

Enter the [SETUP] menu and change the „Interface“ to „ETHERNET “ if not already set.

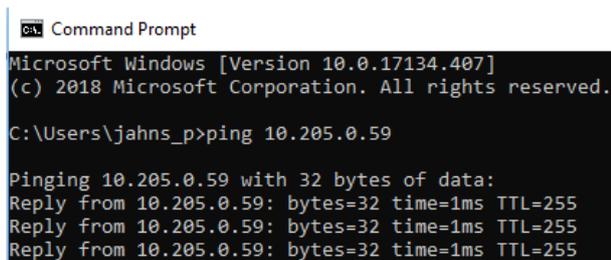
Perform your interface settings on the device (for example DHCP) and check for the assigned IP address. Do that also if you have defined a device name on your server (just to check if the device is really configured correctly).

The IP address can be found when opening the "PARAMETER" menu in the Interface section.

2.4.2 Ping it

Open a command line tool (e.g. search for cmd) and type „ping ip.ad.dre.ss“ followed by [Return].

The power supply should answer now.



```
Command Prompt
Microsoft Windows [Version 10.0.17134.407]
(c) 2018 Microsoft Corporation. All rights reserved.

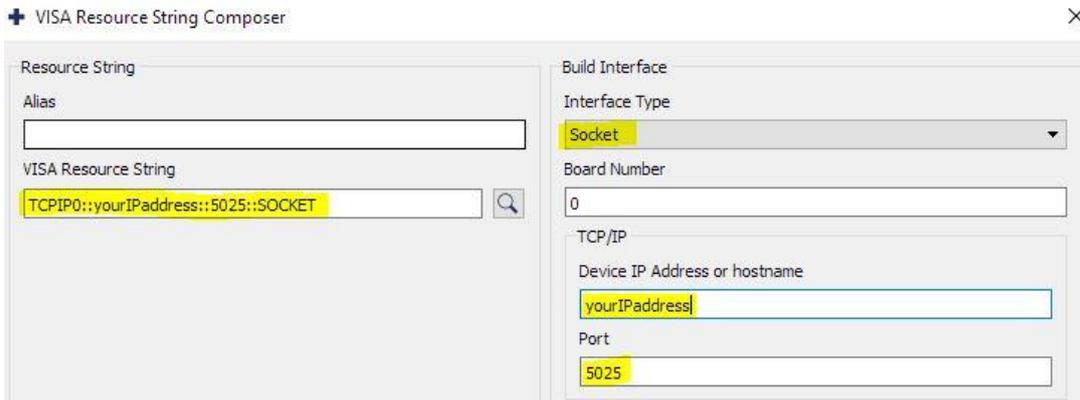
C:\Users\jahns_p>ping 10.205.0.59

Pinging 10.205.0.59 with 32 bytes of data:
Reply from 10.205.0.59: bytes=32 time=1ms TTL=255
Reply from 10.205.0.59: bytes=32 time=1ms TTL=255
Reply from 10.205.0.59: bytes=32 time=1ms TTL=255
```

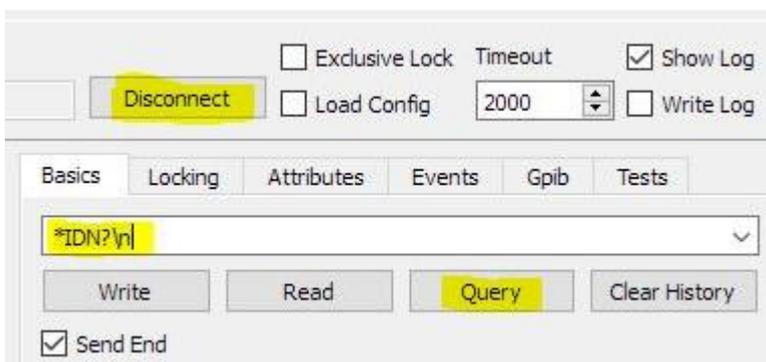
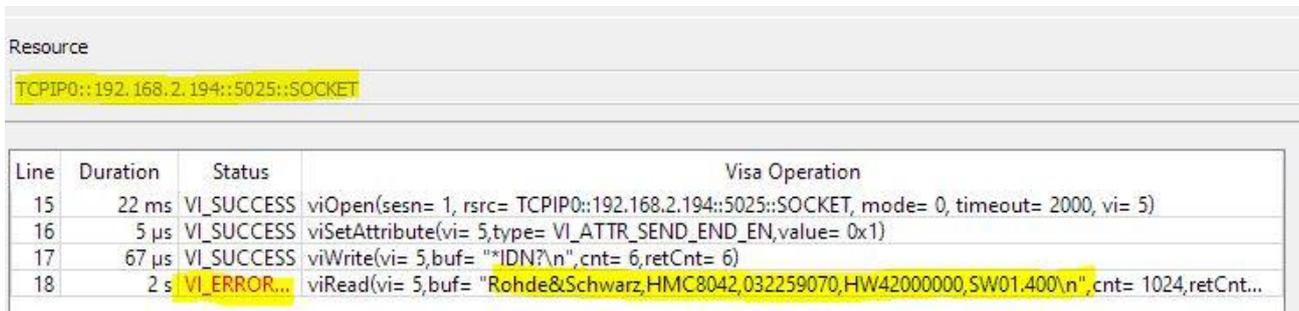
2.4.3 Test the communication in R&S VISA

Open RsVisaTester 64 / 32bit (e.g. searching for rsvisa) and click on "Find Resource". Usually the instrument will not be visible in the network devices at this early point. There may be some alternatives, but best way now is to setup the identifier manually.

- ▶ Click on "RsVisa Config"
- ▶ Click on the bold "+"
- ▶ Change the values in the next window as follows:



- ▶ Copy the VISA Resource String to the clipboard (e.g. by marking it and pressing <CTRL>-C)
- ▶ Click "OK" and leave RsVisaConfigure (confirm saving in the upcoming dialogue)
- ▶ Insert the Resource ID by pressing <CTRL>-V and connect the device.
- ▶ If *IDN? will be queried, the device will answer, but it leads to an unwanted error in this case



(Due to lack of page width this screenshot is divided into two lines)

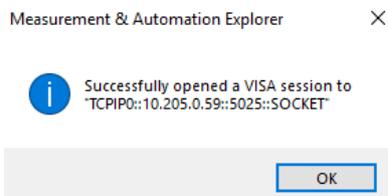
This happens because of missing correct termination while receiving. In the last response string we can see, that the answer arrived, but the Test Panel is waiting for 1024 Bytes (after "cnt=") and so it must end in a read operation error message as the number of Bytes is not reached within the timeout frame. If an adequate termination character would have been defined for these operations, it would override these expected number of characters as soon as a valid "end of transmission" character will be received. Now the question „Why did it work with the other communication types “ comes up. To ensure correct command / answer termination please refer to chapter 3. As the device responds correctly, the

communication is fine so far.

2.4.4 Test the communication in NI VISA

Open NI MAX (e.g. searching for NI MAX). Usually the equipment will not be visible in the network devices at this early point. There are some other possibilities available, but best way now is to setup the identifier manually.

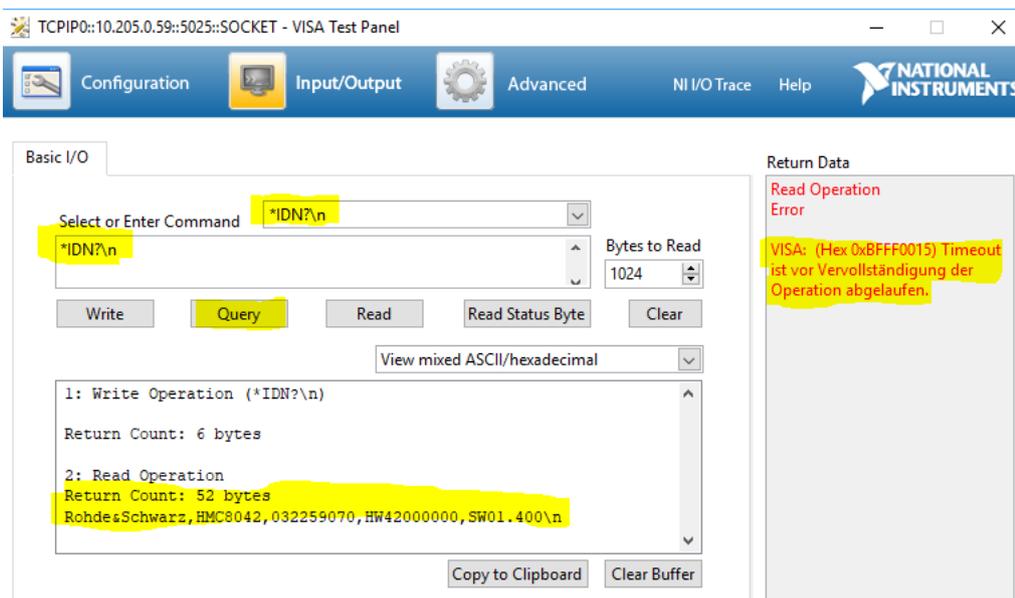
- ▶ Right click on „Network Devices“
- ▶ Choose „Create New VISA TCP/IP Resource“
- ▶ Manually Entry of Raw Socket
- ▶ Set your IP address and port 5025
- ▶ Click „Validate“ → If it works you should get such a message:



- ▶ Confirm the next steps and reach out for your power supply now.

```
o R&S SMW "TCPIP0::10.205.0.249::inst0::INSTR"  
o TCPIP0::10.205.0.59::5025::SOCKET  
o TCPIP0::10.205.0.93::5025::SOCKET  
o TCPIP0::10.205.0.96::5025::SOCKET
```

After you could identify your device, right click the adequate identifier to open the VISA Test Panel. Go to the Input / Output section then and query the *IDN? command:



It works in general, but raises a timeout failure. This is due to missing correct termination.

You can see, that the answer arrived, but the Test Panel is waiting for 1024 Bytes to come and so it must

end in a read operation error message. If an adequate termination character would have been defined for these operations, it would override these expected number of characters as soon as a valid termination character will be received.

Now the question „Why did it work before?“ comes up. To ensure correct command / answer termination please refer to chapter 3. As the device responded correctly, the communication is fine.

2.5 Ethernet Standard Connections

This is the most popular interface type to control instruments over the network.

We decide between VXI-11 and a Hi-SLIP connection.

Both protocols support GPIB-Emulation as well as the Message Exchange Protocol and Device Locking. However, there are some differences: While VXI-11 only supports IPv4 Hi SLIP is also able to use an IPv6 connection. Also, only HiSLIP (from 2.0 on) can be used to encrypt messages or for access authentication.

Typical ID strings are:

VXI-11: TCP/IP::10.205.0.123::inst0::INSTR

Hi-SLIP: TCP/IP::10.205.0.123::hislip0::INSTR

There are several other ways to define TCP/IP device IDs, these are just the most known ones.

2.5.1 Preparing the device

Enter the [SETUP] menu and change the „Interface“ to „Network“ if not already set.

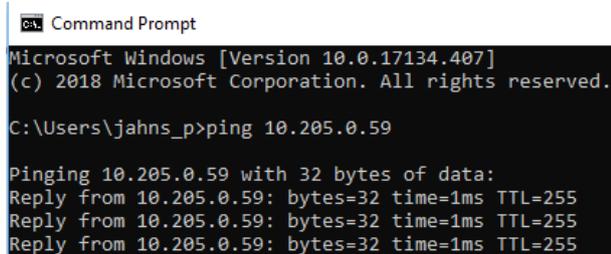
Perform your interface settings on the device (for example DHCP) and check for the assigned IP address. Do that also if you have defined a device name on your server (just to check if the device is really configured the right way).

The IP address can be found when opening the "PARAMETER" menu in the Interface section.

2.5.2 Ping it

Open a command line tool (e.g. search for cmd) and type „ping <ip_address>, for example ping 192.168.1.110, followed by [Return].

The power supply should answer now.



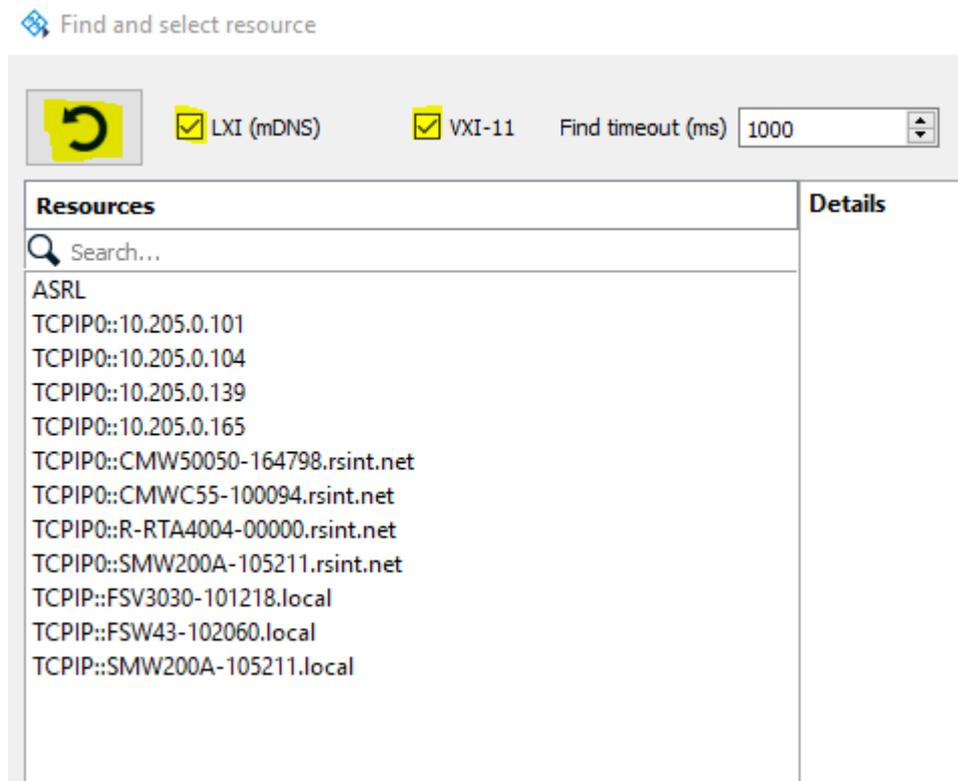
```
Command Prompt
Microsoft Windows [Version 10.0.17134.407]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\jahns_p>ping 10.205.0.59

Pinging 10.205.0.59 with 32 bytes of data:
Reply from 10.205.0.59: bytes=32 time=1ms TTL=255
Reply from 10.205.0.59: bytes=32 time=1ms TTL=255
Reply from 10.205.0.59: bytes=32 time=1ms TTL=255
```

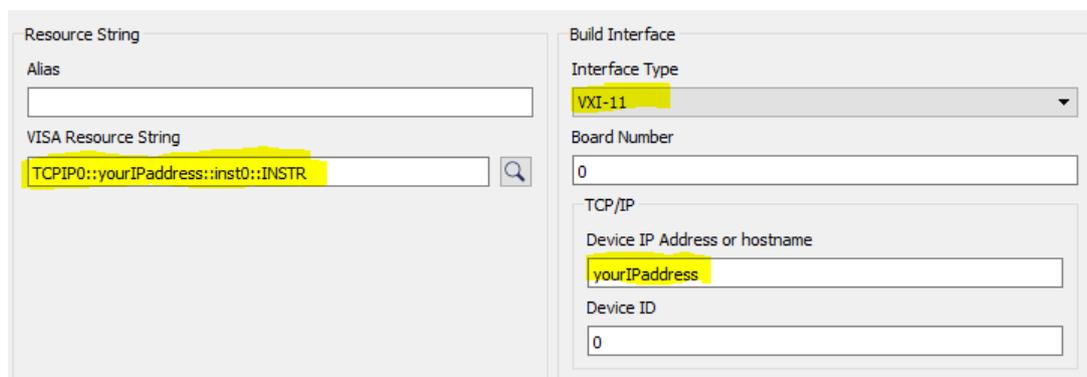
2.5.3 Test the communication in R&S VISA

Open RsVisaTester 64 / 32bit (e.g. searching for rsvisa) and click on "Find Resource". Maybe the instrument will not be visible in the network devices at this early point.



There are some alternatives, but since we already know our IP address, the best way is:

- ▶ Click on "RsVisa Config"
- ▶ Click on the bold "+"
- ▶ Change the values in the next window as follows (VXI-11 or HiSLIP @ Interface Type):



- ▶ Copy the VISA Resource String to the clipboard (e.g. by marking it and pressing <CTRL>-C)
- ▶ Click "OK" and leave RsVisaConfigure (confirm saving in the upcoming dialogue)
- ▶ Insert the Resource ID by pressing <CTRL>-V and connect the device.

- ▶ If *IDN? will be queried, the device will answer.

Resource: TCIP0::10.205.0.69::inst0::INSTR

Line	Duration	Status	Visa Operation
1	14 µs	VI_SUCCESS	viGetAttribute(vi= 1,type= VI_ATTR_RSRC_MANF_...
2	4 µs	VI_SUCCESS	viGetAttribute(vi= 1,type= VI_ATTR_RSRC_IMPL_V...
3	1 µs	VI_SUCCESS	viGetAttribute(vi= 1,type= VI_ATTR_RSRC_MANF_...
4	26 ms	VI_SUCCESS	viOpen(sesn= 1, rsrc= TCIP0::10.205.0.69::inst0:l...
5	14 µs	VI_SUCCESS	viSetAttribute(vi= 4,type= VI_ATTR_SEND_END_E...
6	775 µs	VI_SUCCESS	viWrite(vi= 4,buf= "*IDN?",cnt= 5,retCnt= 6)
7	41 ms	VI_SUCCESS	viRead(vi= 4,buf= "Rohde&Schwarz,HMC8042,03...

Basics Locking Attributes Events Gpib Tests

*IDN?

Write Read Query Clear History

Send End

Rohde&Schwarz,HMC8042,032259070,HW42000000,SW01.400

Clear Text Read count 1024

Your device is working fine now.

2.5.4 Test the communication in NI VISA

Open NI MAX (e.g. searching for NI MAX). Usually it will not be visible in the network devices at this early point. There are some other possibilities available, but best way now is to setup the identifier manually.

- ▶ Right click on „Network Devices“
- ▶ Choose „Create New VISA TCP/IP Resource“
- ▶ Manually Entry of LAN Instrument
- ▶ Set your IP address
- ▶ Click „Validate“ → If fit works you should get such a message:

Measurement & Automation Explorer

Successfully opened a VISA session to *TCIP0::10.205.0.59::INSTR

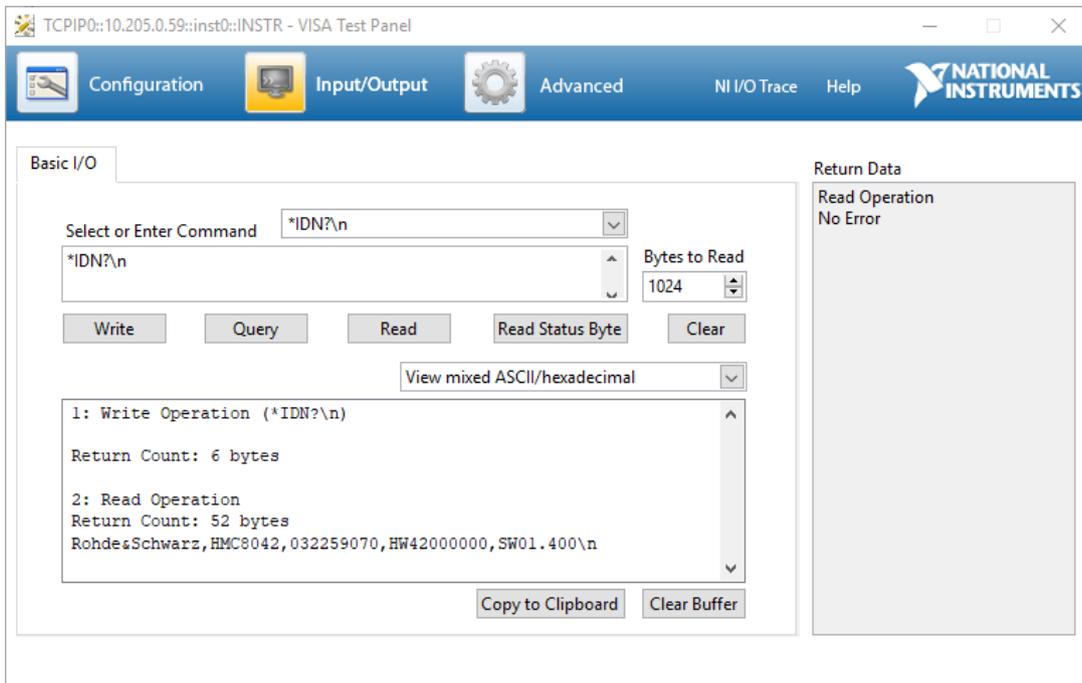
- ▶ Confirm the next steps and reach out for your power supply now.

```

TCIP0::10.205.0.45::5025::SOCKET
TCIP0::10.205.0.59::5025::SOCKET
TCIP0::10.205.0.59::inst0::INSTR
TCIP0::10.205.0.179::5025::SOCKET
TCIP0::10.205.0.237::5025::SOCKET

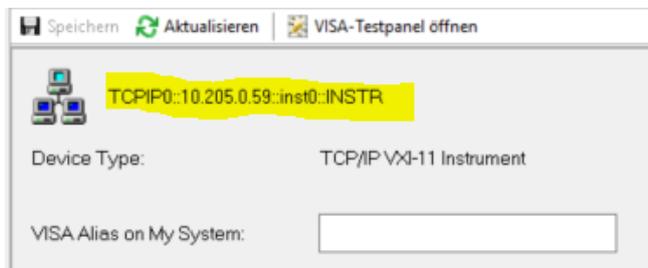
```

Right click the adequate identifier to open the VISA Test Panel. Go to the Input / Output section and query the *IDN? command:



Your device is working fine now.

The correct identifier string can be found in the top line of the Test Panel, but the better way is to return to NI Max. and then mark and copy (use [CTRL]-[C]) the desired information:



3 Termination of command lines

3.1 Explanation

All VISA resource types corresponding to the 488.2 protocol do not need a termination character. The reason is that this protocol has its own way to deal with the end of a message text. This covers the following resource types:

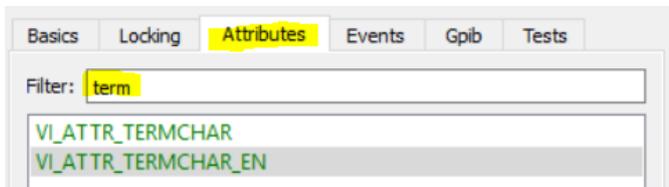
- GPIB Instr
- TCPIP Instr
- USB Instr
- VXI Instr

Only when using e.g. these resource types you have to specify the termination character separately:

- TCPIP Socket (which we use in this example)
- Serial Instr
- USB Raw

3.2 Avoid Read Operation Failures in the R&S VISA Tester

To avoid upcoming Read Operation Errors in the RsVisaTester, the following changes must be performed in the Attributes, after you have connected to the device with viOpen. In the tester change to the Attributes tab and assign "term" as filter.



Then click on the first attribute and the "Get Attribute" button.

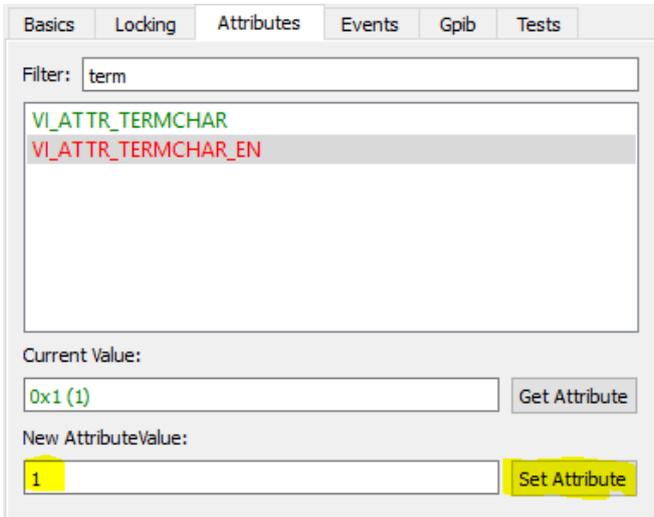
The default termination character is "0xa (10)" which means \n = ASCII 10 = LineFeed.

For most of the devices this is fine. Some instruments use CarriageReturn (\r = ASCII 13 = 0xD).

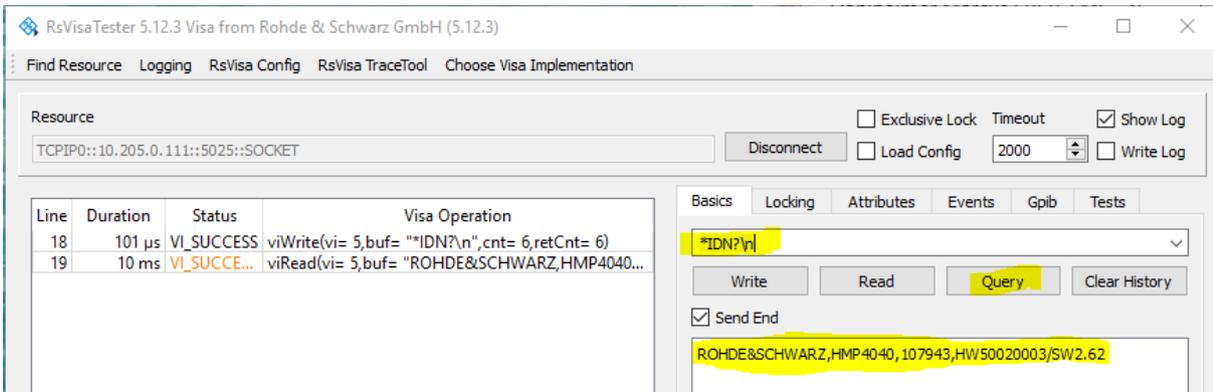
TERMCHAR can be only 1 Byte. In case the instrument has CRLF as TERMCHAR, only LF is fine.

You can change the assignment by entering the adequate number(s) in the "New Attribute Value" and clicking the "Set Attribute" button after.

A valid termination character setup will not help without enabling it.
 For the next step change to the second attribute and press the "Get Attribute" button. It will return with 0x0 (0) as this is the standard setting (not active).
 To activate the usage of the termination character, change this value to 1 by entering it in the "New Attribute Value" line and confirming it with the "Set Attribute" button:



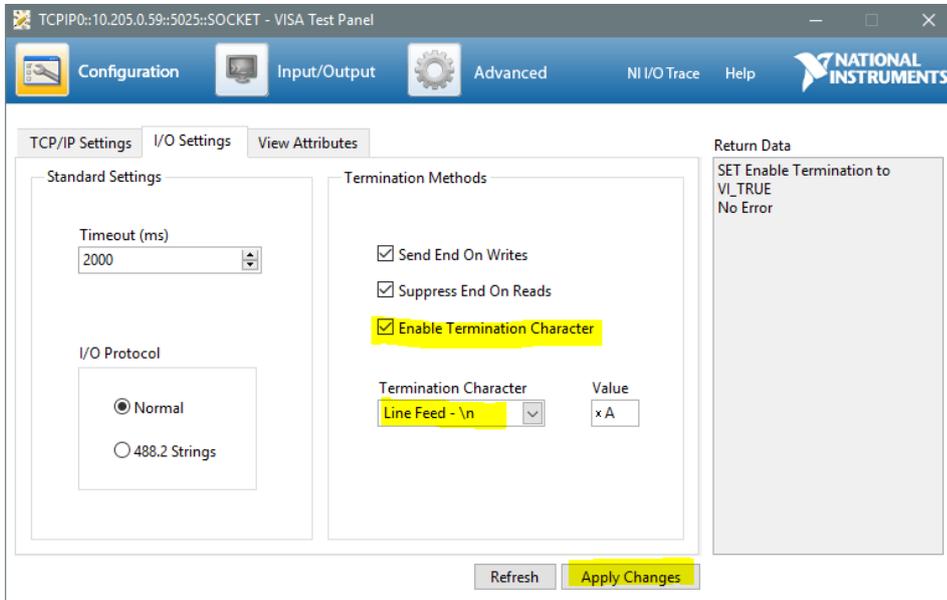
When now trying to query *IDN? it will work properly:



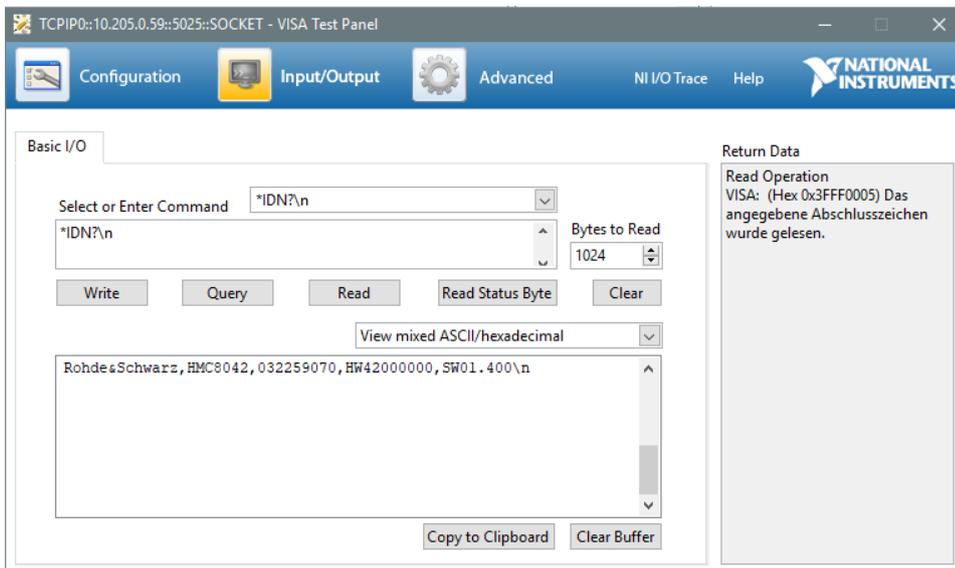
3.3 Avoid Read Operation Failures in the NI VISA Test Panel

To avoid upcoming Read Operation Errors in the VISA Test Panel, set the following in Configuration / I-O Settings.

Important notice: Do not forget to click on „Apply Changes“ to make your settings take effect.



When now trying to query *IDN? it will work without throwing an error message:



3.4 How to handle command line termination in applications

When an application is used to communicate with an instrument without having a dedicated driver available, the generic driver must be configured in terms of termination characters for both ways – receiving and transmitting.

If the application is not able to control that, be sure to use a VISA resource type corresponding to 488.2 protocol.

3.5 Controlling termination in your own programs

This Python example code shows how to handle it using PyVISA when writing your own applications:

```
# Set the termination value (no need to set here when already
# configured for specific device under
# "Settings/Instruments/<instrumentname>/Attributes/term_chars"
HMP4000.instrument.write_termination = '\n'
HMP4000.instrument.read_termination = '\n'
```

Either use the corresponding LabVIEW driver or follow the instructions on the NI website [2]

This is a short example written in Python 3.6 using the R&S RsInstrument. For more information please read the comments in the script:

```
"""
Created on 2021/08

Author: Jahns_P
Version Number: 1
Date of last change: 2021/08/03
Requires: R&S RTA4004, FW 1.600 or newer
- Installed RsInstrument Python module (see the attached RsInstrument_PythonModule folder Readme.txt)
- Installed VISA e.g. R&S Visa 5.12.x or newer

Description: Example to check for basic communication

General Information:
This example does not claim to be complete. All information has been
compiled with care. However errors can not be ruled out.
"""

from RsInstrument import *
from time import sleep

# =====
# Prepare instrument communication
# =====

resource = 'TCPIP0::10.205.0.18::inst0::INSTR' # Assign Instrument VISA resource string
RsInstrument.assert_minimum_version("1.15") # Check for RsInstrument version and stop if version is less than 1.15
#####
# The following initialisation includes these functions:
# Resource = assignment of the VISA resource string to R&S RTA4004
# Perform ID query
# Perform no Reset on the instrument
# Options:
# Set Termination character for read and write operations to \n, (ASCI 10 or Line Feed)
# Enable Logging Mode
# Set Write delay to 1 ms
# Set Read delay to 1 ms
#####
RTA4004 = RsInstrument(resource, True, False, options="TerminationCharacter = '\n',LoggingMode=On, WriteDelay = \"
    \"10, ReadDelay = 10 \")

RTA4004.logger.log_to_console = True # The Log is sent to the console
RTA4004.instrument_status_checking = False # Error check after each command MUST BE OFF for this instrument
RTA4004.visa_timeout = 3000 # Timeout for VISA Read Operations
RTA4004.opc_timeout = 3000 # Timeout for opc-synchronised operations
```

```
RTA4004.clear_status()           # Clear status register

# =====
# Do the first readout and get instrument data
# =====
idn = RTA4004.query_str('*IDN?')
print(f"\nHello, I am: '{idn}'")    # Identification String
print(f"Visa manufacturer: {RTA4004.visa_manufacturer}")
print(f"Instrument full name: {RTA4004.full_instrument_model_name}")

# =====
# Close the session
# =====

RTA4004.close()
```

4 Literature

- [1] Rohde & Schwarz, R&S HMC804x Power Supply User Manual, Version 04.
- [2] National Instruments, "Termination Characters in NI-VISA," [Online]. Available: <https://www.ni.com/de-de/support/documentation/supplemental/06/termination-characters-in-ni-visa.html>. [Accessed 25 August 2020].

Rohde & Schwarz

The Rohde & Schwarz electronics group offers innovative solutions in the following business fields: test and measurement, broadcast and media, secure communications, cybersecurity, monitoring and network testing. Founded more than 80 years ago, the independent company which is headquartered in Munich, Germany, has an extensive sales and service network with locations in more than 70 countries.

www.rohde-schwarz.com



Rohde & Schwarz training

www.training.rohde-schwarz.com

Rohde & Schwarz customer support

www.rohde-schwarz.com/support

