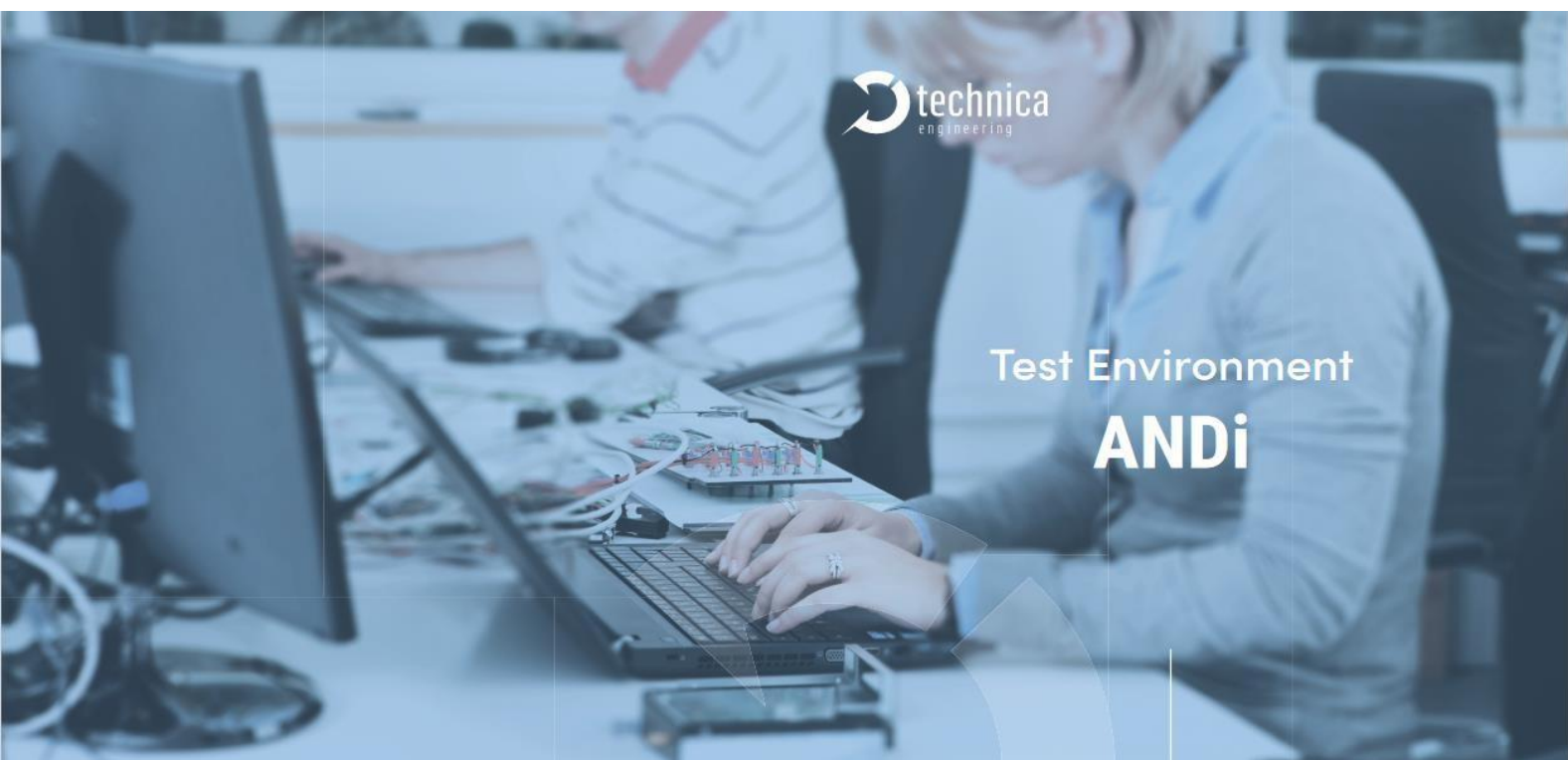




Test Environment
ANDi

ANDi SDK Manual

As Of May 2025

Manual Version : 2.1

Contents

Introduction.....	2
Requirements	2
Redistributables for Windows system	3
Installing .NET	3
Installing CodeMeter	4
Installing Npcap/WinPcap	8
Redistributables for Linux system	11
Installing .NET	11
Installing CodeMeter	12
Installing Libpcap	13
ANDi SDK installation	13
ANDi SDK update	13
User Manual	14
Scripts samples	15
Creating and sending a UDP message.....	15
Creating and sending a SOME/IP message with a database	16

Introduction

This document represents a manual on how to install and use ANDi SDK.

ANDi SDK is the portable lightweight ANDi software development kit. This kit comes shipped with python library that allows the use of ANDi scripting API directly from python 3.

Requirements

ANDi SDK is portable, it can be used on both Windows and Linux machines. Before running ANDi SDK, the following requirements need to be met:

- .NET 6 runtime: responsible for running ANDi library files (DLLs)
- CodeMeter: responsible for the handling of the licensing for ANDi SDK
- Npcap or WinPcap (Windows specific): responsible for hardware and packet handling

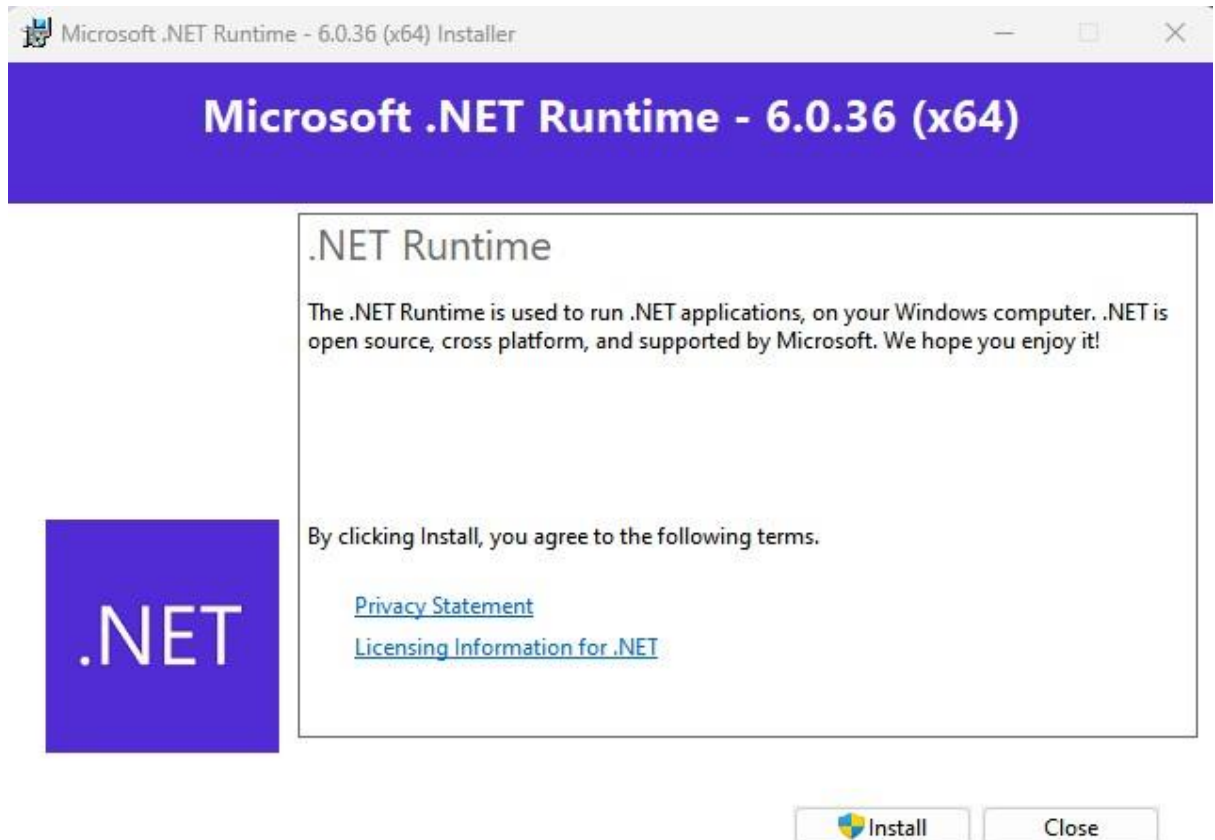
ANDi SDK also uses Python.NET library which enables running .NET code from python runtime. This library is installed automatically when installing ANDi SDK, so an internet connection is needed during packager installation.

Note: ANDi SDK is not functional on virtual machines.

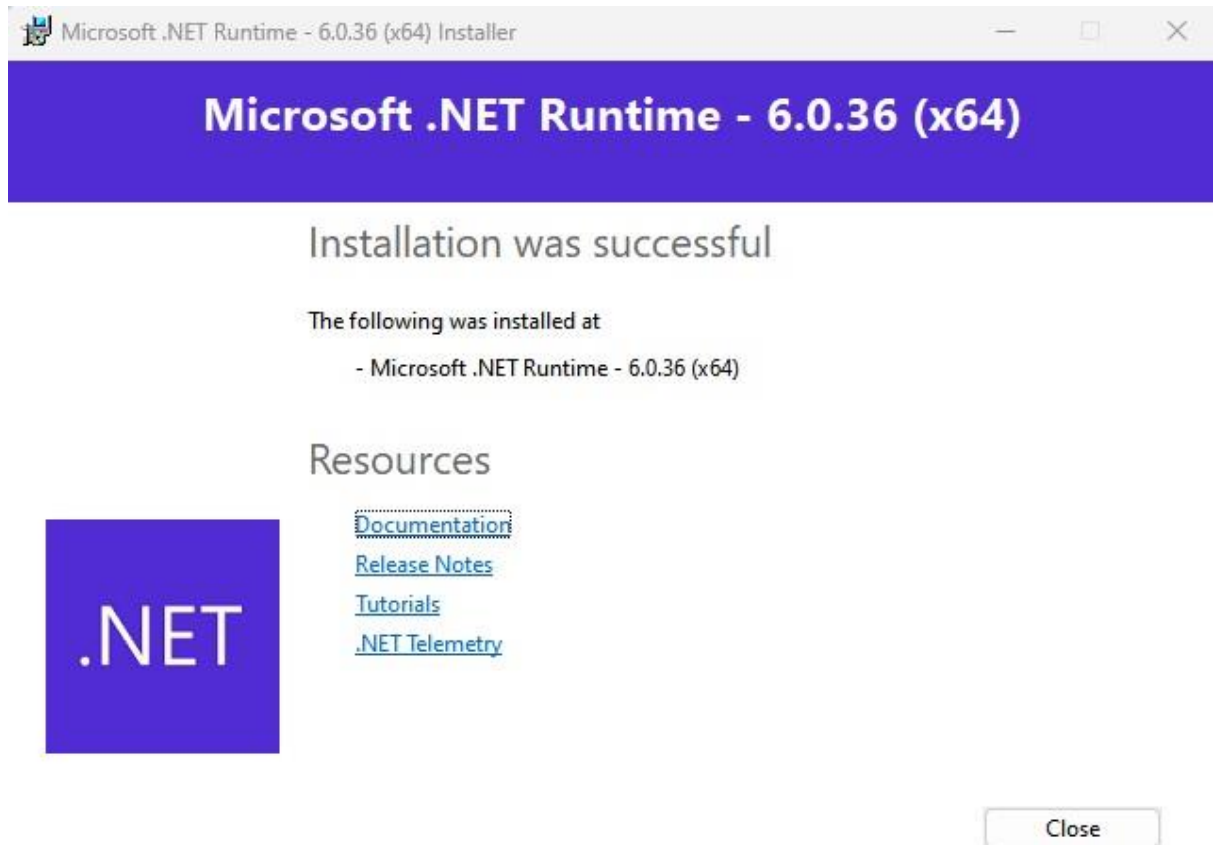
Redistributables for Windows system

Installing .NET

- 1- Download .NET runtime for windows from
<https://builds.dotnet.microsoft.com/dotnet/Runtime/6.0.36/dotnet-runtime-6.0.36-win-x64.exe>
- 2- Run the downloaded executable
- 3- Follow the instructions in the installer, click on install



- 4- Wait for the installation to be concluded and close the installer

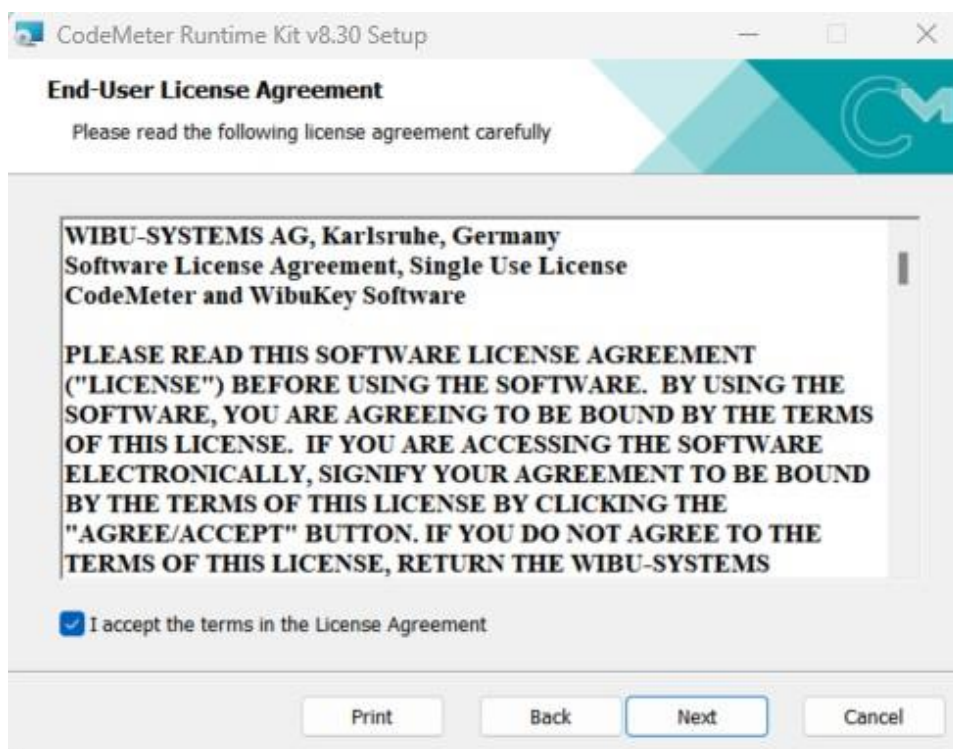


Installing CodeMeter

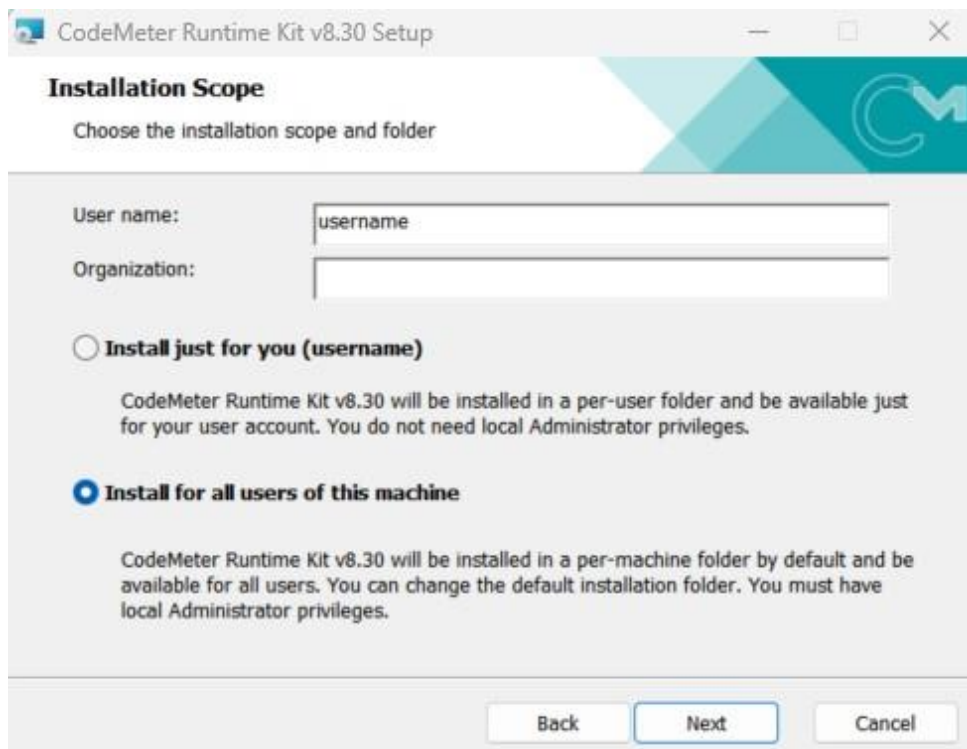
1. Download CodeMeter User Runtime for Windows from https://files.technica-engineering.de/ANDi/Redistributables/CodeMeterRuntime_8_30.msi
2. Run the downloaded executable
3. Follow the instructions in the installer



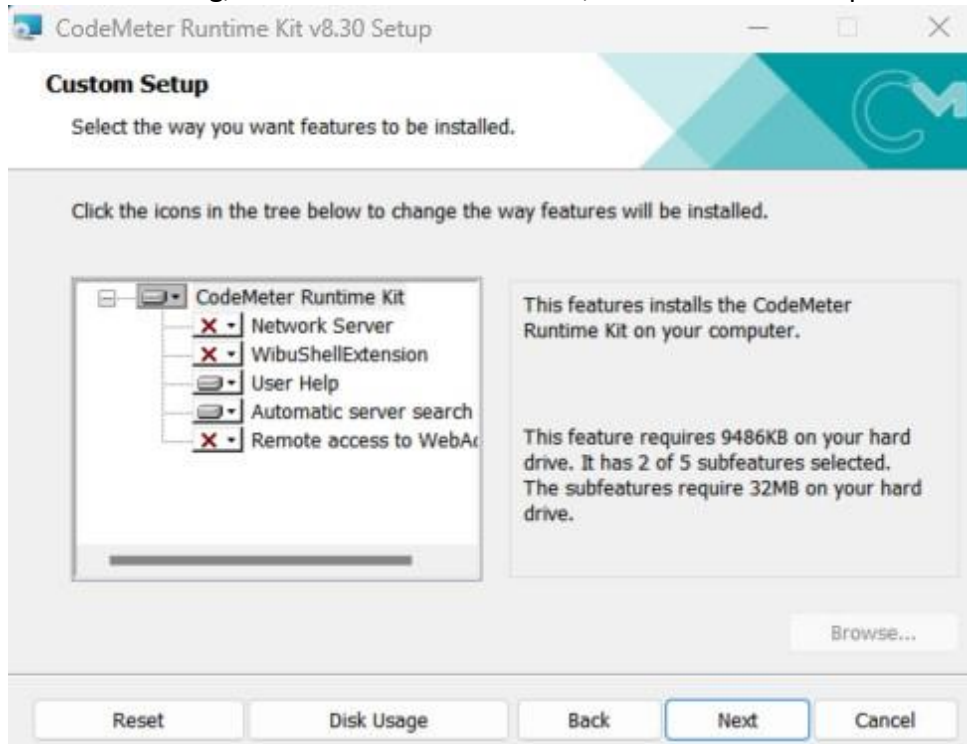
4. Accept the license agreement



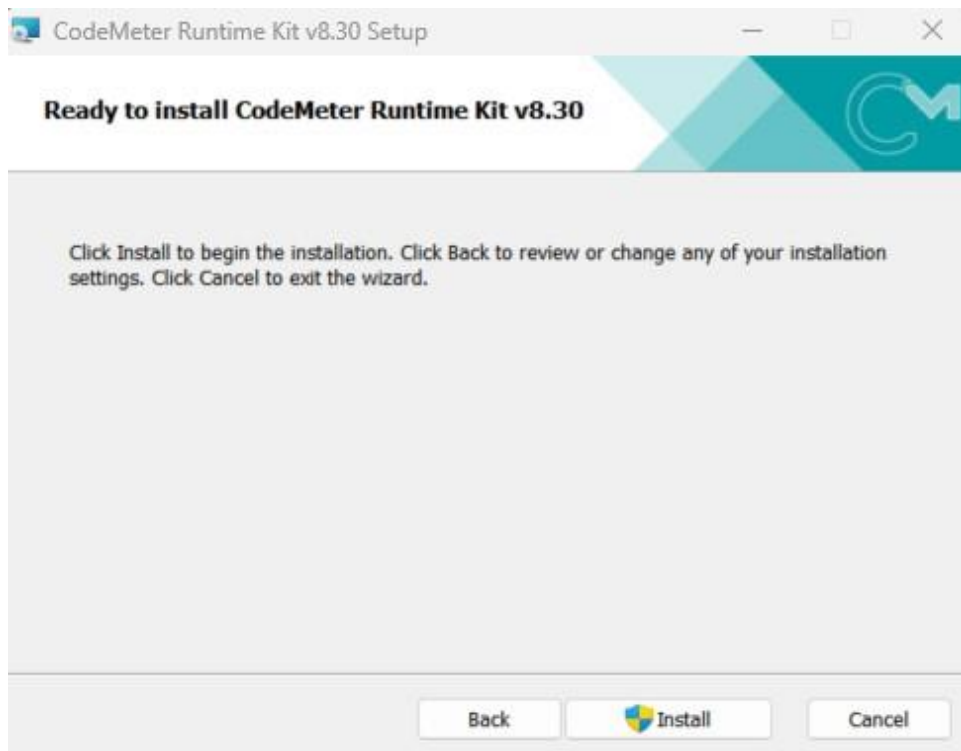
5. Choose a username and install for all users (Recommended)



6. We could choose to install remote access to web admin and network server for remote licensing, but it is not recommended, so it's better to keep the default values



7. Click install and wait for the installation to finish



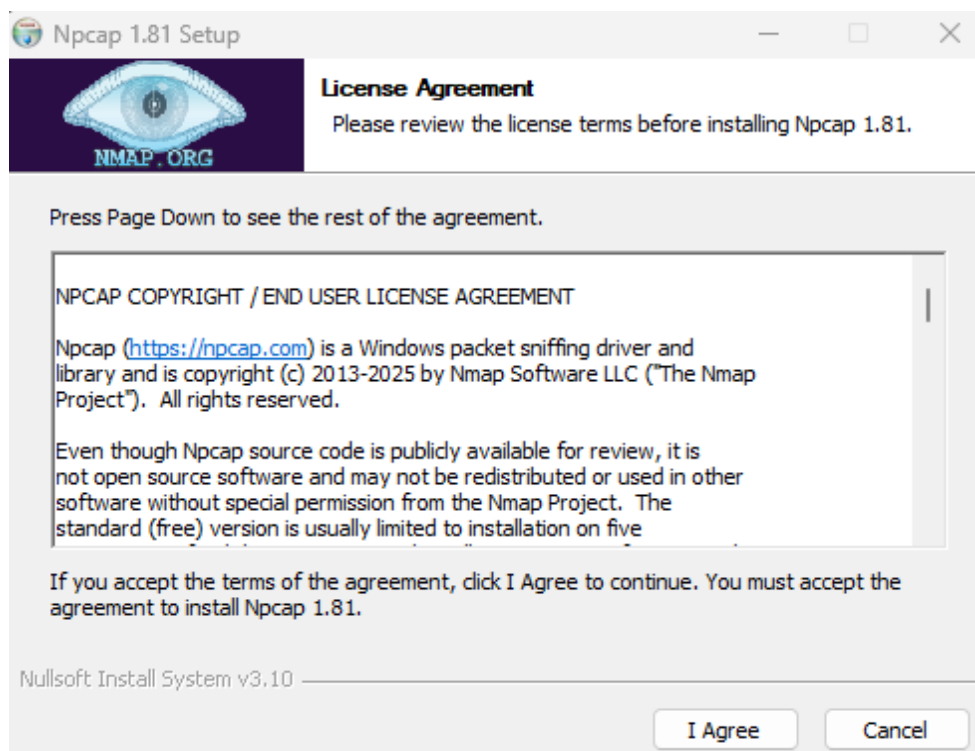
8. Click on finish when the installation is completed



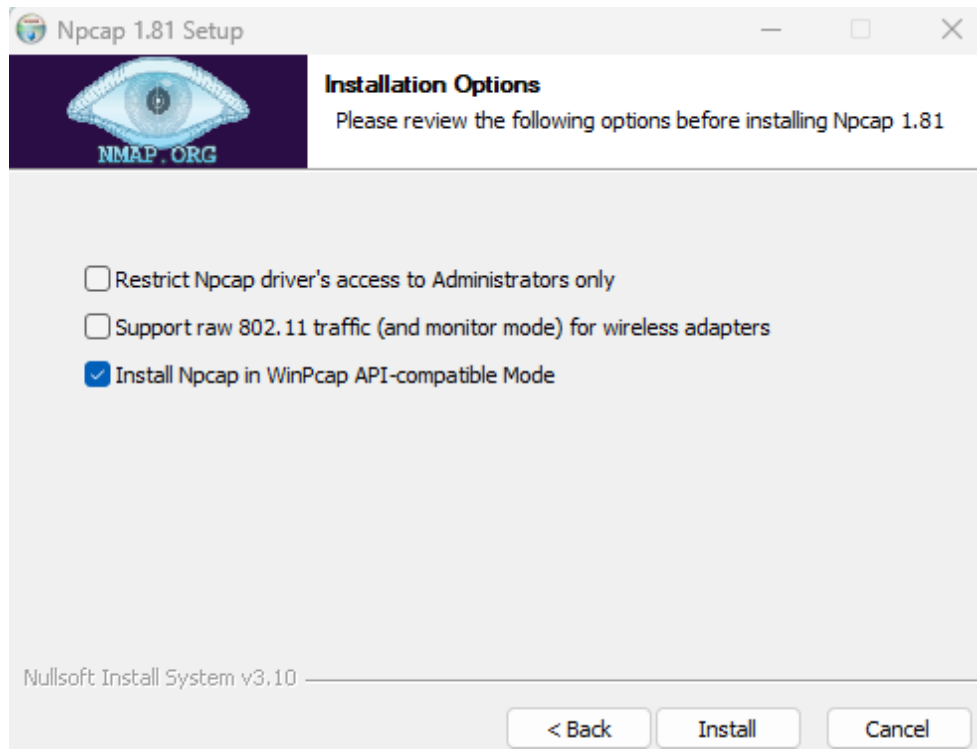
Installing Npcap/WinPcap

Officially, ANDi SDK needs Npcap at least version 1.79 to run properly, installing WinPcap works as well.

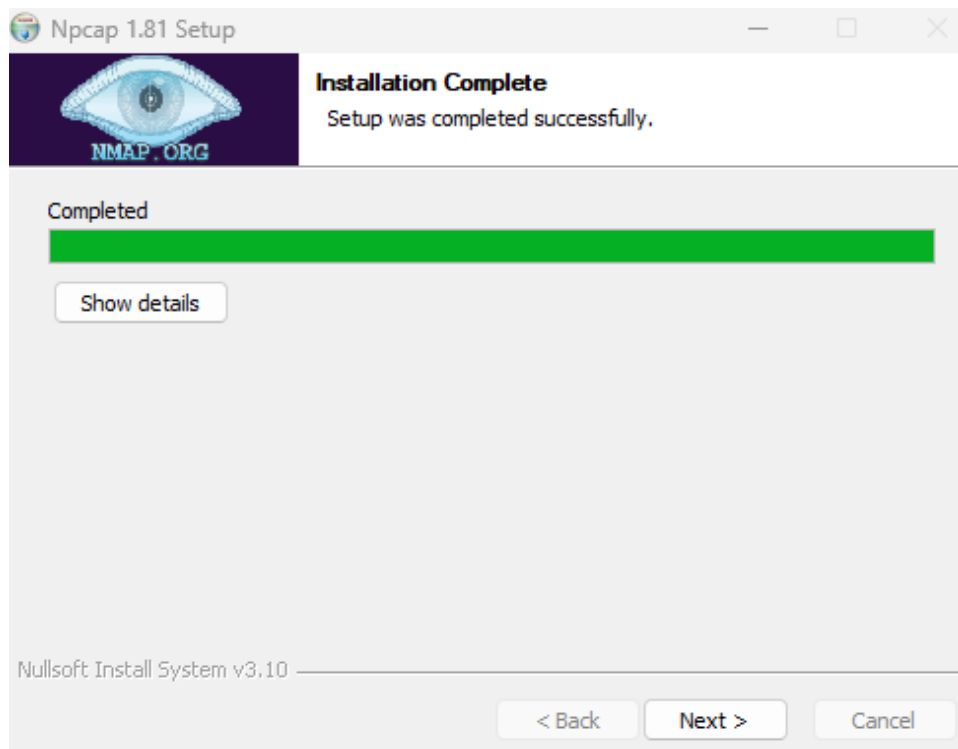
- 1- Download the Npcap installer from the website
<https://npcap.com/#download>
- 2- Run the downloaded executable
- 3- Follow the instructions in the installer
- 4- Agree to the license agreement



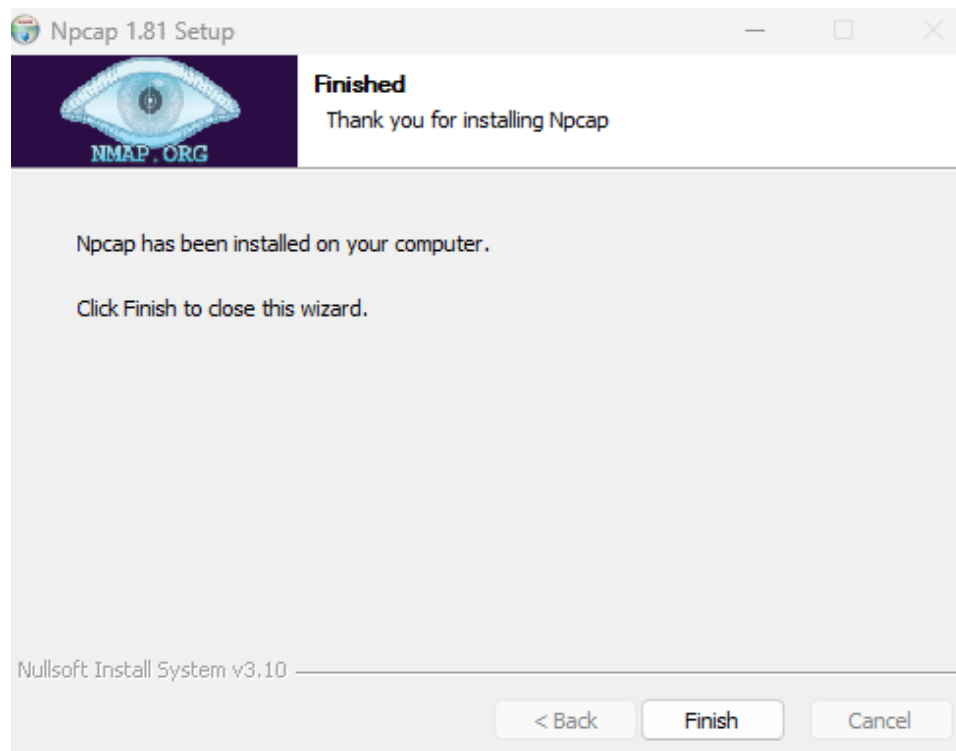
- 5- Keep the option “Install Npcap in WinPcap API-compatible mode” selected and click install



- 6- Wait for the installation to finish and click next



7- Click Finish to close the installer



⇒ Wireshark comes bundled with the latest version of Npcap, so If Wireshark is installed on the machine, this step can be skipped.

Redistributables for Linux system

Installing .NET

- 1- Download the .NET 6 Runtime

```
wget https://dotnetcli.azureedge.net/dotnet/Runtime/6.0.23/dotnet-runtime-6.0.23-linux-x64.tar.gz
```

- 2- Extract the Files

```
sudo tar -xzf dotnet-runtime-6.0.23-linux-x64.tar.gz -C /usr/share/dotnet
```

- 3- Delete the old Symlink

```
sudo rm /usr/bin/dotnet
```

- 4- Create a Symlink for Easy Access

```
sudo ln -s /usr/share/dotnet/dotnet /usr/bin/dotnet
```

- 5- Verify Installation

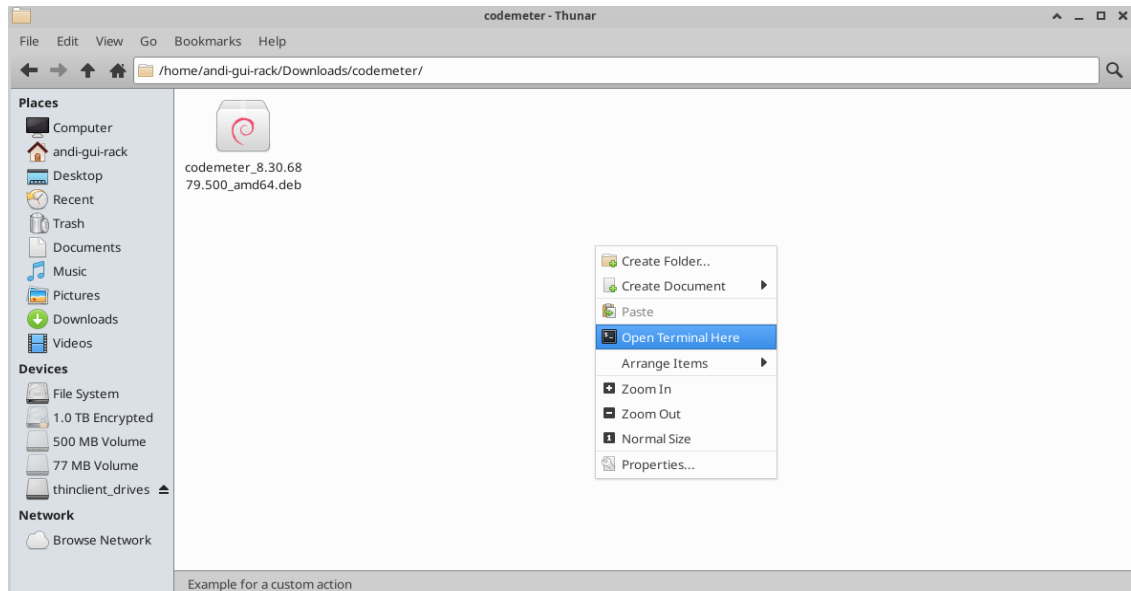
```
dotnet --list-runtimes
```

Note:

If this doesn't work on your machine, you can check this page for your machine installation: https://docs.microsoft.com/en-us/dotnet/core/install/linux?WT.mc_id=dotnet-35129-website

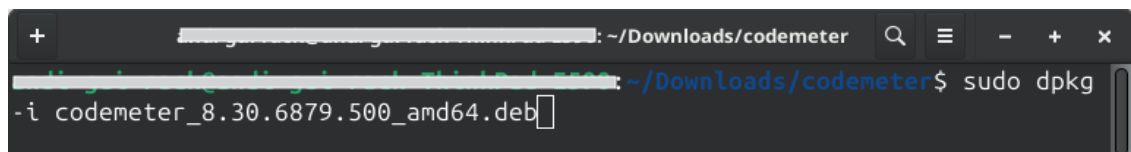
Installing CodeMeter

- 1- Go to the official [Wibu-Systems page](#).
- 2- Navigate to **Support & Downloads** → **User Software**.
- 3- Choose the appropriate package (**.deb** for Debian/Ubuntu, **.rpm** for RHEL/CentOS).
- 4- Open a terminal in the folder where the runtime file is

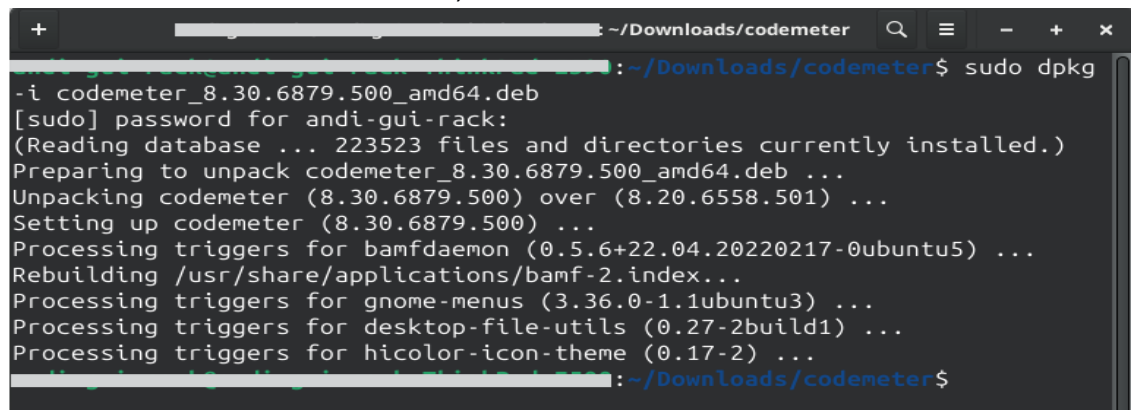


- 5- Run the following command

```
sudo dpkg -i runtime_file_name.deb
```



- 6- Wait for the installation to be over, the result should look like this



After installing CodeMeter, an ANDi-tool-license and a USB-dongle are needed. Both are provided by Technica Engineering. The license is fixed to the dongle. The dongle needs to be plugged into the machine before working with ANDi SDK.

For license related inquiries, this email: support@technica-engineering.de is available from Technica Engineering.

Installing Libpcap

Officially, ANDi SDK needs Libpcap on Linux to run properly.

The Libpcap can be installed using the following command line:

```
sudo apt-get install libpcap-dev
```

ANDi SDK installation

ANDi SDK can be installed using the following command line:

```
pip install andisdk
```

ANDi SDK update

From PyPi Server

If ANDi SDK is already installed, an upgrade is possible using the following command

```
pip install --upgrade andisdk
```

User Manual

ANDi SDK or the needed packages from it can be imported directly in python scripts.

```
from andisdk import load_project
# or
import andisdk
# or, starting from version 0.0.11 message_builder and andi objects are usable
without the need to load a project
from andisdk import message_builder, andi
```

ANDi SDK API is available after the appropriate imports.

- ⇒ When using IDEs that are compatible with python autocomplete, ANDi SDK provides autocomplete and type hinting capabilities.
- ⇒ In Linux, several ANDi SDK features (for example, adapters) require admin privileges.
- ⇒ Script execution will stop if a valid ANDi license is not detected.

ANDi SDK uses python logging module to log the messages happening in ANDi functionalities. Logging configured to log to a file using the function basicConfig

```
import logging
logging.basicConfig(filename="log_output_file.txt", filemode="w")
```

For a better and richer logging experience, rich logging handler can be used as a logging handler

This is done by installing rich python module using the command

pip install rich

And then logging is configured to use the rich handler

```
import logging
from rich.logging import RichHandler

FORMAT = "%(message)s"
logging.basicConfig(
    level="NOTSET", format=FORMAT, datefmt="[%X]", handlers=[RichHandler()]
)
```

This will give logging a look similar to this:

```
[14:49:49] INFO      Server starting... logging.py:62
[14:49:49] INFO      Listening on http://127.0.0.1:8080 logging.py:63
[14:49:50] INFO      GET /index.html 200 1298 logging.py:66
[14:49:50] INFO      GET /imgs/backgrounds/back1.jpg 200 54386 logging.py:67
[14:49:50] INFO      GET /css/styles.css 200 54386 logging.py:68
[14:49:50] WARNING   GET /favicon.ico 404 242 logging.py:69
[14:49:51] DEBUG     JSONRPC request logging.py:72
[14:49:51] DEBUG     --> {'version': '1.1', 'method': 'confirmFruitPurchase', 'params': [['apple',
[14:49:51] DEBUG     'orange', 'mangoes', 'pomelo'], 1.123], 'id': '194521489']}
[14:49:51] DEBUG     <-- {'version': '1.1', 'result': True, 'error': None, 'id': '194521489'}
[14:49:51] ERROR     Unable to find 'pomelo' in database! logging.py:82
[14:49:51] INFO      POST /jsonrpc/ 200 65532 logging.py:83
[14:49:51] INFO      POST /admin/ 401 42234 logging.py:84
[14:49:51] WARNING   password was rejected for admin site. logging.py:85
[14:49:51] ERROR     An error of some kind occurred! logging.py:89
[14:49:51] ERROR     Traceback (most recent call last):
[14:49:51] ERROR     File "/Users/willmcgugan/projects/rich/rich/logging.py", line 87, in
[14:49:51] ERROR     <module>
[14:49:51] ERROR     1 / 0
[14:49:51] ERROR     ZeroDivisionError: division by zero
[14:49:52] CRITICAL  Out of memory! logging.py:91
[14:49:52] INFO      Server exited with code=-1 logging.py:92
```

Scripts samples

Creating and sending a UDP message

```
from andisdk import load_project
import sys
api = load_project("empty_project.atp")

adapters = api.andi.get_adapters()

if (len(adapters) <= 0):
    print("No adapters found, stopping script")
    sys.exit()

adapter = adapters[0]
print("using adapter " + adapter.id + " to send udp message")
channel = api.andi.create_channel("Ethernet")
message = api.message_builder.create_udp_message(channel, channel)

message.payload = tuple([0x01, 0x02, 0x03, 0x04])
message.udp_header.port_source = 1234

print("sending udp message with payload " + str([x for x in message.payload]))
message.send()
```

Script version without a project (starting from version 0.0.11)

```
from andisdk import message_builder, andi
import sys

adapters = andi.get_adapters()

if (len(adapters) <= 0):
    print("No adapters found, stopping script")
    sys.exit()

adapter = adapters[0]
print("using adapter " + adapter.id + " to send udp message")
channel = andi.create_channel("Ethernet")
message = message_builder.create_udp_message(channel, channel)

message.payload = tuple([0x01, 0x02, 0x03, 0x04])
message.udp_header.port_source = 1234

print("sending udp message with payload " + str([x for x in message.payload]))
message.send()
```

Creating and sending a SOME/IP message with a database

```
from andisdk import load_project
import sys
api = load_project("empty_project.atp")

adapters = api.andi.get_adapters()
database = api.andi.load_database("someip_database.xml")

if (len(adapters) <= 0):
    print("No adapters found, stopping script")
    sys.exit()

adapter = adapters[0]
print("using adapter " + adapter.id + " to send someip message")
channel = api.andi.create_channel("Ethernet")
message = api.message_builder.create_someip_message(channel, channel)
message.data_base = database
message.someip_header.service_identifier = 0xb011
message.someip_header.method_identifier = 0x000e
message.set_input_param("statusReceiptRequestInstrument", 2)
message.send()
```

Script version without a project (starting from version 0.0.11)

```
from andisdk import message_builder, andi
import sys

adapters = andi.get_adapters()
database = andi.load_database("someip_database.xml")

if (len(adapters) <= 0):
    print("No adapters found, stopping script")
    sys.exit()

adapter = adapters[0]
print("using adapter " + adapter.id + " to send someip message")
channel = andi.create_channel("Ethernet")
message = message_builder.create_someip_message(channel, channel)
message.data_base = database
message.someip_header.service_identifier = 0xb011
message.someip_header.method_identifier = 0x000e
message.set_input_param("statusReceiptRequestInstrument", 2)
message.send()
```