PRONETA Professional User Manual

PRONETA Professional V1.1 SP2

https://support.industry.siemens.com/cs/ww/en/view/109768642

Siemens Industry Online Support

Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

 **DANGER**	**indicates that death or severe personal injury will result if proper precautions are not taken.**

 **WARNING**	**indicates that death or severe personal injury may result if proper precautions are not taken.**

 **CAUTION**	**indicates that minor personal injury can result if proper precautions are not taken.**

NOTICE	**indicates that property damage can result if proper precautions are not taken.**

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The product/system described in this documentation may be operated only by personnel qualified for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

Siemens AG
Division Digital Factory
Postfach 48 48
90026 NÜRNBERG
Germany

V1.1 SP2
Ⓟ 03/2023 Subject to change

WARNING

Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks.

In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens' products and solutions only form one element of such a concept.

Customer is responsible to prevent unauthorized access to its plants, systems, machines and networks. Systems, machines and components should only be connected to the enterprise network or the internet if and to the extent necessary and with appropriate security measures (e.g. use of firewalls and network segmentation) in place.

Additionally, Siemens' guidance on appropriate security measures should be taken into account. For more information about industrial security, please visit https://www.siemens.com/industrialsecurity.

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends to apply product updates as soon as available and to always use the latest product versions. Use of product versions that are no longer supported, and failure to apply latest updates may increase customer's exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed under https://www.siemens.com/industrialsecurity.

Npcap

OEM NOTE

The PRONETA installation package includes an OEM version of Npcap. This special version includes enterprise features such as the silent installer and commercial support as well as special license rights allowing customers to redistribute Npcap with their products.

Table of contents

1 Preface

Purpose of the Manual

"PRONETA" is a software suite for the analysis and configuration of PROFINET networks with special support for ET 200 distributed IO, consisting of the free component "PRONETA Basic", and the licensed product "PRONETA Professional", which adds extended functionality to PRONETA Basic.

This document explains the installation, purpose and use of PRONETA Professional. It complements the PRONETA Basic User Manual (see \4\).

Scope of this Documentation

This documentation describes the configuration and use of PRONETA Professional as an "MQTT API" client which serves a broker, and the use of its "IoT Hub API" with MQTT and AMQP communication.

It does not describe the subscription process for clients and the further processing of topics sent to the broker by PRONETA Professional, nor does it explain the installation process of the broker.

NOTE Additional information regarding the PRONETA Professional release will also be found in the "readme" file accompanying the software download.

This will especially concern last-minute changes and user information.

Required Basic Knowledge

This manual assumes the reader’s familiarity with the operation principles of PRONETA, which are explained in-depth in the PRONETA Basic User Manual (\4\). A thorough knowledge of the use and application of PRONETA is required to benefit to the greatest extent from the PRONETA Professional software and user manual.

Also, an understanding of basic MQTT operation principles is presumed.

Validity

This document is valid for the following software:

- PRONETA Professional V1.1 SP2
- PRONETA Basic V3.3 and higher

Syntax Conventions

`Highlighting text` is used in this manual for various elements:

- Literal commands the user has to type in,
- XML elements,
- MQTT topics and messages.

When an MQTT topic is enclosed in curly `{braces}`, it represents a variable name rather than a literal value.

Elements in `[square brackets]` are optional.

If, for example, the API configuration file (see 7.1) defines the value for `{Hello}` as:

```
<Hello>MyHello</Hello>
```

then the topic

```
SomeTopic/{Hello}
```

will actually refer to

```
SomeTopic/MyHello
```

2 Introduction to PRONETA Professional

2.1 Description

The PROFINET network analyzer PRONETA Basic is a free tool intended for the rapid analysis and configuration of PROFINET networks and the simple testing of ET 200 distributed I/O systems and other components.

PRONETA Professional is a licensed product (see 3.4) which extends the functionality of PRONETA Basic in several ways:

- It provides two extra "tasks" compared to the PRONETA Basic:
  - The "PROFIenergy Diagnostics" task, which allows you to monitor, evaluate and control the PROFIenergy function of devices which support this PROFINET profile (see 4).
  - The "Record Assistant" task, with which specific data records can be written and read to devices. It furthermore allows the user to analyze results and the Ethernet communication via Wireshark[1] (see 5).
- It comes with an API, based on MQTT, AMQP or HTTPS communication, which allows control of PRONETA's network scanning operation (see 6 and 7) and the running of third-party applications.
- It is possible to use the results of the new command line interfaces commands for further processing. The data is returned to a standard output.

As of Version 1.1, control of PRONETA Professional through the command line is also possible.

NOTE This manual supplements the PRONETA Basic User Manual, but does not replace it.

If you're unfamiliar with PRONETA, it is highly recommended that you consult the PRONETA Basic User Manual (\4\) in advance to get a general understanding for the operation of PRONETA.

2.2 General Data Protection Regulation (GDPR)

Siemens observes the principles of data protection, in particular the principle of data minimization (privacy by design). For this product, PRONETA Professional this means: The product does not process / store any personal data, only technical functional data (e.g. time stamp).

[1] Wireshark is a free and open-source packet analyzer. It must be downloaded independently from the PRONETA Professional distribution. See also \16\.

2.3 PRONETA Professional Communication APIs

PRONETA Professional extends PRONETA with the MQTT API (Message Queuing Telemetry Transport Application Programming Interface) and the IoT (Internet of Things) Hub API functionalities, and with a command line interface (CLI) which allows control of the PRONETA Professional service.

Purpose

The MQTT API, IoT Hub API and CLI API allow the control of PRONETA's functions without employing its graphical user interface. Instead, the PRONETA scanning functions can be invoked, controlled and the results gathered by an external application through the provided interface.

This is particularly useful as a mode of "remote control" for PRONETA, for example to perform an automated inventory of network nodes across many PROFINET cells.

2.3.1 The MQTT API

Principle of Operation: MQTT Communication

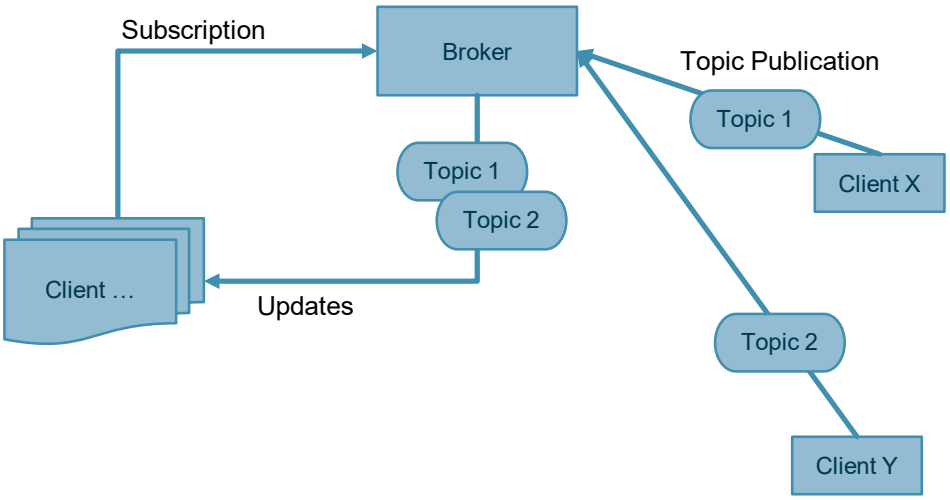
MQTT is a protocol on top of TCP/IP which requires only a small overhead of administrative traffic. This makes it well suited for limited bandwidth applications and embedded systems.

The basic operation of MQTT involves several "clients" which do not communicate directly with each other, but through the intervention of one or several "broker" processes. Data is "published" by the clients to the broker in the shape of "topics" which the broker collects and forwards to those clients which have "subscribed" to them. When two clients each subscribe to topics which are published by their respective counterparts, a two-way communication between clients can be established via topics. Clients receive updates on their subscribed topics immediately after they arrived at the broker.

The communication is "anonymous" in as far as the clients do not know which network nodes publish the topics they subscribe to. Likewise, the publishers do not know which clients are subscribed to their topics.

The topics themselves are organized in a hierarchical, tree-like structure. A client may subscribe to one particular topic, or to one topic and all its sub-topics. In the latter case, it will automatically receive updates about all sub-topics which lie below the subscribed topic in the hierarchy.

Figure 2-1: Basic MQTT operation: Clients X and Y publish their individual topics to the broker. The broker only passes topic updates to the clients that have subscribed to these topics with the broker.

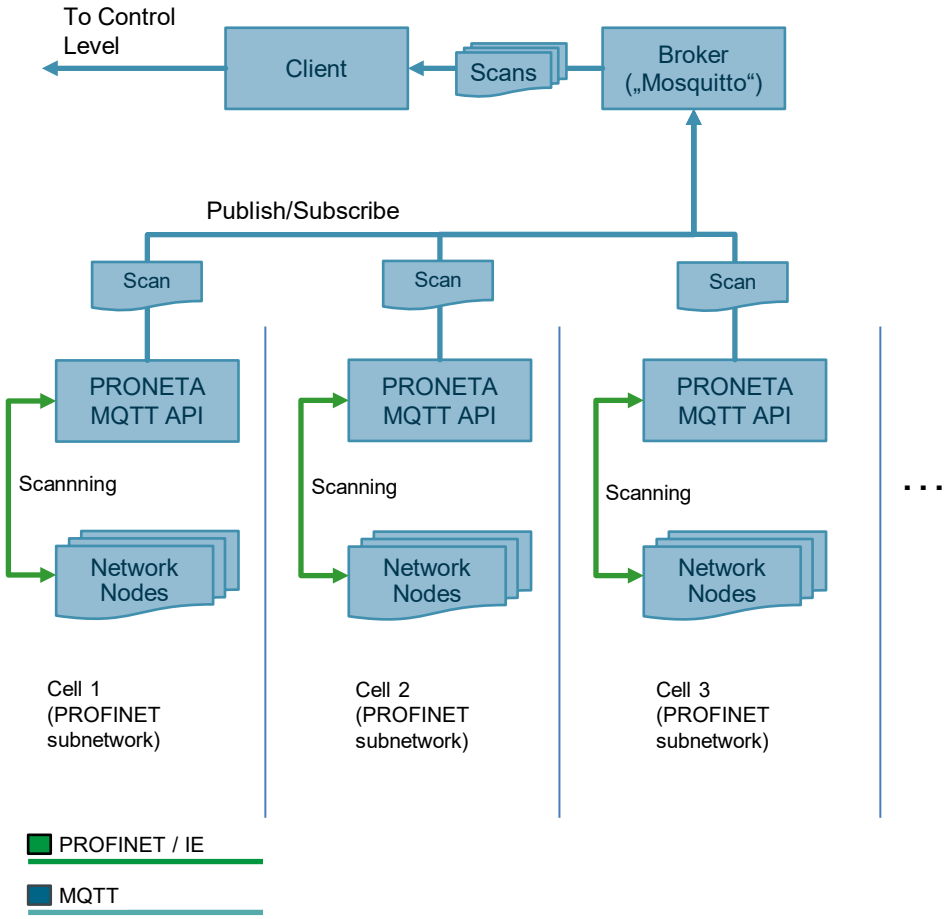


Topics can receive a "Quality of Service" assignment, so that a client will receive notification of an update at most once, exactly once, or possibly several times. While usually topics are only distributed when new updates are published, topics may also receive a "retention" flag. If a new client subscribes to such a topic, he receives a copy of the last topic value which has been stored with the broker immediately upon connection, and the client will not have to wait for the next publication.

MQTT Communication with the PRONETA MQTT API

The MQTT API is designed to be installed on several network "cells". The network scans from these cells are collected and passed on to a central client for further evaluation on the plant control level through MQTT communication.

Figure 2-2



Each cell consists of a PROFINET subnetwork with any number of PROFINET nodes, and exactly one PRONETA Professional service running. The individual services publish their network scans as topics to the broker, which in turn distributes them to the client or clients which have subscribed to them.

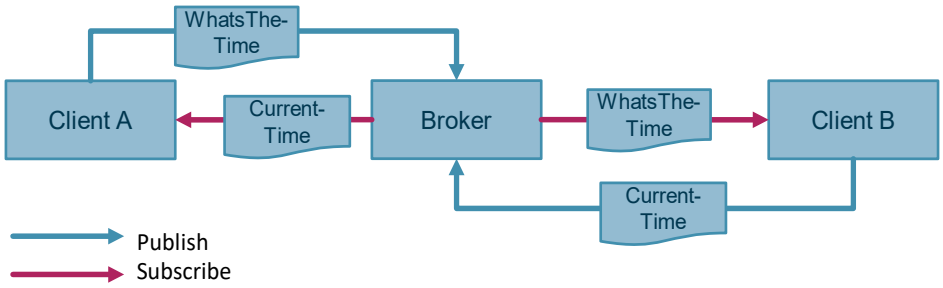
NOTE Having more than one scanner running in any PROFINET subnetwork will lead to communication conflicts.

Establishing a Two-way Communication between PRONETA Professional and other Clients

Per default, the communication on MQTT is one-way, with one client publishing a topic message, and one or several other clients receiving it. To establish a two-way communication between two clients via MQTT, it is necessary that each of the clients subscribes to the published topics of the other. Then one client can issue a query by publishing an update to its topic and then listening to the response when the second client updates its respective topic.

The following graphic shows an example for client A, who wants to receive the current time from client B. To this end, it publishes the topic `WhatsTheTime`, to which client B is subscribed. In return, client B publishes the information in `CurrentTime`, to which client A is subscribed. As always, the information is routed through the broker.

Figure 2-3



In principle, the two topics are independent from each other, and it is the responsibility of the clients to establish the connection between the topics.

2.3.2 The IoT Hub API

Purpose: Microsoft Azure and IoT Hubs

As of Version 1.0 Update 1, PRONETA Professional can connect with Microsoft Azure services using communication services provided by Azure's Internet of Things (IoT) Hub.

Azure is a cloud computing platform provided by Microsoft which provides a large number of services, namely "Infrastructure as a service", "Platform as a service", and "Software as a service." This means that Azure servers can for example be employed to run virtual machines or particular software applications for a local user. Furthermore, Azure provides "data centers" for cloud data storage as well as functions to analyze that data, and acts as a host for general websites and CRM systems.

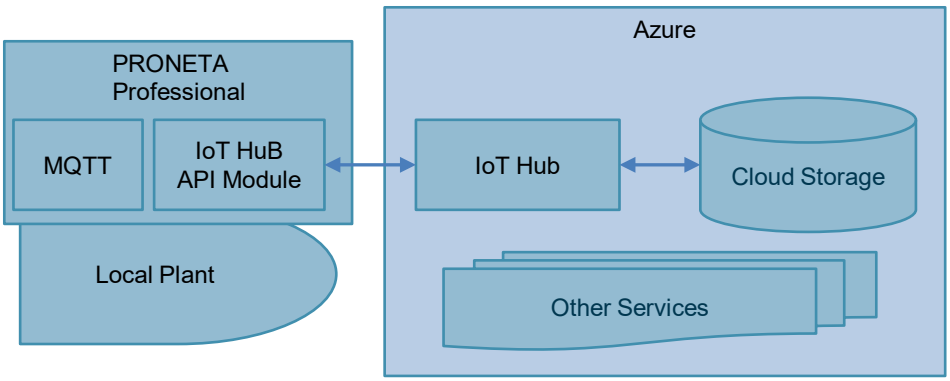
The Azure Internet of Things Hub is an Azure service designed to facilitate bidirectional communication with a large number of "small" interconnected digital devices, such as embedded systems in automation, "smart home" appliances, etc. IoT Hubs support among others the MQTT protocol.

The IoT Hub API establishes the communication between PRONETA Professional and an IoT Hub which is running as an Azure service. The IoT Hub in turn facilitates communication with Azure's Cloud Storage service.

Principle of Operation

Communication between PRONETA Professional and the Azure Cloud Storage works through a chain of elements: The IoT Hub API of PRONETA Professional controls the operation between PRONETA Professional and an IoT Hub. The IoT Hub is a service running in Azure which provides the communication between Azure's cloud storage and outside components.

Figure 2-4



Once the connection between the two is established, the IoT Hub API of PRONETA Professional is configured to listen to messages from the IoT Hub, and in response send replies and transmit files to the Azure storage.

2.3.3 The CLI API

As of Version 1.1 SP1, PRONETA Professional allows using of the new CLI commands for further processing. The data is returned to a standard output. If an error occurs, it is routed to a standard error output and the exit code is set. In case of missing acceptance of terms and conditions the exit code equals to 1, in case of any other error the exit code equals to 2.

2.4 WebSocket Support

As of Version 1.0 Update 1, PRONETA Professional also supports the use of the "BrokerAddress" WebSocket. This also includes using a proxy to communicate with the IoT Hub.

WebSocket communication differs from the HTTP1 protocol in some aspects. One important aspect of this is that HTTP1 relies on the client polling the server for new data. This process is highly resource-intensive, and it therefore throttled by the IoT Hub with possible pooling intervals of up to 25 minutes under certain conditions. In contrast, WebSocket communication ensures more economic and efficient push messages from the server.

NOTE WebSockets are only supported if PRONETA Professional is running under Windows 10 or Windows 11.

Using earlier Windows versions results in "Unsupported feature" errors of the .NET framework.

Alternatively, push messages can also be employed by using a proxy and the MQTT protocol. This technique only works under Windows 10.

3 Download and Installation

3.1 Scope of Delivery

The download from the PRONETA Professional website contains the following:

- PRONETA Professional V1.1 SP2 with extended functionality compared to PRONETA Basic (PROFIenergy diagnosis and Record Assistant)
- PRONETA Professional service software which performs the actual communication between a client and PRONETA
- The console application "PRONETA.API.SERVICE.exe" for installation and configuration of the PRONETA Professional service

3.2 System Requirements

PRONETA Professional

PRONETA Professional has the same system requirements as PRONETA Basic. See the chapter on the installation of PRONETA Basic in the respective manual (\4\) for details.

Windows

The following Windows versions are supported by PRONETA Professional:

- Windows 7, 32 and 64 bit
- Windows 10, 64 bit only
- Windows 11, 64-Bit

WebSockets are only supported when using Windows 10 and Windows 11 see 2.4.

3.3 Installation

Extract the downloaded zip archive into a directory on your PC.

3.4 Connecting of the Hardware Components

Connect the computer where PRONETA Professional runs with the network you want to analyze. Employ a regular PROFINET cable plugged into a suitable Ethernet interface.

3.5 Licensing

PRONETA Professional is a commercial, licensed product. Valid licenses are available from the SIEMENS Industry Mall. (\7\)

PRONETA Professional V1.1 license is required to use all available features. Without a valid license, it is not possible to start scan or run third-party applications. All other features are available (MQTT, IoT Hub and CLI APIs, remote configuration, presence notification and restart).

In order to use all features, the Automation License Manager (ALM) must be installed on the PC on which PRONETA Professional is to be run.

ALM V6.0 SP9 is part of the PRONETA Professional bundle in the \Misc\ALM\ folder.

If ALM is installed, a trial license is automatically generated upon the first "Start scanning" or "Run third-party application" request. The trial license remains active for 21 days and it is not possible to activate it again.

NOTICE	**Manipulating the system clock may destroy SIMATIC product licenses on your PC, including the PRONETA Professional license** Manually changing and resetting the system clock of the PC on which PRONETA Professional runs may result in the cancellation of product licenses installed on this machine. This may also happen, for example, if you travel with the PC through different time zones. Defective licenses can be repaired using the "Recover" function of the Automation License Manager (ALM) or the "Recover Assistant". Aid in license recovery can be obtained from Siemens Technical Support. (\17\)

An overview over the Siemens licensing concepts can be found in \18\.

3.6 Additional Components

Npcap

PRONETA Professional requires an API ("Application Programming Interface") to capture network traffic, e.g. Npcap. Npcap is part of the PRONETA distribution and is installed when PRONETA is first started if there is no Windows Packet Capture Library & Driver on the target PC.

NOTE	Depending on your distribution, the .NET Framework and Npcap drivers may not be included in the default installation for Windows Embedded, so you may require manual installation. The 32-bit version of Microsoft Visual C++ can also be used for 64-bit versions of Windows.

Broker

MQTT communication requires the services of a broker to enable the data transfer between publishing and subscribing clients. A broker must be properly installed and configured prior to the use of the PRONETA Professional MQTT API.

When communication security is enabled, PRONETA Professional can work with any compliant broker in TLS mode, supporting TLS V1.2.[2] For this reason, the broker component is not part of the distribution, and this document does not describe the installation and configuration of a particular broker.

PRONETA Professional has been extensively tested with the "Eclipse Mosquitto Open Source" broker.

[2] Encryption can be disabled, see 0

3.7 Firewall Configuration

Depending on the communication protocol used, PRONETA Professional requires that different ports must not be blocked by the firewall on the computers participating in the communication.

Table 3-1

Protocol	Port
MQTT (encrypted)	8883
MQTT via WebSockets	443
MQTT without encryption	1883
AMQP	5671
AMQP via WebSockets	443
HTTPS	443

In addition, tracing the PRONETA Professional communication requires port "1234" to be kept open for local connections. (See 6.6 for details.)

4 PROFIenergy Diagnostics

4.1 Overview

PROFIenergy is a PROFINET profile (\19\) to remotely supervise and monitor the energy consumption of devices, and to change between the states "regular operation", "standby" and "sleep".

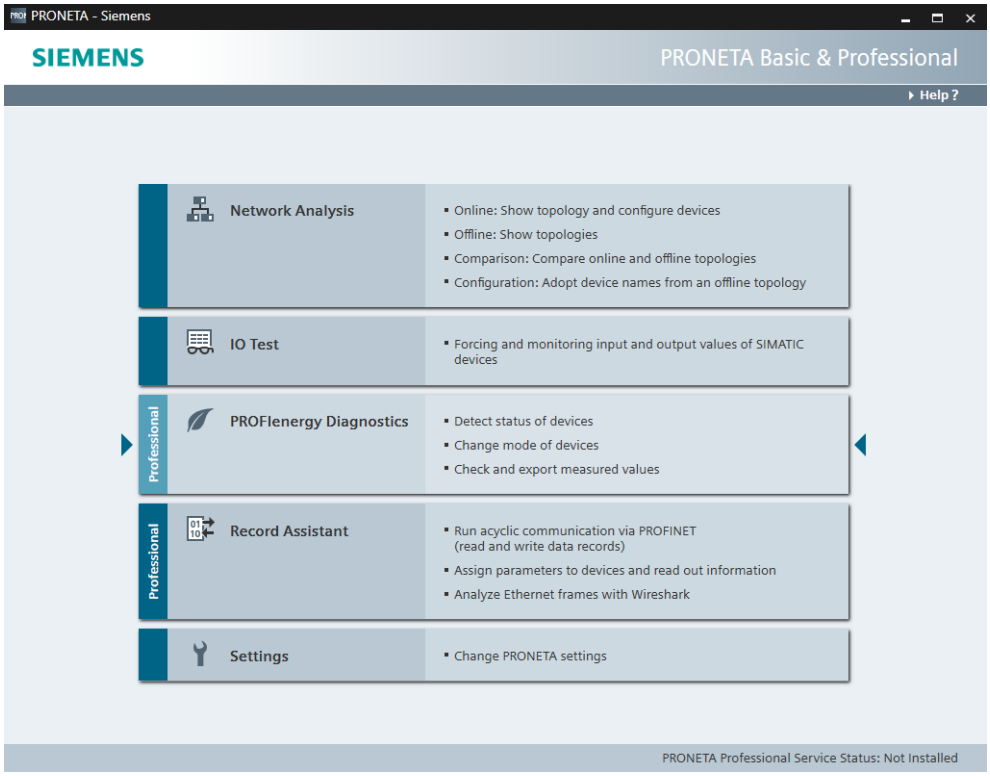
PRONETA Professional provides a task to the PRONETA user interface which allows to perform these operations on PROFIenergy-compliant devices found during a network scan.

4.2 Application

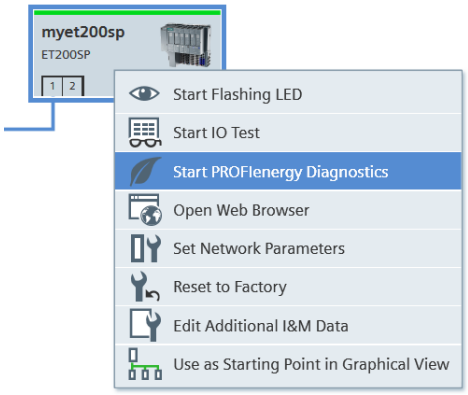
Start

PROFIenergy Diagnostics is launched from the PRONETA Professional home screen by clicking on the "PROFIenergy Diagnostics" panel.

Figure 4-1



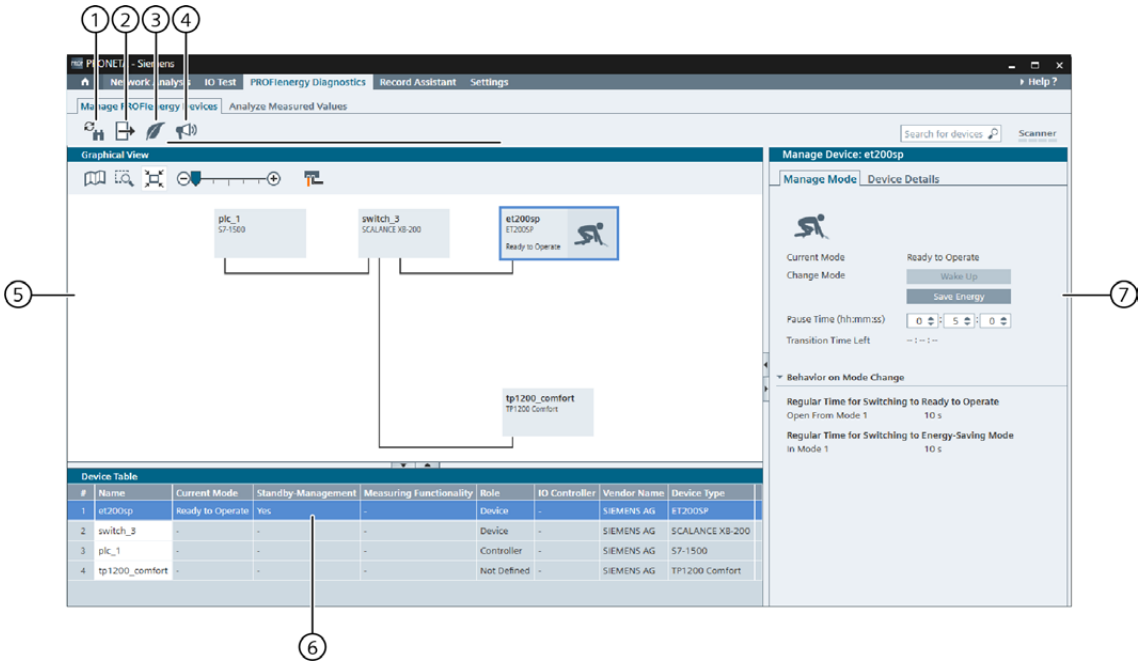
Alternatively, you can access PROFIenergy diagnostics from the Network Analysis Online mode. By right-clicking on any PROFIenergy-compliant device in a Graphical View or in the corresponding Device Table, you can bring up a context menu and choose the "Start PROFIenergy Diagnostics".



Operation

The main screen of the PROFIenergy task looks like this:

Figure 4-2



It provides two views, "Manage PROFIenergy Devices" and "Analyze Measured Values". The function bar at the top provides the following functions:

- A "Refresh" button (1), which starts a new network scan.
- An "Export Overview Information for All PROFIenergy Devices" button (2), which creates a file in CSV format with information about all PROFIenergy-compliant devices found.
- A "Switch on Management Mode" button (3), which enables the manual change of the operating mode of PROFIenergy-compliant devices (in Manage PROFIenergy Devices" view only).
- A "Wake Up a Device Using Its MAC Address" button (4), which allows to bring back a specific device to full operation (in "Manage PROFIenergy Devices" view only). The respective device is identified by its MAC address.

There are three panes with information about the devices connected to PROFINET:

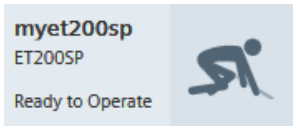
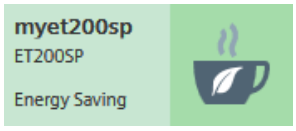
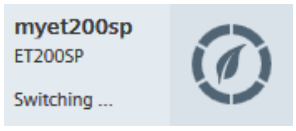
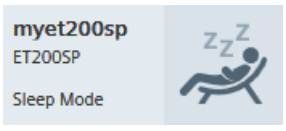
- The "Graphical View" (5) which contains a representation of the PROFINET network scan.
- The "Device Table" (6) with information about the devices found in the scan.
- The "Manage Device" window (7) with information about a device selected in the Graphical View or Device Table.

For information about the Graphical View and the Device Table, see the PRONETA Basic manual (\4\).

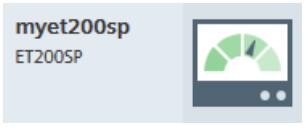
The "Manage Device" pane provides three sub-views:

- "Manage Mode" (if used from the "Manage PROFIenergy Devices" view)
  This mode allows changing the PROFIenergy operating mode of the device to "Energy Saving", "Sleep" or wake it up to "Ready to Operate".
  The Graphical View makes the device operating mode visible for PROFIenergy-compliant devices:

  Table 4-1

	Ready to Operate Regular operation status
	Energy Saving Energy saving status
	Switching Status change
	Sleep Mode Sleep mode status

- When the user switches to the "Analyze Measured Values" tab, the "Manage Mode" is replaced by "Current Values", which contains the measured values. Devices that support PROFIenergy measurement functionality are displayed in the Graphical View with a meter icon:



- "Device Details" tab shows the parameters of a particular device.

5 Record Assistant

5.1 Overview

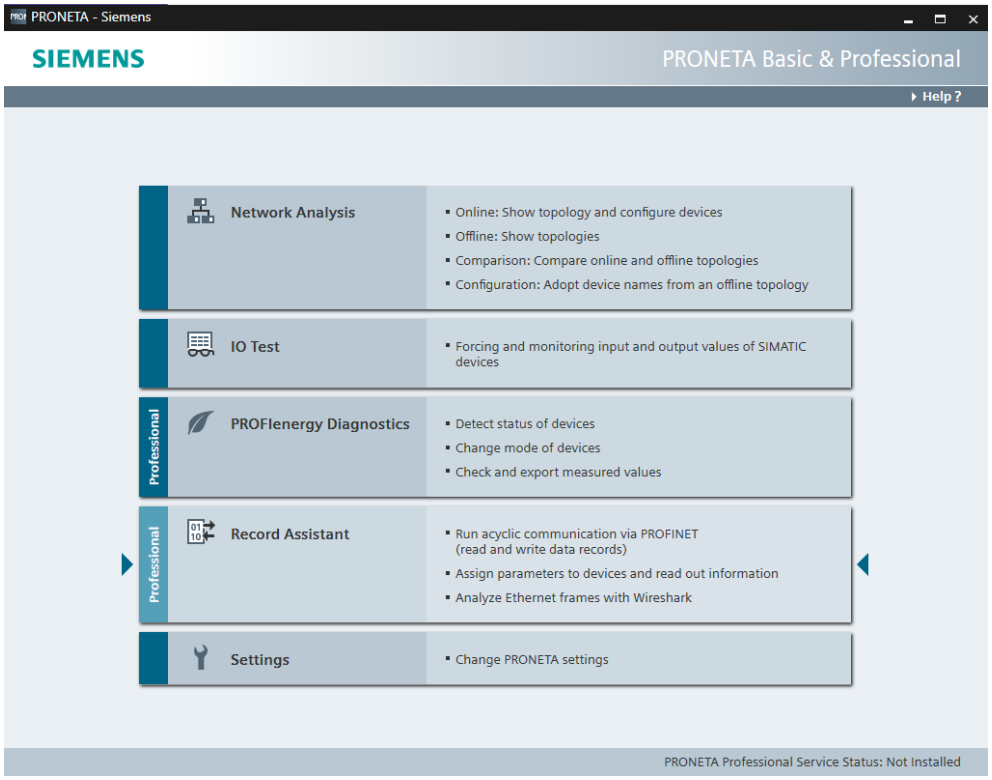
The Record Assistant offers a possibility to exchange record sets with PROFINET devices. If the Wireshark tracing application is installed (see \16\), it is also possible to perform a low-level analysis of Ethernet communication to the devices.

5.2 Application

Start

The Record Assistant is started from the PRONETA Professional Home screen by clicking on the "Record Assistant" panel.

Figure 5-1

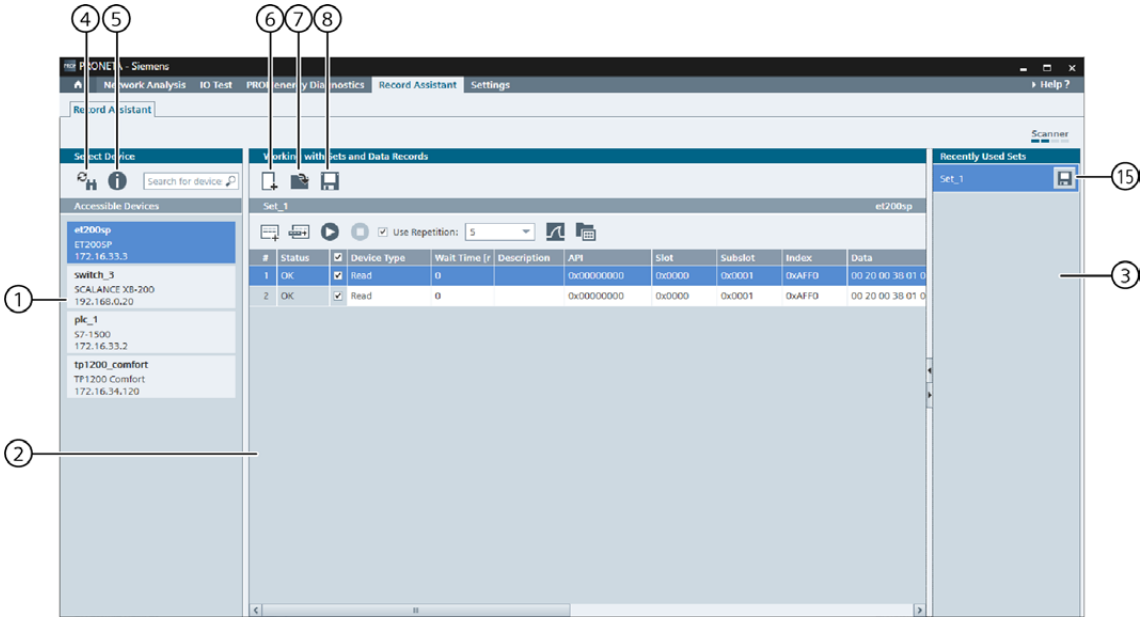


Operation

The purpose of the Record Assistant is to read and write "Sets" of Data Records from and to a device and evaluate the respective responses.

The main screen of the Record Assistant task looks like this:

Figure 5-2



It provides a view with three panes:

- The "Select Devices" (1) shows a list of devices found in the current PROFINET scan. A device with which communication is to be established can be selected here.
- The "Working with Sets and Data Records" (2) allows the creation, editing and execution of custom records (sets), a sequence of read/write operations to the selected device. Details like timeouts or a display of data results are included.
- The "Recently Used Sets" (3) shows a list of sets which have currently been employed.

In the function bar of the "Selected Device" pane, two functions are provided:

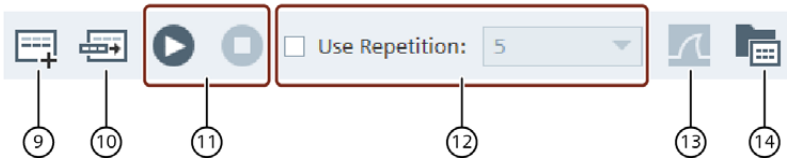
- The "Refresh Scan" (4) starts a new scan of the current PROFINET network.
- The "Show Device Details" (5) brings up a sub-view with details regarding the currently selected device.

In the function bar of the "Working with Sets and Data Records" pane, the following functions are provided:

- The "Create New Set" (6) opens a mask for a new set of records to be read and/or written.
- The "Open Set" (7) opens a previously stored set of records. (Set of records here means the assembly of read/write operations, not the data resulting from the transmissions.)
- The "Save Set" (8) saves the current set of records to disk.

The function bar for the currently open set contains the following functions:

Figure 5-3



- The "Add Data Record" (9) adds a new record to the currently open set.
- The "Import Data Records from Wireshark Capture" (10) imports a number of data records from a file to the current set. (The file format must be PCAP.)
- The "Start Acyclic Communication"/"Stop Process" (11) performs the read/write operations as defined in the list of records of the current set or cancels the processing.
- The "Use Repetition" (12) allows for the automated repetition of the read/write operations as defined in the set. The dropdown list next to the checkbox determines the number of repetitions.
- The "Open Current Recording in Wireshark" (13) displays the captured network traffic which occurred during the read/write operations of the set. (Only available if Wireshark is installed.)
- The "Open Folder with last Wireshark Recording" (14) opens a file explorer with the destination folder of the last recordings as default.

The (15) button in the "Recently Used Sets" pane enables the selected set to be saved on the disk.

6 Console Application "PRONETA.API.SERVICE.exe"

6.1 Overview

"PRONETA.API.SERVICE.exe" is a console application to install, uninstall, and restart the PRONETA Professional service, and to show its current status. This chapter explains the functionality and operation of the tool.

"PRONETA.API.SERVICE.exe" can be run in one of two modes:

- Interactive
- As a command line tool

Installation and uninstallation must be run with administrator privileges.

Interactive

For interactive mode, perform the following steps:

1. Go to your installation directory for PRONETA Professional, and right-click on the icon "PRONETA.API.SERVICE.exe".
2. From the context menu, select "Run as administrator".

The console application then starts in interactive mode with administrator privileges.

Command Line

The command line option is most useful to control the application remotely.

To use the command line tool from Windows, use the following procedure:

1. Open the Windows Start menu and enter `cmd` in the search field.
2. When the Windows command prompt application ("cmd.exe") is found in the list of applications, click on the entry. (For installation or uninstallation, right-click on the entry and in the drop-down menu, click on "Run as administrator".)
3. Change the current directory to your PRONETA Professional directory.

You can now run the application from the command prompt.

NOTE If there is a user typo, the command argument or mandatory argument is missing, the command fails with the corresponding error.

Getting Help

To get help, start the interactive mode, type `help` and hit return.

Quitting the Interactive Mode

To quit the interactive mode, type `exit` and hit return.

The console window closes.

6.2 Installation of the PRONETA Professional Service

General Procedure

To install the PRONETA Professional service on your computer, extract the files from the download archive to a directory of your choice.

Then chose one of the following procedures:

Interactive

1. Start the "PRONETA.API.SERVICE.exe" tool as administrator.
2. In the console window, enter the command `install` and hit return.
3. Confirm your agreement to the terms and conditions with `y` when prompted to do so.

NOTE The terms and conditions and security disclaimers can be found in the files "LicenseAgreement.txt" and "SecurityInformation.txt", resp., in your download archive.

Command Line

1. Open a Windows command shell in the installation directory, enter `PRONETA.API.SERVICE.exe -install -agreeWithTermsAndConditions` and hit return.

NOTES

- You must provide both options for the installation to succeed.
- Command line installation may be triggered remotely and result in the unwanted installation of PRONETA Professional service on a computer. Exercise this feature with caution.

Return Value

After running the installation, the tool exits with one of the following return values:

Table 6-1

Value	Meaning
0	Successful installation
1	Error: The user did not agree to the terms and conditions of the installation
2	Error: General unknown error

After the Installation

After a successful installation, the PRONETA Professional service will automatically start every time the PC is rebooted.

6.3 Uninstallation of the PRONETA Professional Service

To uninstall PRONETA Professional from your computer, chose one of the following procedures:

Interactive

1. Start the "PRONETA.API.SERVICE.exe" tool as administrator.
2. In the console window, enter the command `uninstall` and hit return.

Command Line

1. Open a Windows command shell in the installation directory, enter `PRONETA.API.SERVICE.exe -uninstall` and hit return.

This will stop and uninstall the PRONETA Professional service and keep it from being restarted automatically when the computer is rebooted. No files are deleted.

6.4 Restarting the PRONETA Professional Service

At times a restart of the PRONETA Professional service might be required. The application "PRONETA.API.SERVICE.exe" lets you initiate a service restart.

In interactive mode, enter the command `restart` and hit return.

The service will be restarted, and an appropriate message is printed to the screen.

6.5 Querying the PRONETA Professional Service Status

The application "PRONETA.API.SERVICE.exe" provides information whether the PRONETA Professional service is currently running.

In interactive mode, enter the command `status` and hit return.

A message regarding the current server status is printed to the screen. It contains the same information as is printed in the PRONETA Basic home screen ("PRONETA Professional Status in PRONETA Basic").

It is not possible to query the service status from the command line.

6.6 Tracing the PRONETA Professional Communication

To trace events during the operation of the PRONETA Professional service (configuration errors, API initialization, API command processing,…), start the interactive mode, type `listen` and hit return.

This will display information which is currently written into the PRONETA Professional log file (see 7.2.4). Tracing will stop as soon as the next keypress is performed.

NOTE Tracing employs port "1234" for communication on localhost address 127.0.0.1. This port must not be blocked by a firewall.

6.7 Dialogue Mode

In the interactive mode, it is possible to enter a "dialogue mode" to obtain live logging data while at the same time being able to enter new commands to the application.

To enter dialogue mode, enter the command `interactive` and hit return.

The usage can be easily obtained by typing `Help`. Detailed help with examples can be obtained by typing `Help (command)`.

For example:

```
Help Configure
Help ConfigureApplications
Help Dcp Reset
```

To leave dialogue mode, hit return with an empty command line or type `exit`.

As long as you are in the dialogue mode, the top part of the terminal window will be filled with a live stream of PRONETA Professional service log entries. In the bottom part of the window, you can enter a range of commands to control the service.

Available Command Summary

```
Configure (eg. Configure OutputFolder={target path})
StartScanning
SayHello
SendLog
SendConfiguration
Dcp set
Dcp reset
List adapters
List interfaces
Restart
RunApplication Id={ApplicationId}
ConfigureApplications Id={ApplicationId}
SendApplicationsConfiguration
ConfigureCredentials AppRunnerPassword=newPassword;
AppRunnerUserName=newUserName; AppRunnerDomain=newDomain
```

6.8 Service Control from CLI

By starting `PRONETA.API.SERVICE.exe` with command line arguments, it is possible to not only install, uninstall and restart the service, but also to control its operation. For example, the terminal command `PRONETA.API.SERVICE.exe --Configure OutputFolder="d:\anyPath\Results\"` will set the path used for service output data to `"d:\anyPath\Results\"`.

This CLI can be used when the command returns results, too. There are new commands possible that return errors on the standard error output and data on the standard output, such as DCP Set/Reset.

In most cases the data are returned as a formatted table that can be further automatically processed.

Available Commands Summary

```
PRONETA.API.SERVICE.exe --Configure OutputFolder={target path}
PRONETA.API.SERVICE.exe --StartScanning
PRONETA.API.SERVICE.exe --SayHello
PRONETA.API.SERVICE.exe --SendLog
PRONETA.API.SERVICE.exe --SendConfiguration
PRONETA.API.SERVICE.exe --Dcp set
PRONETA.API.SERVICE.exe --Dcp reset
PRONETA.API.SERVICE.exe --List adapters
PRONETA.API.SERVICE.exe --List interfaces
PRONETA.API.SERVICE.exe --Restart
PRONETA.API.SERVICE.exe --RunApplication Id={ApplicationId}
PRONETA.API.SERVICE.exe --ConfigureApplications Id={ApplicationId}
PRONETA.API.SERVICE.exe --SendApplicationsConfiguration
PRONETA.API.SERVICE.exe --ConfigureCredentials
AppRunnerPassword=newPassword; AppRunnerUserName=newUserName;
AppRunnerDomain=newDomain
```

7 PRONETA Professional Service

Installation and Uninstallation of the PRONETA Professional Service

Prior to use, the PRONETA Professional service must be properly installed on the host machine. To install and uninstall the PRONETA Professional service, see 4.

7.1 The "ApiConfiguration.xml" Configuration File

The PRONETA Professional operation is configured by means of the "ApiConfiguration.xml" file in the PRONETA Professional directory.

The XML format is human-readable, and the file can be adapted to your configuration either manually with an appropriate editor, or through MQTT or IoT Hub API access to the API service.

NOTE When applying changes to the PRONETA Professional configuration, see also 8 for troubleshooting hints.

Changes that take Effect Immediately and Required Restarts

Not all changes to the configuration take effect immediately. See the following list for details:

- Manual changes to "ApiConfiguration.xml" require a manually triggered restart of PRONETA Professional to go into effect.
- Changes applied through MQTT, IoT Hub API, Interactive and CLI take effect immediately.
- Changes to an API-specific part of the configuration automatically trigger a restart and reconnect of the service.
- Changes to non-API-specific parts of the configuration are applied immediately and do not require a restart.

7.1.1 Parameters in the "ApiConfiguration.xml" Configuration File

The "ApiConfiguration.xml" file contains both general configuration parameters and API-specific parameters.

The following parameters are included in the file:

Table 7-1

Element	Description
<ApiConfiguration>	Root element.
<ResultConfiguration>	Configuration of the resulting scans.
<LimitPerMac>	Maximum number of stored scans per network adapter. A value of "-1" means unlimited storage.
<OutputFolder>	Directory for the storage of the resulting scan files. The directory name must be given with an absolute or relative path.
<OutputFormats>	File format of the scan result: one or more of PNG, XML, and XPS.
<ScanConfiguration>	Configuring the network scanning process.
<NicMacs Mode= "Allowed|Blocked">	Mode for Network Interface Controller (NIC) MACs, either "Allowed" or "Blocked". This is followed by a list of actual MACs for network cards on the scanning computer.
<NicMac>	Entries for a list of NIC MACs. If NicMacs Mode="Allowed", then only NICs included in this list will be used in the scans. If NicMacs Mode="Blocked", then a NIC with this MAC will be blocked from the scans, and all network cards with MACs **not** included in the list will be used for the scans. If NicMacs Mode="Blocked" and the list of NicMACs is empty, then all interface cards will be employed for scans. See NIC MAC Address Format below for NIC MAC address syntax details.
<NicNames Mode= "Allowed|Blocked">	Mode for NICs, either "Allowed" or "Blocked". Its use is similar to <NicMacs…> above, but it is followed by a collection of network adapter **names** on the scanning computer, rather than MAC addresses. See NicName Collections below for the format of NIC names.
<NicName>	Entries for a list of NIC names. Use is analogous to <NicMac>. Mixed use of MACs and adapter names is possible.
<AutomaticNameAssignment>	Shall scanned network modules without a name automatically receive a temporary name? Either "false" or "true".
<AutomaticIpAssignment>	Shall scanned network modules without an IP address automatically receive a temporary IP? Either "false" or "true".
<AutomaticIpAddress>	Subnet base address from which IP addresses will be assigned to network modules during automatic IP assignment.
<AutomaticSubnetMask>	Subnet Mask which will determine the address range from which IP addresses will be taken during automatic IP assignment.

Element	Description
<SendEpmToControllers>	Shall modules be requested to return I&M data? From V1.1 SP1 is deprecated, and this value is ignored (set internally to "false").
<ScanIpRange>	Shall the network scan for non-PROFINET devices be limited to a certain range of IP addresses? Either "false" or "true".[3]
<LowerIpLimit>	Lower bound of the IP address range for scans for non-PROFINET devices.
<UpperIpLimit>	Upper bound of the IP address range for scans for non-PROFINET devices.
<ScanSubnets>	Allow scanning of subnets. Either "false" or "true". Default = "false".
<Subnets>	Example: <Subnet IsActive="false" IpAddress="..." SubnetMask="..." GatewayIpAddress="1..."/>.
<OutputMaxTraffic>	Maximum baud rate with which the scanner operates on the network. Value between 125,000 and 12,500,000 Byte/second.
<WaitForScan>	Maximum wait time for the scanner to be available, in seconds. Default = 60.
<ReadDiagnosticData>	Read diagnostic data. Either "false" or "true". Default = "false".
<ReadProfiEnergyData>	Read PROFIenergy data. Either "false" or "true". Default = "true".
<ReadAssetManagementData>	Read asset management data. Either "false" or "true". Default = "false".
<ReadCentralIoModulesData>	Read central IO modules data. Either "false" or "true". Default = "false".
<ReadFiberOpticData>	Read fiber optics data. Either "false" or "true". Default = "true".
<ApiSpecificConfigurations>	Sections for various module-specific configurations.
<ApiSpecificConfiguration Key= {moduleName} Disabled= "false|true">	Shall the module {moduleName} of PRONETA Professional be disabled? Either "false" or "true". Example: <ApiSpecificConfiguration Key= "Mqtt" Disabled= "true">
<Value xsi:type= "MqttConfiguration">	Following are configuration settings for the MQTT API broker.
<BrokerAddress>	IP address of the broker to which the client subscribes. If the address begins with wss:// or ws://, a WebSocket connection will be used to communicate with this broker (Windows 10 or Windows 11 only).
<BrokerPort>	Port of the broker.
<CoreTopic>	Base name of the topic for communication between the PRONETA Professional scanner and the receiver of the scans.
<Cell>	Name of the cell the client is scanning which is passed to the MQTT broker as client ID. The name must be unique. If empty, the client PCs name is used.
<DisableSecurity>	Disables encrypted communication, see 7.1.7 for details.

[3] In V1.1 SP2, the internal switch to a local subnet is used. This could result in scanning a wider range of IP addresses defined by the Lower and Upper limits.

Element	Description
<Group>	Sub topic to address groups within the cell.
<Hello>	Sub topic for handshake messages between PRONETA Professional and receiver clients.
<CaCertificatePath>	Absolute or relative path to a CA.crt certificate that is used to authenticate client/broker communication.
<Value xsi:type= "IoTHubConfiguration">	Following are configuration settings for the IoT Hub.
<Protocol>	Protocol for the communication between the IoT Hub API and the IoT Hub itself. One of "Amqp", "Amqp_WebSocket_Only", "Amqp_Tcp_Only", "Mqtt", "Mqtt_WebSocket_Only", "Mqtt_Tcp_only", or "Http1". (See 3.7 for required Windows firewall settings.)
<Http1PollingInterval>	Polling interval in seconds between queries for message updates. Depending on the size of the IoT solution, values of up to 1500 may be required.
<ProxyConfiguration>	If the element is set to <ProxyConfiguration />, then no proxy is used for Azure/MQTT broker communication.
<Address>	Address of the proxy. If left empty, no proxy is used.
<Domain />	Domain or name of the computer which will verify the credentials.
<BypassProxyOnLocal>	Should the proxy not be used when requesting local resources? Either "false" or "true".
<UseDefaultCredentials>	Use the credentials of the user which is currently logged on to identify yourself at the proxy? Either "false" or "true".

NIC MAC Address Format

NIC MAC addresses consist of five pairs of two hexadecimal digits (0-9, a-f) each. The pairs can be separated from each other through a colon ":" or a hyphen "-". The characters "a" to "f" are not case-sensitive: the addresses "01:ab:cd:89:ef" and "01-AB-CD-89-EF" are equivalent.

7.1.2 Changing the Configuration

In addition to manual changes to the API configuration file, it is also possible to read and write the various configuration parameters by using MQTT, IoT Hub communication, or through the CLI API. This is useful to configure PRONETA Professional remotely.

NOTES

- The "ApiConfiguration.xml" file is not secured against unauthorized access from a third party which may result in a corrupt configuration file. With a corrupt configuration file, PRONETA Professional will no longer run properly.
- The MQTT API can only change MQTT-specific parameters (<Broker>, <Cell>,...) and general parameters of the configuration.
- The IoT Hub API can only change IoT Hub-specific parameters (<Protocol>, <Http1PollingInterval>,...) and general parameters of the configuration.
- Messages received through the CLI API can only change general parameters of the configuration.

When valid changes are sent to the XML file regarding group or API level topics via MQTT, all API clients will be reinitialized immediately with the new configuration parameters.

MQTT Configuration Messages

PRONETA Professional is subscribed to the following topics to listen for configuration messages:

```
{CoreTopic}/Config
{CoreTopic}/{Group}/Config
{CoreTopic}/{Cell}/Config
```

Messages sent to any of these topics can change the API configuration in one of two ways:

- If the message contains a valid XML file, this file is used as the new "ApiConfiguration.xml".
- If the message contains key/value pairs, individual parameters are set to the respective keys. (In `<ResultConfiguration>` and `<ScanConfiguration>` only.)

IoT Hub Configuration Messages

IoT Hub message:

```
Properties      : Command = Configure
Message body    : key1=value1 key2=”value 2”
```

Examples of Configuration Messages

The key/value pairs have to be separated from each other by a semi-colon `;` in the message. Keys and values have to be separated by an equal sign `=`. If the values consist of a list of items, these have to be separated by commas `,` from each other.

If the value contains whitespaces, it is necessary to use "key"="value with spaces".

The following are a few examples for messages to change configuration parameters:

- `OutputFolder = Results\` will set the directory for scan storage to "Results".
- `OutputFormats = XML,PNG,XPS` will set the possible scan file formats to XML, PNG and XPS.
- `AutomaticNameAssignment = false; AutomaticIpAssignment = false; AutomaticIpAddress = 192.168.0.0; AutomaticSubnetMask = 255.255.255.0` will update the configuration for automatic naming of components accordingly.

Special Topics for the Configuration

To avoid the transmission of complete variable lists, there are also several topics which allow the addition or removal of one or more individual values to and from the lists.

- `AddOutputFormat` and `RemoveOutputFormat` allow the addition or removal of individual entries from the list of scan result file formats.
- `AddNicMacs`, `RemoveNicMacs` and `AddNicNames`, `RemoveNicNames` add or delete individual adapters to the list of NICs which are used for network scans, depending on their MAC addresses or names.

Changing the MQTT and Proxy Configurations via MQTT

The MQTT and proxy configurations can also be changed by sending MQTT messages to PRONETA Professional.

The following are schematic examples which will change the broker configuration through messages sent via Mosquitto:

```
mosquitto_pub.exe -t Proneta/Api/DEFAULTGRP/Config -m
"BrokerAddress= 192.168.0.1"
mosquitto_pub.exe -t Proneta/Api/DEFAULTGRP/Config -m
"BrokerAddress= www.broker.com"
```

will set the address of the broker to "192.168.0.1" or "www.broker.com", respectively. For TLS-based communication (see 7.6.1) these examples will need to be expanded with the path to the certificate directory and a broker name.

```
mosquitto_pub.exe -t Proneta/Api/DEFAULTGRP/Config -m "Hello=
HelloFromPronetaProfessional"
```

sets the sub-topic to which the MQTT API will send "Hello" messages when connected to the broker.

The following example will change the proxy configuration through messages sent via Mosquitto:

```
mosquitto_pub.exe -t Proneta/Api/DEFAULTGRP/Config -m "Address=
192.168.1.1"
```

will change the address of the proxy to "192.128.1.1". To disable the proxy, use an empty value for the proxy address: `... "Address=" ...`

For a list of parameters, see 7.1.1.

7.1.3 Reading the Configuration via MQTT

The PRONETA Professional service is subscribed to the following topics to listen for requests to submit its configuration:

```
{CoreTopic}/ConfigStatus
{CoreTopic}/{Group}/ConfigStatus
{CoreTopic}/{Cell}/ConfigStatus
```

When an empty message is sent to one of these topics, PRONETA Professional in response publishes a message to the topic

```
{CoreTopic}/{Cell}/Config/Result
```

This message contains the serialized current API configuration as a UTF-8-coded string which can be further evaluated by the client.

In addition, a message with the content "OK" will be sent to the topic

```
{CoreTopic}/{Cell}/Config/Status
```

to indicate a successful transmission.

7.1.4 Reading the Configuration via IoT Hub

Send a JSON message from IoT Hub to PRONETA Professional.

IoT Hub message:

```
Properties: Command = SendConfiguration
Message body :
```

The status message configming successfull configuration upload will be:

```
{
"Properties":{"Topic":"Config-Status"},
"Body":{"Status":"OK","Message":"Api configuration published."}
}
```

7.1.5 Testing the PRONETA Professional Status

To test if the PRONETA Professional Service is properly running, a client can publish the following topics (optionally with empty content):

```
{CoreTopic}/SayHello
{CoreTopic}/{Group}/SayHello
{CoreTopic}/{Cell}/SayHello
```

If the PRONETA Professional service is installed properly and the MQTT API is running, the service responds by publishing messages with the content "Hello" to the following topics:

```
{CoreTopic}/{Hello}
{CoreTopic}/{Group}/{Hello}
{CoreTopic}/{Cell}/{Hello}
```

The `{Hello}` topic can be configured through the API configuration file (see 7.1.1)

A message with content "Hello published" will also be sent to the following topic:

```
{CoreTopic}/{Cell}/Api/Status
```

The client can subscribe to one of these topics to verify a successful two-way communication with the MQTT API via the configured broker.

The same can be used for IoT Hub API.

Send a JSON message from IoT Hub to PRONETA Professional.

IoT Hub message:

```
Properties      : Command = SayHello
Message body    :
```

The reply will be:

```
{
   "Properties":{"Topic":"Hello"},
   "Body":Hello
}
```

And the status message that Hello message was successfully published:

```
{
   "Properties":{"Topic":"Api-Status"},
   "Body":{"Status":"OK","Message":"Hello published."}
}
```

7.1.6 NicName Collections

NicNames

The `ScanConfiguration` section of the configuration file contains the `NicNames` Element. This element contains a list of Network adapter names that will be allowed / blocked during scanning.

```
<ScanConfiguration>
  <NicMacs Mode="Blocked" />
  <!--Collection of network adapter names.
  <NicName>VMware Network Adapter VMnet1</NicName>-->
    <NicNames Mode="Blocked">
      <NicName>Local Area Connection* 1</NicName>
      <NicName>VMware Network Adapter VMnet1</NicName>
      <NicName>Cisco Systems VPN Adapter</NicName>
    </NicNames>
</ScanConfiguration>
```

Adapters listed in Mode="Blocked" will be barred from scanning, adapters listed in Mode="Allowed" will be used for scans.

The content of NicMacs and NicNames collections (see below) can be used to create well-defined NIC filters. For example, it is possible to allow scanning of all available NICs by MAC, except those that point to corporate networks, and to block scanning on known virtual adapters by their names.

Changing a NicNames Collection

It is possible to change the contents of a NicNames collection in the following ways:

- MQTT message:

```
mosquitto_pub.exe -t Proneta/Api/{cellName}/Config -m "NicNames =
[Blocked|Allowed],"Adapter Name1",["Adapter Name2"]"
mosquitto_pub.exe -t Proneta/Api/{cellName}/Config -m
"AddNicNames = "Adapter Name1",["Adapter Name2"]"
mosquitto_pub.exe -t Proneta/Api/{cellName}/Config -m
"RemoveNicNames = "Adapter Name1",["Adapter Name2"]"
```

- IoT Hub message:

```
Properties   : Command = Configure
Message body : NicNames = [Blocked|Allowed],"Adapter
Name1",["Adapter Name2"]
Properties   : Command = Configure
Message body : AddNicNames = "Adapter Name1",["Adapter Name2"]
Properties   : Command = Configure
Message body : RemoveNicNames = "Adapter Name1",["Adapter Name2"]
```

- Interactive mode:

```
Command>Configure NicNames = [Blocked|Allowed],"Adapter
Name1",["Adapter Name2"]
Command>Configure AddNicNames = "Adapter Name1",["Adapter Name2"]
Command>Configure RemoveNicNames = "Adapter Name1",["Adapter
Name2"]
```

- CLI:

```
PRONETA.API.SERVICE.exe --Configure NicNames =
[Blocked|Allowed],"Adapter Name1",["Adapter Name2"]
PRONETA.API.SERVICE.exe --Configure AddNicNames = "Adapter
Name1",["Adapter Name2"]
PRONETA.API.SERVICE.exe --Configure RemoveNicNames = "Adapter
Name1",["Adapter Name2"]
```

7.1.7 Disabling Security

If the MqttConfiguration element `DisableSecurity` is set to "True", the MQTT encryption is disabled, and the CA certificate of server will not be verified. This usually also requires changing the MQTT port to 1883. (This port is the default for unencrypted communication on many brokers.)

```
<ApiSpecificConfiguration Key="Mqtt" Disabled="false">
  <Value xsi:type="MqttConfiguration">
…
<!--Disable encryption of communication (Default = false). All
communication will be unencrypted when set to true!-->
    <DisableSecurity>false</DisableSecurity>
    …
  </Value>
</ApiSpecificConfiguration>
```

NOTICE	**Disabling data transmission security poses a great security risk** Do not use this option in a production environment, since it may result in your data security being compromised, leading to third-party interference with your plant and data.

It is recommended to use this option only in combination with the MQTT broker running on Localhost during development and testing. This option provides an easy approach to get familiar with PRONETA Professional features without the generation of broker certificates.

7.2 General Operation

Unless otherwise noted, this information is valid for both the MQTT and the IoT Hub API. If the APIs work differently, this will be specifically pointed out.

7.2.1 Network Scan

Automatic scanning of networks is the main application of the PRONETA Professional service. Network scans can be generated in one of three modes:

- Single-cell
- Group of cells (MQTT API only)
- All connected cells (MQTT API only)

The scan is initiated by the client, which publishes the corresponding topics. The API listens to these and starts scanning as soon as a message is received. When the scan is complete, the API publishes the scan results (see 7.2.2).

NOTE Depending on the size of the scanned network and available bandwidth, there can be a notable delay between commencement of the scan, and publication of the results.

Scan of a Single Cell

To initiate the scan of a single network cell, send the message `scanner=start` to the topic `{CoreTopic}/{Cell}/Scanner/Operating`, where `{CoreTopic}` and `{Cell}` were defined in the API configuration file (see 7.1), and `{Cell}` contains the name of the cell to be scanned

If `{Cell}` is undefined, it defaults to the PC's name.

Scan of Group of Cells

To initiate the scan of a group of cells, send the message `scanner=start` to the topic `{CoreTopic}/{Group}/Scanner/Operating`, where `{CoreTopic}` and `{Group}` were defined in the API configuration file.

NOTE This feature is only available with MQTT API.

Scan of all Cells

To initiate the scan of all connected cells, send the message `scanner=start` to the topic `{CoreTopic}/Scanner/Operating`, where `{CoreTopic}` was defined in the API configuration file. This is not possible via IoT Hub API.

NOTE This feature is available only with MQTT API. Scanning all cells can incur a high network load if many PRONETA Professional instances are running in the network.

Start scanning

MQTT Message:

```
mosquitto_pub.exe -t Proneta/Api/{cellName}/Scanner/Operating -m "
scanner=start"
```

IoT Hub message:

```
Properties      : Command = StartScanning
Message body    :
```

Interactive mode:

```
Command>StartScanning
```

CLI:

```
Proneta.Api.Service.exe --StartScanning
```

Stop the running scan

MQTT message:

```
mosquitto_pub.exe -t Proneta/Api/{cellName}/Scanner/Operating -m "
scanner=stop"
```

IoT Hub message:

```
Properties      : Command = StopScanning
Message body    :
```

Interactive mode:

```
Command>StopScanning
```

CLI:

```
Proneta.Api.Service.exe --StopScanning
```

7.2.2 Scan Results

After a scan is performed, the API publishes the results for the scanned cells in a series of topics. The exact name of the topic depends on the configuration of the API, namely group and cell names, as well as the supported scan format (see 7.1). The client must have subscribed to these topics to receive the corresponding messages.

Formats and Topics

For the scan results, three different formats can be configured, which will be sent to different topics:

Table 7-2

Format	Topic	Scan results
XML	`{CoreTopic}/{Cell}/Scanner/Result`	Textual representation of the topology
PNG	`{CoreTopic}/{Cell}/Scanner/Result/PNG`	Image, sent as a byte array. Bitmap (PNG) and/or vector (XPS) representation.
XPS	`{CoreTopic}/{Cell}/Scanner/Result/XPS`	

If a particular file format is not configured in the API configuration, no messages for these topics will be sent after the scan is finished.

NOTE

- If several network adapters are used in scanning, the results of each adapter's scan are published before the next adapter begins scanning.
- For large networks, the standard Windows XPS viewer will no longer render the scanned topologies correctly. In this case, use a different viewer (e.g. the PDF-XChange viewer).

Contents

For an XML file, the contents of the scan are:

- Topology of the scanned network (devices, connections)
- I&M ("Identification and maintenance") data (if provided by the device)
- AMR (Asset Management Record) data (if provided by the device)
- Information about the scanning PC

There is also optional content that can be read if the appropriate option is set in "ApiConfiguration.xml", such as:

- ReadDiagnosticData (Default= false)
- ReadProfiEnergyData (Default= true)
- ReadAssetManagementData (Default= false)
- ReadCentralIoModulesData (Default= false)
- ReadFiberOpticData (Default= true)

Import of Results to PRONETA

XML files from PRONETA Professional network scans can be imported in PRONETA Basic, like the scans created by PRONETA Basic itself.

See the PRONETA Basic User Manual (\4\) for more details.

Local Scan Storage

The results of a particular scan are not only transmitted via the API, but also stored locally on the file system of the scanning PC.

Per default, scan results are stored in the "\Results" subdirectory of the PRONETA Professional directory. The structure of an automatically generated scan file name is:

```
topology_{mac}_{yyyy}-{MM}-{dd}_{hh}-{mm}-{ss}.{filetype}
```

It contains the MAC address of the network adapter used, date and type of the scan, and the format of the file.

Local scan results can be configured in the "ApiConfiguration.xml" file. The relevant parameters are:

- `<OutputFolder>PathToFolder</OutputFolder>`
  sets the storage location for the resulting files to "PathToFolder".
- `<LimitPerMac>n</LimitPerMac>`
  sets the amount of scans stored per network adapter to `n`.
  If already `n` scans are stored, the next scan will result in the deletion of the oldest stored scan.
  If a value of "-1" is used for `n`, the number of stored scans is unlimited.

If a particular file format is not configured in the API configuration, no files are created for that format.

7.2.3 Status Information

PRONETA Professional provides constantly updated information about its status to a number of topics.

For MQTT API:

Table 7-3

Topic	Content
`{CoreTopic}/{Cell}/Api/Status`	Various
`{CoreTopic}/{Cell}/Config/Status`	Various
`{CoreTopic}/{Cell}/Log/Status`	Log-related
`{CoreTopic}/{Cell}/Scanner/Status`	Scanner-related

A client interested in the current status can subscribe to these topics.

Each `Status` topic contains an XML file with two elements: a `<Status>` element, which gives a classification of the current state, and a `<Message>` element, which contains a plaintext description of the state.

IoT Hub topics have a JSON format:

Table 7-4

Topic	Content
`Api-Status`	Various
`Config-Status`	Various
`Log-Status`	Log-related
`Scanner-Status`	Scanner-related

Example of an IoT Hub message response:

```
{
   "Properties":{"Topic":"Api-Status"},
   "Body":{"Status":"OK","Message":"Hello published."}
}
```

Possible States in the Logs

The `<Status>` elements can have one of the three following values:

- OK
- WARN
- ERROR

Status Messages

The different topics can contain the following messages for the respective status values:

Table 7-5: "API" topic

Status	Message
OK	API configuration updated.
OK	"Hello" published.
ERROR	Could not export Topology to file: {0}
ERROR	No topology for MAC {0} available in the scan result object.
ERROR	Any error caused publishing "Hello" message

Table 7-6: "Config" topic

Staus	Message
OK	API configuration published
ERROR	Any error caused by deserialization
ERROR	Any error caused by serialization
ERROR	Any error caused by configurating of API with new parameters
ERROR	Any error caused by providing empty configuration parameters
ERROR	Any error caused by setting API configuration to clients
ERROR	Any error caused by saving API configuration

Table 7-7: "Log" topic

State	Message
OK	Log published
ERROR	Any error caused log file not found
ERROR	Any error caused publishing log file

Table 7-8: "Scan" topic

State	Message
OK	Starting Scanner.
OK	Scan finished.
OK	Scan finished for MAC address {0}.
WARN	Scanning skipped, no active adapters that respect MacCollection.
WARN	Network scan was blocked by another running scan.
WARN	Scanning skipped; network scanner manager does not allow to start scan.
WARN	Adapter {0} skipped, network not available.
WARN	Stop of service network scan has been requested.
ERROR	Scanner is already running, please try the start scan later.
ERROR	Not supported scanner command parameter {0} - only Operating is supported

7.2.4 Tracing and Error Log

Local Error Log

PRONETA Professional keeps a log of error messages and traces.

The log is stored in the "\Logs" subdirectory of the PRONETA Professional directory in a file called "ApiLog.log" and can be viewed with any text editor.

The tool "PRONETA.API.SERVICE.exe" also allows an online trace of the log (see 6.6).

Error Log Distribution

A client can request log file transfer from PRONETA Professional PC in several ways.

Using MQTT

The PRONET Professional service is subscribed to the following topics:

```
{CoreTopic}/Log
{CoreTopic}/{Group}/Log
{CoreTopic}/{Cell}/Log
```

Whenever a message is received for one of these topics (the message may be empty), the PRONET Professional publishes the contents of the current log to the topic in response

```
{CoreTopic}/{Cell}/Log/Result
```

as a UTF-8 encoded string. It will also send an "OK" status message to

```
{CoreTopic}/{Cell}/Log/Status
```

Using the IoT Hub API

```
Command = SendLog
```

Using the CLI

```
PRONETA.API.SERVICE.exe --SendLog
```

Using the dialogue mode

```
Command>SendLog
```

7.2.5 Restarting the PRONETA Professional

A restart of the PRONETA Professional service can be triggered in several ways:

Using MQTT

Send an empty message to one of the following topics:

```
{CoreTopic}/Restart
{CoreTopic}/{Group}/Restart
{CoreTopic}/{Cell}/Restart
```

PRONETA Professional is subscribed to these topics and will perform a restart when it receives a message to any of them. The service does not respond with an acknowledgement message.

For example, if a Mosquitto broker is running, the following command line would send a restart command to PRONETA Professional:

```
Mosquitto_pub.exe -t Proneta/Api/DEFAULTGRP/Restart -m ””
```

Using IoT Hub Messages

Send a message with the following property and an empty message body to the IoT hub connected to the PRONETA Professional service:

```
Command = Restart
```

PRONETA Professional will perform a restart when it receives this message. The service does not respond with an acknowledgement message.

Using the CLI

Enter the command:

```
PRONETA.API.SERVICE.exe --SendLog
```

Using the dialogue mode

Send the command:

```
Command>Restart
```

7.3 Comparison functionality

Comparison functionality is part of PRONETA Professional and requires a valid license. It allows to programmatically compare two topologies and get results regardless of whether they are different or not.

If they do not differ, it is possible to display the differences automatically in the PRONETA GUI. The result of the comparison is available as a return code from the application call.

Table 7-9: Return codes

Return code	Meaning	Standard output	Error output
0	Topologies are equal		
1	Error in command processing		Error message
2	Topologies are not equal		Comparison result

The functionality is available by calling PRONETA.exe with the correct arguments. It is not available by calling PRONETA.API.SERVICE.exe as it is currently related to GUI only. The syntax of the command differs depending on the use case.

- Comparison of two offline topologies:

```
PRONETA.exe --compare <File1={}> <File2={}> [ShowGui={false}]
[Connections={true}] [FwVersion={true}] [Modules={true}]
[Name={true}] [OrderNumber={true}] [IpAddress={true}]
[IpRange={192.168.0.0,192.168.0.255}]
```

- Comparison of offline and online topology:

```
PRONETA.exe --compare <File1={}> <Nic={}> [ShowGui={false}]
[Connections={true}] [FwVersion={true}] [Modules={true}]
[Name={true}] [OrderNumber={true}] [IpAddress={true}]
[IpRange={192.168.0.0,192.168.0.255}]
```

<> means mandatory argument
[] means optional argument
{} means the default value of the argument

Table 7-10: All arguments

Argument name	Function
File1	Path to the first topology XML.
File2	Path to the second topology XML.
Nic	Identifier of the network adapter. Could be the MAC address, the name of the adapter name or its GUID.
ShowGui	If the true PRONETA GUI is shown in case the compared topologies are not equal.
Connections	Allows comparison of device connections.
FwVersion	Allows comparison of firmware versions.
Modules	Allows comparison of device modules.
Name	Allows comparison of device names.
OrderNumber	Allows comparison of order numbers.
IpAddress	Allows comparison of IP addresses.
IpRange	Limits the comparison to devices in a specific IP address range.

Recommendation on the test strategy

The easiest way to test the functionality is to use the routing of standard and error outputs to text files from the command line. The application exit code can be obtained by reading %errorLevel% variable in the command line. All this is possible in one line.

The following minimal commands are terminated when the comparison is completed and appears in the console return code. The “/V:ON /c” arguments allow to continue the script to continue running after the comparison is complete and the correct return code is returned to the console.

- Comparison of two topologies:

```
cmd /V:ON /c proneta.exe --compare file1="path/to/topoA.xml"
file2="path/to/topoB.xml" 1>output.txt 2>error.txt || echo
%errorLevel%
```

- Comparison of offline and online topology:

```
cmd /V:ON /c proneta.exe --compare file1="path/to/topoA.xml"
nic="NicName" showGui=true 1>output.txt 2>error.txt || echo
%errorlevel%
```

General error use cases

There are several validations in the PRONETA GUI API that can lead into an error message. Exact messages may be different, but their meaning is the same.

Table 7-11: Error messages

Use case	Exit code	Error message
Internal error	1	Unable to initialize PRONETA GUI API.
Missing or invalid Npcap/WinPcap version and user denied installation	1	Npcap is not installed!
Missing or invalid license	1	Command not processed due to a license error: Invalid license
Not accepted terms and conditions	1	The terms and conditions of the following products must be accepted in PRONETA GUI to continue: PRONETA Professional.
Invalid argument key	1	Invalid arguments detected: test
Invalid argument value	1	ShowGui does not contain a valid Boolean value
Missing required argument	1	Mandatory parameter File1 is missing or empty.
Invalid topology path	1	File c:\testTopo.xml does not exist.
Invalid topology	1	Deserialization of topology from the file c:\testTopo.xml failed with an error. There is an error in the XML document (4, 6).
Not connected adapter	1	Ethernet adapter not connected.
Another PRONETA instance scanning	1	Unable to start scan because another PRONETA instance is already scanning.

Using the ShowGui argument

It is possible to display the PRONETA GUI navigated for Comparison if the compared topologies are not equal. For the equal topologies, no PRONETA GUI is shown regardless of the value of the ShowGui argument. The exit code and standard/error output values are available once the PRONETA GUI is closed.

Comparison of two topologies

With correct input data, the comparison can only lead in two use-cases:

Table 7-12: Error messages

Use case	Exit code	Error messages
Two equal topologies	0	
Two different topologies	2	Compared topologies are not equal.

Comparison options

By specifying optional arguments, the comparison can be limited to specific topology properties. There are the most common use cases:

- Default:

```
cmd /V:ON /c proneta.exe --compare file1="topoA.xml"
file2="topoA.xml" ShowGui=false Connections=true FwVersion=true
Modules=true Name=true OrderNumber=true IpAddress=true
IpRange=192.168.0.0,192.168.0.255 1>output.txt 2>error.txt ||
echo %errorLevel%
```

- Show GUI if compared topologies are not equal:

```
cmd /V:ON /c proneta.exe --compare file1="topoA.xml"
file2="topoA.xml" ShowGui=true 1>output.txt 2>error.txt || echo
%errorLevel%
```

- Do not compare IP addresses:

```
cmd /V:ON /c proneta.exe --compare file1="topoA.xml" file2="topoA.xml"
IpAddress=false 1>output.txt 2>error.txt || echo %errorLevel%
```

- Limit comparison to specific IP range only:

```
cmd /V:ON /c proneta.exe --compare file1="topoA.xml"
file2="topoA.xml" IpRange=192.168.0.0,192.168.0.255
```


NOTE **Periodic scanning disabled**

Due to implementation difficulties, using PRONETA.exe as an API automatically disables periodic scanning if allowed.

NOTE **No PRONETA Professional license**

If there is no valid PRONETA Professional license and ALM (Automation License Manager) is installed, a trial version is automatically activated.

NOTE **No Npcap installed**

If no Npcap driver is installed, or an outdated version is detected, the automatic installation of a new driver is offered in the dialog.

7.4 Running of Third-Party Applications

PRONETA Professional facilitates the running of third-party applications, waiting for their results to be sent to the MQTT broker or Azure storage. If the waiting time exceeds a given time limit, the application terminates, and an error message is returned.

As some applications must run under a valid Local/Domain account, user credentials for the applications must be provided. If the user has insufficient rights to run applications, or to access files or folders on the file system, the start of third-party applications will fail. Applications started from the service run without a visible UI.

Definition of Third-Party Applications

The definition of third-party applications is stored in the "ApplicationRunner.xml" file. This XML file has the following structure:

```
<?xml version="1.0"?>
<ApplicationConfiguration
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://siemens.com/ApplicationConfigurationSchema">
  <Applications>
    <Application ID="{UniqueId}">
      <!--Application path.-->
      <AppPath>{PathTo.Exe}</AppPath>
      <!--Output folder where results are expected.-->
      <OutputFolder>{PathToOutputFolder}</OutputFolder>
      <!--Number of result files expected in output folder.-->
      <OutputCount>0</OutputCount>
      <!--Number of ms (Default = 60000) before waiting for results
is canceled. -->
      <Timeout>60000</Timeout>
      <!--Arguments passed to application.-->
      <Args><![CDATA[{applicationArgs}]]></Args>
    </Application>
  </Applications>
</ApplicationConfiguration>
```

There are two options to define the third-party arguments in the `<Args><![CDATA[ ]]></Args>` section:

- Without the executable name as the first argument:
  `<Args><![CDATA[arg1 arg2]]></Args>`
- With the executable name as the first argument:
  `<Args><![CDATA[{exeName} arg1 arg2]]></Args>`

Depending on the third-party application, it may be mandatory or illegal to provide the name of the executable as the first argument. If the application is not starting properly, switch to the previously unused option.

Credentials to be Used with Third-Party Applications

User credentials are stored in an encrypted file called "ApplicationRunnerCredentials.dat" located in the root folder of PRONETA Professional. It is possible to create an "ApplicationRunnerCredentials.conf" file and fill it with specific user credentials.

Example structure of "ApplicationRunnerCredentials.conf":

```
username
domain
password
```

Once PRONETA Professional service starts-up it encrypts this file. It is also possible to start the service without this unencrypted file and use configuration messages to change the default/empty credentials.

Change of Credentials

Credentials can be changed remotely by MQTT or IoT Hub message, or locally via interactive console or CLI.

- MQTT message:

```
mosquitto_pub.exe -t Proneta/Api/{cellName}/Credentials -m
"AppRunnerPassword=newPassword; AppRunnerUserName=newUserName;
AppRunnerDomain=newDomain"
```

- IoT Hub message:

```
Properties      : Command = ConfigureCredentials
Message body    : AppRunnerPassword=newPassword;
AppRunnerUserName=newUserName; AppRunnerDomain=newDomain
```

- Interactive mode:

```
Command>ConfigureCredentials AppRunnerPassword=newPassword;
AppRunnerUserName=newUserName; AppRunnerDomain=newDomain
```

- CLI:

```
PRONETA.API.SERVICE.exe --ConfigureCredentials
AppRunnerPassword=newPassword; AppRunnerUserName=newUserName;
AppRunnerDomain=newDomain
```

Download of Complete Third-Party Application Configurations

It is possible to download the content of the "ApplicationRunner.xml" file to PRONETA Professional via the following methods:

- MQTT message:

```
mosquitto_pub.exe -t Proneta/Api/{cellName}/Applications/Configure -
m "{contentOfNewXmlConfiguration}"
```

- IoT Hub message:

```
Properties      : Command = ConfigureApplications
Message body    : {contentOfNewXmlConfiguration}
```

- Interactive mode:

```
Command> ConfigureApplications
ConfigPath="pathToNewXmlConfigurationFile"
```

- CLI:

```
PRONETA.API.SERVICE.exe --ConfigureApplications
ConfigPath="pathToNewXmlConfigurationFile"
```

Upload of Complete Third-Party Application Configurations

It is possible to upload the content of ApplicationRunner.xml file from PRONETA Professional via the following methods:

- MQTT message:

```
mosquitto_pub.exe -t
Proneta/Api/{cellName}/Applications/ConfigStatus -m ""
```

- IoT Hub message:

```
Properties      : Command = SendApplicationsConfiguration
Message body    :
```

- Interactive mode:

```
Command> SendApplicationsConfiguration
```

- CLI:

```
PRONETA.API.SERVICE.exe --SendApplicationsConfiguration
```

Adding and Modifying Third-Party Application Definitions

There is no difference between adding or modifying the definition of existing thir-party applications. If a definition of a third-party application with given ID exists, its data are overwritten instead of duplicated.

- Message syntax:

```
AppPath={pathToExe}; [OutputFolder=PathToOutputFolder];
[OutputCount=NumberOfExpectedResults];
[Timeout=TimeBeforeApplicationIsKilled]; [Args=CliArguments]
```

- Message example:

```
AppPath="C:\Windows\System32\notepad.exe" OutputFolder=""
OutputCount=0 Timeout=60000 Args=C:\Windows\notepad.exe
```

- MQTT message:

```
mosquitto_pub.exe -t
Proneta/Api/{cellName}/Applications/Configure/{ApplicationId} -m
"AppPath={pathToExe}; [OutputFolder=PathToOutputFolder];
[OutputCount=NumberOfExpectedResults];
[Timeout=TimeBeforeApplicationIsKilled]; [Args=CliArguments]"
```

- IoT Hub message:

```
Properties      : Command=ConfigureApplications, Id={ApplicationId}
Message body    : AppPath={pathToExe};
[OutputFolder=PathToOutputFolder];
[OutputCount=NumberOfExpectedResults];
[Timeout=TimeBeforeApplicationIsKilled]; [Args=CliArguments]
```

- Interactive mode:

```
Command>ConfigureApplications Id={ApplicationId}
AppPath={pathToExe}; [OutputFolder=PathToOutputFolder];
[OutputCount=NumberOfExpectedResults];
[Timeout=TimeBeforeApplicationIsKilled]; [Args=CliArguments]
```

- CLI:

```
PRONETA.API.SERVICE.exe --ConfigureApplications Id={ApplicationId};
AppPath={pathToExe}; [OutputFolder=PathToOutputFolder];
[OutputCount=NumberOfExpectedResults];
[Timeout=TimeBeforeApplicationIsKilled]; [Args=CliArguments]
```

Removing Third-Party Application Definitions

Removing the definition of a third-party application is possible via the following methods:

- MQTT message:

```
mosquitto_pub.exe -t
Proneta/Api/{cellName}/Applications/Configure/{ApplicationId}/Remove
-m ""
```

- IoT Hub message:

```
Properties   : Command = ConfigureApplications, Id =
{ApplicationId}, Action=Remove
Message body :
```

- Interactive mode:

```
Command> ConfigureApplications Id={ApplicationId}; Remove
```

- CLI:

```
PRONETA.API.SERVICE.exe --ConfigureApplications Id={ApplicationId};
Remove
```

Running of Third-Party Applications

It is possible to run a third-party application if its definition exists. Non-default parameters can be specified in the message body of the invocation. If optional parameters are present, they override the default values as defined in the "ApplicationRunner.xml" file. If the number of expected result files (`OutputCount`) is non-zero, PRONETA Professional service will periodically check the defined output folder. When at least `OutputCount` number of new files are created before the defined timeout is reached, the files are uploaded.

- Syntax of parameters:

```
[OutputFolder=PathToOutputFolder];
[OutputCount=NumberOfExpectedResults];
[Timeout=TimeBeforeApplicationIsKilled]; [Args=CliArguments]
```

Use one of the following methods to run a single third-party application:

- MQTT message:

```
mosquitto_pub.exe -t
Proneta/Api/{cellName}/Applications/Run/{ApplicationId} -m
"[OutputFolder=PathToOutputFolder];
[OutputCount=NumberOfExpectedResults];
[Timeout=TimeBeforeApplicationIsKilled]; [Args=CliArguments]"
```

- IoT Hub message:

```
Properties   : Command = RunApplication, Id={ApplicationId}
Message body : [OutputFolder=PathToOutputFolder];
[OutputCount=NumberOfExpectedResults];
[Timeout=TimeBeforeApplicationIsKilled]; [Args=CliArguments]
```

- Interactive mode:

```
Command> RunApplication Id={ApplicationId};
[OutputFolder=PathToOutputFolder];
[OutputCount=NumberOfExpectedResults];
[Timeout=TimeBeforeApplicationIsKilled]; [Args=CliArguments]
```

- CLI:

```
PRONETA.API.SERVICE.exe --RunApplication Id={ApplicationId};
[OutputFolder=PathToOutputFolder];
[OutputCount=NumberOfExpectedResults];
[Timeout=TimeBeforeApplicationIsKilled]; [Args=CliArguments]
```

Third-Party Status Messages and Results

When an application is started or an error occurs, a corresponding status message is sent to the topic

```
Proneta/Api/{cellName}/Application/{ApplicationId}/Status
```

When results are available, they are uploaded to the MQTT topic

```
Proneta/Api/{cellName}/Application/{ApplicationId}/Result
```

Running of All Defined Third-Party Applications

All third-party applications that are defined in the ApplicationRunner.xml file with their default configuration can be started at once with a single call via the following methods:

- MQTT message:

```
mosquitto_pub.exe -t Proneta/Api/{cellName}/Applications/Run -m ""
```

- IoT Hub message:

```
Properties   : Command = RunAllApplications,
Message body :
```

- Interactive mode:

```
Command> RunAllApplications
```

- CLI:

```
PRONETA.API.SERVICE.exe --RunAllApplications
```

7.5 IoT Hub API-specific functions

7.5.1 Azure Device Provisioning

To use Azure device provisioning for the automatic registration of new PRONETA Professional instances, create an "IoTHubProvisioningCredentials.conf" file with the following contents:

- IdScope
- PrimaryKey
- SecondaryKey
- RegistrationId (optional)

These entries have the following meaning:

Table 7-13

Entry	Meaning
`IdScope`	ID Scope of selected the Device Provisioning Service
`PrimaryKey`	Primary key as defined in "Group Enrollment - Attestation Type - Symmetric Key"
`SecondaryKey`	Secondary key as defined in "Group Enrollment - Attestation Type - Symmetric Key"
`RegistrationId`	The Provisioning service Registration ID for the PRONETA Professional IoT Hub device. The ID must be in lowercase alphanumeric and may contains hyphens. If this ID is not defined then an ID will be created from the PC name converted to lowercase, with all non-alphanumeric characters discarded.

Restarting PRONETA Professional will automatically register it with the IoT Hub with the credentials provided.

For more information about the automated registration process, see \14\ and \15\.

7.5.2 File Upload to Azure

File Upload to the Azure Cloud Storage requires previous configuration of the IoT Hub. Details about this can be found in \13\.

The files are automatically uploaded automatically by PRONETA Professional after it received requests through the corresponding topics. The target container is located in Azure's storage space at `{Storage account}\Blobs\{blobName}\{deviceid}`, and file names are automatically generated in the process of the upload.

The following table shows some examples of file names created in that manner:

Table 7-14

Filename	Explanation
`ApiConfiguration_2019-09-13_01-59-49.xml`	API Configuration file with timestamp
`ApiLog.log`	API log file without timestamp
`topology_XX-XX-XX-XX-XX-XX_2019-09-19_12-25-31.png`	Various scanned topologies in PNG, XML, and XPS format, resp. The filename includes the MAC address of the network adapter that was used for the scan (XX-XX- XX- XX- XX- XX) and a timestamp.
`topology_XX-XX-XX-XX-XX-XX_2019-09-19_12-25-31.xml`	
`topology_XX-XX-XX-XX-XX-XX_2019-09-19_12-25-31.xps`	

From the cloud service, files can be further processed with the usual Azure services.

7.5.3 Proxy Configuration

A proxy can be configured to route data between the MQTT API and IoT Hub API of PRONETA Professional and the MQTT broker.

NOTE Use of a proxy is only possible when using a WebSocket broker address, and consequently only when using Windows 10 or Windows 11.

Establishing Credentials

Per default, the proxy requires no credentials.

There are several credentials in this version that are located in "Examples" folder:

MqttProxyCredentials.conf

MqttCredentials.conf

IoTHubProxyCredentials.conf

IoTHubProvisioningCredentials.conf

IoTHubCredentials.conf

ApplicationRunnerCredentials.conf

If you do need to provide credentials, perform the following steps.

- Create the "IoTHubProxyConfidentials.conf" with the required credentials in the folder where PRONETA Professional resides.
- Restart PRONETA Professional.

The credentials must be given as a key/value pair of username and password. Upon restart, the contents of the "IoTHubProxyConfidentials.conf" will be encrypted and stored as "IoTHubProxyConfidentials.dat". The original "IoTHubProxyConfidentials.conf" file is subsequently deleted.

7.6 Secure Data Transmission

To provide secure data transmission, PRONETA Professional provides optional user authentication and traffic encryption via TLS (Transport Layer Security). The procedures to establish security differ between the APIs.[4]

7.6.1 MQTT API

Authentication with Username and Password

For the MQTT API, to set up an "account" to control access to the communication by means of a username and password, you must create a file with the name "mqttCredentials.conf" in your PRONETA Professional root directory **before** starting the PRONETA Professional service.

The file must be a plain text file containing two words as its sole content, the first being the username, and the second being the password. For example, if the file reads

```
virgil thunderbird
```

the username defined would be "virgil" with the associated password "thunderbird".

The password can be of any length and has no requirements to contain digits, special characters, etc. Case is significant for both the username and password.

If on starting up of the PRONETA Professional service, and before the connection to the broker is established, the "mqttCredentials.conf" file exists, the credentials will be extracted, encrypted on the file system of the PRONETA Professional PC, and forwarded to the broker. After that, the original "mqttCredentials.conf" file is deleted from the PRONETA Professional PC.

From now on, the PRONETA Professional topics service will provide the correct username and password to connect with the broker.

Change of Credentials

It is possible to change the valid username/password credentials while PRONETA Professional is running through publishing MQTT messages.

For this purpose, the API is subscribed to the following topics:

```
{CoreTopic}/Credentials
{CoreTopic}/{Group}/Credentials
{CoreTopic}/{Cell}/Credentials
```

A message sent to either of them should contain key/value pairs for the keys `MqttUserName` and `MqttPassword`. These are used as the new credentials after the MQTT API client has reinitialized.

Encrypted Communication with TLS

PRONETA Professional allows the encryption of its MQTT communication with TLS V1.2 to avoid the listening in of unauthorized stations. Encryption is not mandatory, but highly recommended.

For this purpose, both the clients and the broker must be configured to use TLS. Detailed information about secure MQTT TLS communication can be found on the web, for example at \5\. Most importantly, the client in question – PRONETA Professional – must possess a valid CA (Certification Authorities) certificate from the broker, or a valid CA certificate must be installed in the Windows "Trusted Root Certification Authorities" certificate store.

[4] To disable the secure communication, see 7.1.7.

For the PRONETA Professional service, the following entries must be present and correct in the "ApiConfiguration.xml" file (see 7.1.7):

- `BrokerAddress`
- `BrokerPort`
- `CaCertificatePath`

The following conditions must be met:

- `BrokerAddress` must be the URL or IP address of the computer on which the broker is running, and from which the CA certificate stored at the client was derived.
- `BrokerPort` is the port of the broker for encrypted communication; the default value is "8883".
- `CaCertificatePath` is a directory which contains the CA certificate issued by the broker. It can be left empty if a certificate is present in the "Trusted Root Certification Authorities" certificate store.

NOTE To enable secure MQTT communication via TLS, the firewall of the PRONETA Professional PC must not block port 8883.

7.6.2 IoT Hub API

"IoTHubCredentials.conf" and the Primary Connection String

Credentials for accessing the IoT Hub are stored in the "IoTHubCredentials.conf" file on the machine on which PRONETA Professional and the IoT Hub API are running. The "IoTHubCredentials.conf" contains the "Primary Connection String" with the information required to connect to the IoT Hub with the credentials set up for a specific device or instance of PRONETA Professional.
An example of this would be:

```
HostName=PronetaProfessionalV1Update1Dev.azure-
devices.net;DeviceId=provisioneddeviceid;SharedAccessKey=XK/opTQcDo=
```

Upon startup of the machine, the file is encrypted, and the encrypted information stored as "IoTHubCredentials.dat", while the original file is deleted.

Valid credentials can be created by means of the Azure Device Explorer. (See \9\ for more details.)

Change of Credentials

To convey the credentials to the IoT Hub API, there are two options:

- **Manually**: Copy the new Primary Connection String into the "IoIHubCredentials.conf" and trigger a restart of the service.
- **Through the IoT Hub**: The IoT Hub needs to send a message to the IoT Hub API with the property and property value of `Command = ConfigureCredentials`, and a message which contains the new Primary Connection String.

An IoT Hub message must look like this:

```
Properties      : Command = ConfigureCredentials
Message body    :
IotHubConnectionString=”HostName=PronetaProfessionalV1Update1Dev.azu
redevices.
net;DeviceId=provisioneddeviceid;SharedAccessKey=XK/opTQcDo=”
```

Change of Proxy Credentials

IoT Hub message:

```
Properties      : Command = ConfigureCredentials
Message body    : IotHubProxyUserName=virgil IotHubProxyPassword=
thunderbird
```

Once the credentials are established and PRONETA Professional is restarted, the IoT Hub API is ready to communicate with the IoT Hub, in particular, to receive request messages from the Hub and to send answering messages and files for storage in Azure.

7.7 PRONETA Professional and PRONETA Basic

PRONETA Professional Status in PRONETA Basic

When the PRONETA Professional service is installed, the PRONETA home screen is extended with another panel which displays the current status of the PRONETA Professional service. The status can have one of the following values:

- "Not installed"
- "Stopped", service is installed but not running
- "Running", service is installed and running

Parallel Operation of PRONETA Basic and PRONETA Professional

In principle, the PRONETA Professional service can run in parallel with a single instance of PRONETA Basic. This means that it is possible to use PRONETA's features (like the IO Test, see the PRONETA manual \4\) simultaneously with the automatic creation of a network scan. There are some limitations to this process, however:

- At any time, there must only be one network scanner in operation on any PC. This means that there can be temporary conflicts when PRONETA Basic's network scan is employed at the same time as an automated network scan by PRONETA Professional.
  Error messages are displayed in this case, but problems arising from this are transient.
- Only PRONETA versions 2.7 and higher are compatible with PRONETA Professional. If an older PRONETA version is active when PRONETA Professional starts up, the process of the old version is killed and PRONETA Professional service is launched.

8 Troubleshooting

This section contains hints and tips to ensure proper operation of PRONETA Professional.

The PRONETA Professional license becomes invalid after the system clock of the PC has been changed.

Manipulation of the PC system clock can lead to a loss of SIMATIC licenses on the PC.

- Solution:
  For more information about this error and how to reactivate the PRONETA Professional license, see the notice in 3.5.

The PRONETA Professional service is running but performs no scans.

If there is an error in the configuration file "ApiConfiguration.xml" (see 7.1) in the `<ResultConfiguration>` or `<ScanConfiguration>` section, then the PRONETA Professional service will start, but only in limited mode.

In this case, the service will send a message about its invalid configuration through all enabled APIs (MQTT, IoT Hub), and will subsequently process all requests **except** scan requests. Upon a scan request it will reply that scans are impossible due to an invalid configuration.

- Solution:
  Check the log (see 7.2.4) for error messages and provide a correct "ApiConfiguration.xml" file.

The PRONETA Professional service is running, but performs no scans, and "ApiConfiguration.xml" contains no errors.

If in the "ApiConfiguration.xml" file `<NicMacs Mode>` or `<NicNames Mode>` is set to "Allowed" (see 7.1.1), but the corresponding `<NicMac>` or `<NicName>` list is empty, no scans are performed.

The same problem occurs if `<NicMacs Mode=Blocked>` and `<NicNames Mode=Allowed>`, while both lists are empty.

- Solution:
  Provide valid NIC MACs addresses or NIC names in the `<Nic Mac>` and `<NicName>` lists.

The PRONETA Professional service doesn't start scanning and PRONETA Basic is performing a scan at the same time.

At any time, only one scanner – either PRONETA Basic **or** PRONETA Professional – can be active on a computer.

- Solution:
  If the PRONETA Professional service detects a running instance of PRONETA Basic, it will display a popup dialog, prompting the user to cancel the PRONETA Basic scan. If the user agrees, the PRONETA Basic scan will be aborted immediately. If the user refuses, PRONETA Professional will wait a certain amount of time before terminating itself, and then issue an error message.

This behavior gives the user an opportunity to manually abort a PRONETA Basic scan.

NOTE The time for PRONETA Professional to wait for a cancellation of the PRONETA Basic scan can be set to the desired value through the parameter `<WaitForScan>` in the "ApiConfiguration.xml" file (see 7.1.1).

I can't send an updated "ApiConfiguration.xml" file to my computer.

If possible, you should send modifications to your new API Configuration via MQTT/IoT Hub API to the target computer. However, if there is error in the `<ApiSpecificConfigurations>` section of the configuration file, communication over the API concerned will no longer work.

- Solution:
  Edit the configuration file manually and restart the PRONETA Professional service.

NOTE System administrators may deploy new versions of the "ApiConfiguration.xml" file and reboot the service remotely, using Windows / third-party mechanisms. However, these mechanisms are not part of PRONETA Professional.

No communication between PRONETA Professional and Azure/IoT Hubs can be established.

A) Communication between PRONETA Professional and IoT Hubs can use a variety of protocols running over different TCP/IP ports which may be blocked by a firewall in corporate environments. (See \11\ for more information about IoT Hub protocols.)

- Solution:
  Configure your firewall to keep the following ports need open, depending on the particular protocol used:

Table 8-1

Protocol	Port
HTTP1	443
AMQP	5671
MQTT	8883

NOTE Even when AMQP or MQTT communication is utilized for messaging, HTTP1 is still used for file upload to Azure Cloud Storage. This will only work with port "443" not being blocked.

B) When using the HTTP1 protocol, it is also possible that communication **appears** to be defunct, but actually runs very slowly. Depending on the number of registered devices, the intervals at which polling requests are sent must be carefully balanced: an interval that is to large could lead in sluggish communication, an interval that is to small could lead to the IoT Hub being overwhelmed with queries and unable to respond.

- Solution:
  - Switch to a different protocol or WebSockets to achieve faster communication. (See also \10\ to learn more about IoT Hub throttling, and \12\ for information about differences between HTTP1 and WebSockets.)
  - Set the "Http1PollingInterval" parameter to an appropriate value for the number of registered devices.

Fatal system errors (crashes) occur randomly during network scans.

If the Npcap packet sniffer is present on the PC running PRONETA Professional, it can interfere with the WinPcap sniffer installed with PRONETA Professional, resulting in occasional fatal system errors and "bluescreens".

PRONETA Professional is compatible with WinPcap V4.1.3 and Npcap V1.0 or higher. If both WinPcap and Npcap are running on the computer, PRONETA's communications will be routed to Npcap, leading to unpredictable system behaviour.

- Solution:
  Remove all instances of Npcap from your PC running PRONETA Professional.

Only the Restart command can be executed, all other commands receive no response-

This is the case when neither the WinPcap nor the Npcap driver are present on the PC.

In this case, the API will send following error message after startup:

"WinPcap version 4.1.3 or Npcap version 1.0 or higher is not installed. PRONETA Professional Service was started in limited mode".

Any further messages to the PC will receive an error message as reply:

"Command was not processed because WinPcap version 1.0 or higher is not installed. Please install it and reboot PRONETA Professional service."

- Solution:
  Install WinPcap or Npcap. (For more information, see also the PRONETA Basic User Manual (\4\).

Tracing of the PRONETA Professional communication as described in section 6.6 is not possible.

The tracing of the communication is performed via loopback TCP communication employing port "1234".[5] If a firewall blocks this port, attempt at tracing will fail.

- Solution:
  Make sure that port "1234" is open and not blocked by any firewall.

[5] This port is fixed and cannot be changed.

9 Appendix

9.1 Service and Support

Industry Online Support

Do you have any questions or need assistance?

Siemens Industry Online Support offers round the clock access to our entire service and support know-how and portfolio.

The Industry Online Support is the central address for information about our products, solutions, and services.

Product information, manuals, downloads, FAQs, application examples and videos – all information is accessible with just a few mouse clicks at:
https://support.industry.siemens.com

Technical Support

The Technical Support of Siemens Industry provides you fast and competent support regarding all technical queries with numerous tailor-mades offers – ranging from basic support to individual support contracts. You send queries to the Technical Support via Web form:
www.siemens.com/industry/supportrequest

Service offer

Our range of services includes, inter alia, the following:

- Product trainings
- Plant data services
- Spare parts services
- Repair services
- On-site and maintenance services
- Retrofitting and modernization services
- Service programs and contracts

You can find detailed information on our range of services in the service catalog:
https://support.industry.siemens.com/cs/sc

Industry Online Support app

You will receive optimum support wherever you are with the "Siemens Industry Online Support" app. The app is available for Apple iOS, Android, and Windows Phone:
https://support.industry.siemens.com/cs/sc

9.2 Related literature

Table 9-1

	Topic
\1\	Siemens Industry Online Support https://support.industry.siemens.com
\2\	Download page of this entry https://support.industry.siemens.com/cs/ww/en/view/109768642
\3\	PRONETA website https://new.siemens.com/global/en/products/automation/industrial-communication/profinet/portfolio/proneta.html
\4\	PRONETA Basic download page (with PRONETA Basic User Manual) https://support.industry.siemens.com/cs/ww/en/view/67460624
\5\	Secure MQTT communication: http://www.steves-internet-guide.com/mosquitto-tls/
\6\	SIEMENS Industry Mall: https://mall.industry.siemens.com
\7\	PRONETA Professional in the SIEMENS Industry Mall: https://mall.industry.siemens.com/mall/en/WW/Catalog/Products/10356514
\8\	.NET Core Download page (Microsoft): https://dotnet.microsoft.com/download/dotnet-core/2.2
\9\	Azure Device Explorer: https://iotforum.advantech.com/discussion/89/how-to-use-device-explorer-for-iot-hub-devices/
\10\	"Throttling" of IoT Hubs: https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-devguide-quotas-throttling#operation-throttles
\11\	Protocols used with IoT Hubs: https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-devguide-protocols
\12\	Differences between HTTP1 and WebSocket protocol variants: https://docs.microsoft.com/en-us/azure/application-gateway/application-gateway-WebSocket#how-does-WebSocket-work
\13\	Configuration of file upload to Azure through IoT Hubs: https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-configure-file-upload
\14\	Group enrollment procedure for Azure: https://docs.microsoft.com/en-us/azure/iot-dps/how-to-manage-enrollments#create-a-device-enrollment
\15\	Procedure to link IoT Hubs to the Device Provision Service: https://docs.microsoft.com/en-us/azure/iot-dps/quick-setup-auto-provision
\16\	Official Wireshark download page: https://www.wireshark.org
\17\	Siemens Technical Support: https://support.industry.siemens.com/cs/my/src?lc=en-WW
\18\	Overview over Siemens license concepts: https://new.siemens.com/global/en/products/automation/topic-areas/simatic/licenses.html

	Topic
\19\	PROFIenergy as a PROFINET profile: https://new.siemens.com/global/en/products/automation/industrial-communication/profinet/profile.html
\20\	Npcap website: https://nmap.org/npcap/

9.3 Documentation changes

Table 9-2

Version	Date	Modifications
V1.0	06/2019	First version
V1.01	11/2019	Updated documentation to the software version 1.0 Upd 1
V1.1	08/2020	Updated documentation to the software version 1.1
V1.1. SP1	02/2022	Updated documentation to the software version 1.1 SP1
V1.1. SP2	03/2023	Updated documentation to the software version 1.1 SP2