zenon manual

Installation and updates - zenon Software Platform

v.14

Table of contents

1 Welcome to COPA-DATA help

GENERAL HELP

If you cannot find any information you require in this help chapter or can think of anything that you would like added, please send an email to documentation@copadata.com.

LICENSES AND SERVICES

If you find that you need other zenon services or licenses, our staff will be happy to help you. Email sales@copadata.com.

PROJECT SUPPORT

You can receive support for any real project you may have from our customer service team, which you can contact via email at support@copadata.com.

2 Installation zenon Software Platform

During the first installation of zenon, the installation routine automatically starts and leads you through the entire installation process. If the autoplay of media is disabled in the operating system of the computer, the installation routine will not start automatically. Start the installation by executing the **START.exe** file in the root folder of your zenon installation medium.

> ### Hint
>
> The autoplay of media can be enabled in the settings of your operating system.
>
> - Press the **Windows button + I** to open the Windows settings dialog.
> - Enter *Enable/Disable Auto Play* in the input field.
>   This opens the system configuration dialog.
> - Select the **Use AutoPlay for all media and devices** option.

Notes for the installation:

- Before installing zenon:
    - All current operating system updates must be installed
      Note: If you always use the latest version (Service Pack) of your operating system, you not only avoid compatibility issues but also security problems.
    - There must not be a restart pending
- During the installation of zenon, the **COPA-DATA Multiple Network Protocol Driver (cdprotdrv.sys)** is installed. To start the driver, the operating system must be restarted after installation.

  > ⚠**Attention**
  >
  > From version 7.10 on, zenon cannot be installed on systems on which the **Microsoft SQL Server Data Engine** (**MSDE**) is already installed. This affects all systems in which zenon 6.01 or 6.20 have been installed.

> **Information**
>
> If you receive an error message during installation stating that a service cannot be started, then:
>
> - first reboot the computer
> - then start the zenon setup again

3 System requirements

Changes due to Service Packs, Hotfixes or Patches from Microsoft can cause incompatibilities and affect the functionality of the software. In this case, COPA-DATA will provide an updated version of zenon as soon as possible. In this case you can get more information from COPA-DATA support: support@copadata.com.

⚠Attention

Note when configuring the project:

- For the optimal display ofzenon in the Service Engine, the standard setting (corresponds to *100%*) is recommended for the Windows **display**. Higher values can lead to graphic elements, symbols, texts, etc. not being displayed correctly.
- Windows themes can overlay elements in Service Engine. Ensure, when configuring a project, that there is an appropriate distance from the elements to the screen edge.

Note: According to Windows conventions, hostnames may not contain more than 15 characters.

3.1 Operating systems

In this node, you will find information about different operating systems supported by the zenon Software Platform version 14.

3.1.1 Windows Operating Systems (1/2)

Supported operating systems and minimum required Windows Service Pack/version:

Operating system	zenon Engineering Studio	Service Engine	Logic Service	zenon Report Engine
Windows 10 (Pro, Enterprise)	Only x64 from version 1607	From version 1607	From version 1607	Only x64 from version 1607
Windows 10 (Enterprise LTSC, IoT Enterprise LTSC)	Only x64 from version 2016	From version 2016	From version 2016	Only x64 from version 2016
Windows 11 (Pro, Enterprise)	Only x64 from version 21H2	From version 21H2	From version 21H2	Only x64 from version 21H2
Windows Server 2016 (All editions with the exception of Core)	Only x64 from version 1809	From version 1607	From version 1607	Only x64 from version 1607
Windows Server 2019 (All editions with the exception of Core)	Only x64 from version 1809	From version 1809	From version 1809	Only x64 from version 1809
Windows Server 2022 (All editions with the exception of Core)	Only x64 from version 21H2	From version 21H2	From version 21H2	Only x64 from version 21H2

LTSC VERSIONS

An overview of LTSC versions and their **Equivalent GA Channel releases** can be found on the Microsoft Homepage (https://learn.microsoft.com/en-us/windows/whats-new/ltsc/).

3.1.2 Windows Operating Systems (2/2)

Supported operating systems and minimum required Windows Service Pack/version:

Operating system	zenon Smart Server	zenon Smart Client	zenon HTML Web Engine Web Visualization Service	IIoT Services
Windows 10 (Pro, Enterprise)	From version 1607	From version 1607	From version 1607	From version 1607
Windows 10 (Enterprise LTSC, IoT Enterprise LTSC)	From version 2016	From version 2016	From version 2016	From version 2016
Windows 11 (Pro, Enterprise)	From version 21H2	From version 21H2	From version 21H2	From version 21H2
Windows Server 2016 (All editions with the exception of Core)	From version 1607	From version 1607	From version 1607	From version 1607
Windows Server 2019 (All editions with the exception of Core)	From version 1809	From version 1809	From version 1809	From version 1809
Windows Server 2022 (All editions with the exception of Core)	From version 21H2	From version 21H2	From version 21H2	From version 21H2

3.1.3 Linux

Supported Operating Systems and zenon Services for Linux:

Hardware	Operating system	Service Engine for Linux	Logic Service	Legacy T5 Logic Service for Linux
PC	*Ubuntu 22.04, x86*	--	--	X
PC	*Ubuntu 22.04, amd64*	X	X	--
Raspberry Pi 4	*Raspberry Pi OS 11, armhf*	--	--	X
Raspberry Pi 4	*Raspberry Pi OS 11, arm64*	X	X	--
Siemens IoT2050	*Siemens Industrial OS 3.2.3, arm64*	X	X	--

Key:

X: supported

--: not supported

REQUIREMENTS

To be able to run Service Engine in a Linux environment, the following applies:

- zenon projects are supported from version 12.
- zenon projects can only be run for the operating systems listed in the table.

- For Service Engine, a 64-bit operating system is required.

3.1.4 Linux-based container (Docker)

Supported zenon apps for Linux amd64 Docker containers:

- Service Engine for Linux
- zenon Diagnosis Server for Linux
- **HTML Web Engine**
- IIoT Services:
  - Data Hub
  - Data Storage
  - Certificate Management
  - Identity Service
  - Identity Management
  - Platform Configuration
  - IIoT API
  - Persistence Service
  - Service Configuration Studio
  - Device Management
  - Proxy Service
  - Trend Visualizer
  - Data Modeling

3.1.5 Windows-based Docker

Windows Docker Container supports the Service Engine.

3.2 System requirements when using DirectX

The following minimum requirements must be met when using *DirectX hardware* or *DirectX software*:

Note: For extensive projects or several projects loaded at the same time you will need accordingly faster/stronger hardware. The minimum requirements can increase as a result of this.

Parameter	Minimum requirements	Recommended
CPU:	Single core with SSE2 support.	Quad Core or more cores

Parameter	Minimum requirements	Recommended
Graphics adapter: (DirectX hardware only)	DirectX 11 mainstream graphics card. **Note:** When an integrated graphics chip is used in particular, it is possible, depending on the driver used, that there are impairments to the display quality.	Dedicated DirectX 11 AMD or nVidia high-end graphics card
Graphics memory: (DirectX hardware only)	1 GB VRAM **Note:** The size that is actually needed depends on the number of screens called up and the elements displayed.	2 GB VRAM
Graphics card driver: (DirectX hardware only)	The graphics card manufacturer's most recent driver.	
Operating system:	*DirectX Hardware* and *DirectX Software* only work on operating systems with *DirectX 11.1* support. If the system does not support *DirectX 11.1*, it is automatically switched to *Windows based* . The current *DirectX*-Service Engine must be installed. For zenon it is installed together with the setup. It must be manually installed for **Smart Client**.	

You can check the DirectX hardware compatibility of the graphics card and the driver with the **dxdiag.exe** of the Windows operating system.
As of Windows 8: All supported versions of DirectX are displayed in the **Display** tab under **Feature Levels**. For example, DirectX 11 is displayed as *11.0*.

3.3 User authorization

Windows administrator rights are required for installation.

Standard Windows user rights are required for ongoing operation. The user account control (UAC) can be activated at the highest security level.

⚠Attention

Keep in mind that Windows user rights settings are not overridden by internal security software.

3.4 Hardware requirements

In this node, you can find the hardware requirements for the individual versions of Engineering Studio and Service Engine, as well as **Smart Server** and **Smart Client**. This information represents the requirements for a system with average complexity and project size. When a zenon service has additional restrictions, this is documented accordingly in the relevant node of the Help.

For your planning, also take into account consumption of zenon independent system resources such as the size of the storage medium, main memory, CPU performance, etc.

⚠Attention

Graphics cards with their own graphics memory and DirectX support are recommended. Shared-memory graphics cards may require too much working memory and may thus lead to performance impairments. Note the system requirements when using DirectX (on page 14) node. The recommended configuration from this node is to be noted for the use of Multi-Touch.

ENGINEERING

Engineering Studio uses a Microsoft SQL Server as an SQL Server and has higher hardware requirements than Service Engine. If Engineering Studio and Service Engine are to be running on a system simultaneously, the requirements increase.

Information

In Service Engine, an automatic check is carried out to determine whether the computer has sufficient hardware and operating system resources for the current process.

For further information, see the **Service Engine** node in the **System integrity monitoring** node.

PERFORMANCE OPTIMIZATION

Note that all information stated only constitutes the minimum requirements for your system. Better hardware equipment improves the performance of zenon considerably.

- Equip your hardware - both clients and most of all the server - with sufficient memory (RAM).
- Optimize the hardware for data backup, for example with fast SSD data storage.
- Match the hardware of the clients and the network to one another. A system is only as powerful as its weakest component.
- Optimize your network architecture, for example with the use of cabling with a high data transfer rate and corresponding devices (switches and routers).
- When using a virtual environment, ensure that the virtual system is correspondingly configured with performance optimization.

3.4.1 Engineering Studio

The minimum requirements are based on a complete installation of Engineering Studio. For extensive projects or several projects loaded at the same time you will need accordingly faster/stronger hardware. The minimum requirements can increase as a result of this.

Hardware	Minimum requirements	Recommended
CPU	Single core with SSE2 support.	Quad Core
Memory	From 4 GB **Note**: The more projects you have simultaneously available in memory, the more memory you need.	8 GB
Storage medium	40GB of free memory is required to install the entire zenon Software Platform. The following applies for individual installations: - Engineering Studio: 10 GB - Service Engine: 6 GB - Report Engine: 7 GB - Reporting Studio: 4 GB - **Smart Server**: 2 GB - **Smart Client**: 3 GB - IIoT Services: 4 GB - License Management: 1 GB In addition, free memory is required for	

Hardware	Minimum requirements	Recommended
	projects.	
Monitor resolution	Extended VGA with 1024 x 768 pixels. **Attention:** Some dialogs, e.g. the filter of the **AML/CEL** image, are difficult to operate or possibly non-operable at a height of less than 850 pixels.	Double monitor setup: 2 times *1920 x 1080*
Graphics adapter	64 MB dedicated memory. Cards with shared memory can lead to performance loss. Note the system requirements when using DirectX (on page 14) node.	
Input devices	Standard keyboard or standard mouse.	
USB interface or DVD drive	For the installation, regardless of installation medium. The installation is also possible via network. Installation files can also be downloaded from the customer area of the COPA-DATA website.	
Parallel or USB interface	In case of dongle licensing required for dongle. For network dongle only required for the dongle server.	
Network connection (optional)	10 Mbit/s with TCP/IP protocol for Remote Transport, network dongle, project backups on central file server, multi-user capable Engineering Studio, etc.	1000 Mbit/s

Note: When using comprehensive multi-user projects, note the information in the hardware requirements node.

3.4.2 Service Engine

Minimum requirements refer to a complete installation of the Service Engine.

For extensive projects or several projects loaded at the same time you will need accordingly faster/stronger hardware. The minimum requirements can increase as a result of this.

Hardware	Minimum requirements	Recommended
CPU	Single core with SSE2 support.	Quad Core

Hardware	Minimum requirements	Recommended
Memory	from 512 MB. **Note:** Projects with large amounts of data, network projects, several parallel projects and projects in redundant operation require more memory.	4096 MB
Storage medium	40GB of free memory is required to install the entire zenon Software Platform. The following applies for individual installations: - Engineering Studio: 10 GB - Service Engine: 6 GB - Report Engine: 7 GB - Reporting Studio: 4 GB - **Smart Server**: 2 GB - **Smart Client**: 3 GB - IIoT Services: 4 GB - License Management: 1 GB In addition, free memory is required for projects. **Attention:** If you log historical data (e.g. archived data or alarm/CEL data), there will need to be sufficient memory available or the configuration has to ensure that the historical data is evacuated or deleted.	
Monitor resolution	1024 x 768. Note also the information in the infobox under the table.	
Graphics adapter	64 MB dedicated memory. Cards with shared memory can lead to performance loss. Note the System requirements when using DirectX (on page 14) node in relation to this.	
Input devices	Keyboard and/or mouse. Operation via touchscreen is also possible. There are many individual, customizable touch keyboards	

Hardware	Minimum requirements	Recommended
	available for the touchscreen. In addition, there is the possibility of Multi-Touch operation.	
USB interface **(optional)**	▸ For installation. Installation also possible via network or other storage media. ▸ For dongle. Network dongle also available.	
Network connection (optional)	64 kBits/s for standard Client/Server projects. 100 Mbit/s full duplex for redundant operation.	1000 Mbits/s full duplex for standard Client/Server projects.
Remote connection (optional)	Minimum requirements: Dial-up modem with 9600 Bit/s.	1 Mbit/s full duplex.
WAN connection (optional)	Any desired conncetion via router, e.g. per ISDN or DSL. Data transfer is slower in a WAN than in a local network for technical reasons. Be sure to check the possible data transfer rates of your WAN technology already at the time when you create the project.	
Message Control (optional)	Please refer to the Message Control node for the requirements.	
Interfaces (optional)	The necessary interfaces depend on the requirements of the PLC and/or the bus connection, for example serial RS232 or RS422/485 interfaces, ISA/PCI slots, etc.	

Information

The minimum recommended resolution in Service Engine is 1024 x 768 pixels. Smaller resolutions can also be configured. However it may then not be possible to operate some online dialogs. If these are not used, the resolution can be selected as lower.

3.4.3 Report Engine

The following prerequisites are applicable for work with Report Engine:

HARDWARE

Report Engine Server:

Parameters	Recommended	Minimum
CPU	CPU XEON (maximum 24 cores/4 sockets)	XEON processor
RAM	64GB or larger	32 GB
Free memory	1TB or more (depending on the data to be saved)	1 TB

Engineering computer:

Parameters	Recommended	Minimum
CPU	CPU i7 or more powerful	CPU i7
RAM	64GB or larger	32 GB
Free memory	1TB or more (depending on the data to be saved)	2 GB
Monitor (pixels)	Full HD or higher	1024 x 768

CONNECTORS

The following is applicable for the Service Engine Connector:

- Timeout: is independent of the report timeout.
  Default: 5 minutes (can be configured)
- Variables: Only variables that are listed in metadata are requested
- String variable: maximum of 4000 characters

The performance of a Connector depends on the:

- Performance of the Report Engine server
- Performance of the Service Engine server

- Service Engine server load (connector runs with lower priority)
- Network performance and network load

PROJECTS AND FILTERS

Reports can generally be created throughout several projects.

> ⚠**Attention**
>
> Archive data can only be evaluated if the variables and archive configuration are in the same project.
>
> This means: For example, in an integration project, if a variable from a subproject is archived in an archive, then Report Engine cannot access this variable.

SCHEDULES

- **Calendar days in months** are limited to *1 - 28* (corresponds to February in non-leap years)
- The "*Month end*" event is not available

> **Information**
>
> Do not use zenon color palettes for dynamic limit values for zenon projects whose data is to be exported for Report Engine. Limit values cannot be dynamically changed in Report Engine. Information from color palettes can therefore not be evaluated. This can lead to illegible graphics.

3.4.4 Smart Server

The minimum requirements are based on a complete installation of **Smart Servers**. For extensive projects or several projects loaded at the same time you will need accordingly faster/stronger hardware. The minimum requirements can increase as a result of this.

Hardware	Minimum requirements	Recommended
CPU	Single core with SSE2 support.	
Memory	From 1024 MB	
Storage medium	256 MB free harddrive space.	1 GB free harddrive space.

Hardware	Minimum requirements	Recommended
Network connection	10 Mbit/s full duplex.	1000 Mbit/s full duplex.
Remote connection (optional)	Minimum requirements: Dial-up modem with 9600 Bit/s.	1 Mbit/s full duplex.

3.4.5 Smart Client

The minimum requirements are based on a complete installation of **Smart Client**. For extensive projects or several projects loaded at the same time you will need accordingly faster/stronger hardware. The minimum requirements can increase as a result of this.

Hardware	Minimum requirements	Recommended
CPU	Single core with SSE2 support.	Quad Core
Memory	From 1024 MB	
Storage medium	64 MB of free space for **Smart Client** plus space for the projects.	80 GB free harddrive space.
Network connection	10 Mbit/s full duplex.	1000 Mbit/s full duplex.
Remote connection (optional)	Minimum requirements: Dial-up modem with 9600 Bit/s.	1 Mbit/s full duplex.
Graphics adapter	64 MB dedicated memory. Cards with shared memory can lead to performance loss. Note the System requirements when using DirectX (on page 14) node in relation to this.	

4 Platform setup

In this node, you can find detailed information on setup for apps and components of the zenon software platform.

4.1 zenon Software Platform standard installation

zenon will automatically start its installation routine and guide you through the whole installation process when the zenon installation medium is connected. Alternatively, it is possible to start the installation by executing **START.exe** in the root folder of your zenon installation medium.

The zenon software platform is available in different embodiments with different names. Product names and the scope may differ from the standard installation described.

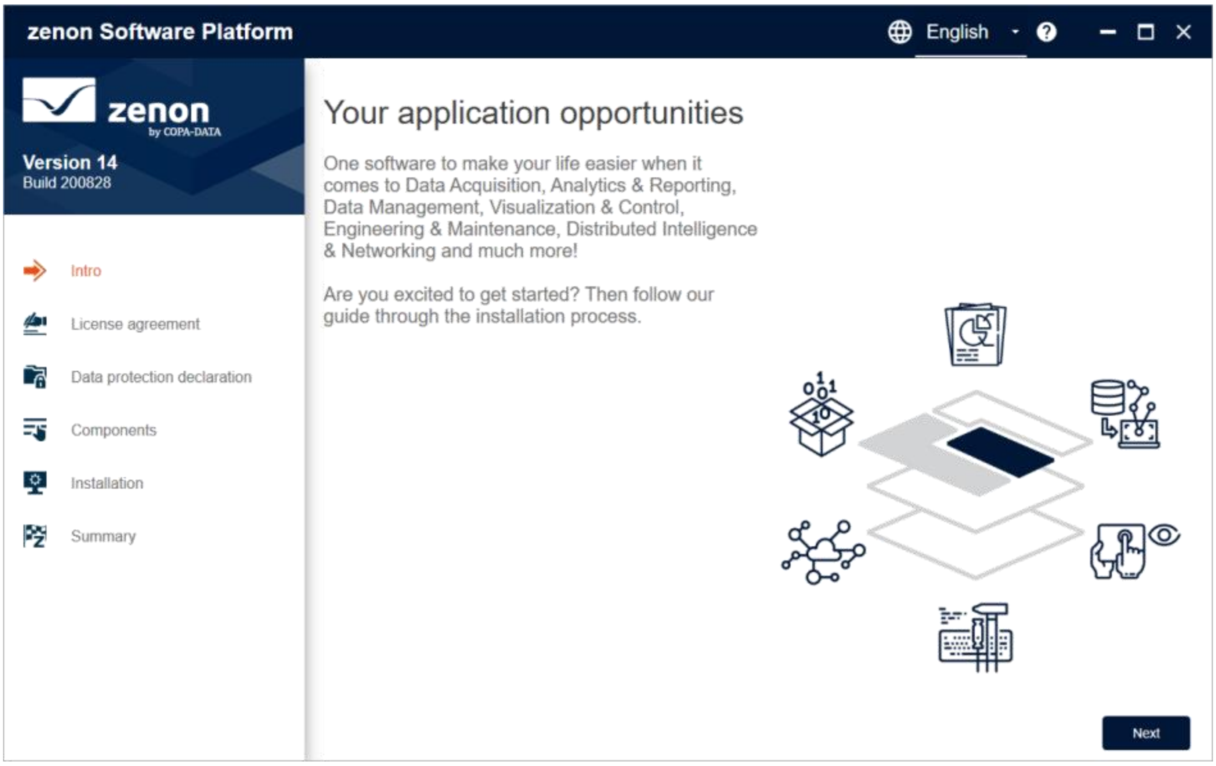
Notes:

- Administrator rights are required for the installation process on the computer!
- A restart may be necessary during installation.
  Close all other programs before installation.
- The minimum screen resolution for the setup is 1280 x 800 pixels.

4.1.1 Start window

You are given general information about the zenon Software Platform in the start window.

The information in the left window shows you the current status of the installation process. You switch to the next respective window with the **Next** button.
You can get help on installation by clicking on the Help symbol at the top right.



Your application opportunities

One software to make your life easier when it comes to Data Acquisition, Analytics & Reporting, Data Management, Visualization & Control, Engineering & Maintenance, Distributed Intelligence & Networking and much more!

Are you excited to get started? Then follow our guide through the installation process.

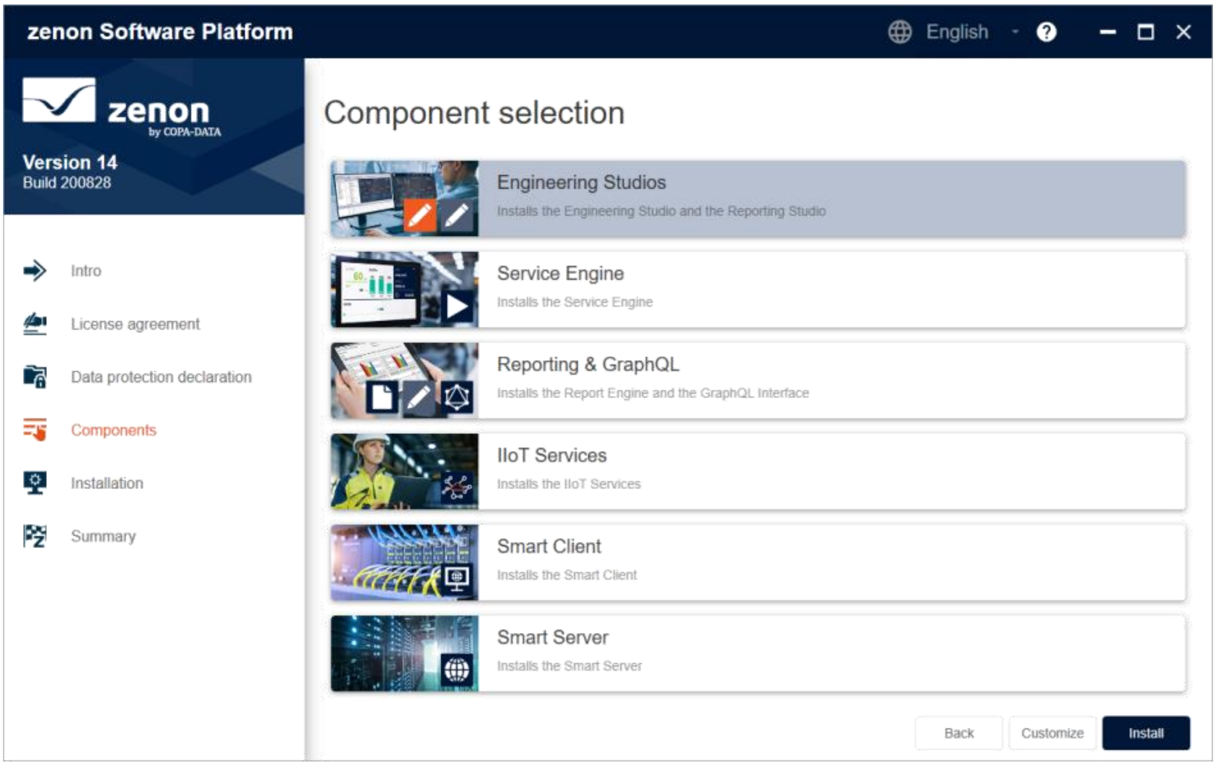
1. From the drop-down list at the top right, select the desired language for installation.
   The following languages are available:
   - German
   - English
   - Italian
   - French
   - Spanish
   - Czech
   - Japanese
   - Korean
   - Chinese
   - Russian

   Note: The language can only be changed on this page. In the following steps, the language is shown but can no longer be amended.
2. Clicking on the **Next** button opens the window with the license conditions.
3. Confirm the license conditions by activating the corresponding checkbox.
   If you do not accept the license conditions, you cannot install the product.

   You can also print the license conditions out by clicking on the **Print** button.
4. Clicking on the **Next** button opens the privacy policy.
   Read the privacy policy carefully.
   You can print out the privacy policy by clicking on the **Print** button.
5. Activate the checkbox for the privacy policy.
   This will confirm that you have read this. If you do not accept the privacy policy, the product cannot be installed.
6. Clicking on the **Next** button opens the window to select the desired product.
   Note: The **Next** button is only available if you have agreed to the license conditions by activating the checkbox.

4.1.2 zenon Standard installation

Select the desired components. It is only possible to select components that have not already been installed. If you want to carry out a reinstallation, you must first uninstall the previously-installed components using the Windows **Apps** application or the Control Panel.



Component selection

Engineering Studios
Installs the Engineering Studio and the Reporting Studio

Service Engine
Installs the Service Engine

Reporting & GraphQL
Installs the Report Engine and the GraphQL Interface

IIoT Services
Installs the IIoT Services

Smart Client
Installs the Smart Client

Smart Server
Installs the Smart Server

INSTALLATION PACKAGES

There are 6 collections of packages available for installation. By default, only the pre-selected components are installed, as well as components that you choose to be installed. You can individually add or remove further components before installation by using the **Customize** button.

Note: If an exclamation mark is shown for a package, this indicates incompatibility for the installation. For example, already-installed components that do not permit the installation of a further version. In this case, select the adjusted installation.

Package	Components
Engineering Studios	Installs Engineering Studio a Reporting Studio. **Preselected:** - Engineering Studio - Reporting Studio - Device Management Interface components

Package	Components
	- License Manager **Optional:** - Service Engine (pre-selected) - Report Engine - GraphQL Interface - Smart Client - Smart Server - IIoT Services
Service Engine	Installs Service Engine. **Preselected:** - Service Engine - Device Management Interface components - License Manager **Optional:** - Engineering Studio - Report Engine - GraphQL Interface - Reporting Studio - Smart Client - Smart Server - IIoT Services
Reporting & GraphQL	Installs Report Engine and the **GraphQL Interface**. **Preselected:** - Report Engine - IIoT Services - License Manager **Optional:** - Engineering Studio

Package	Components
	▶ Service Engine ▶ GraphQL Interface ▶ Reporting Studio ▶ Smart Client ▶ Smart Server ▶ Device Management Interface components **Attention:** The Report Engine 14 cannot be installed in parallel with a Report Engine version 10, 11 or 12 on the same computer. If one of these versions already exists on the system, a corresponding message is displayed when you select it in the setup. The installation of Report Engine 14 is only possible after the uninstallation of Report Engine 10, 11 or 12. Versions of **Analyzer 3.40** or older can be installed in parallel with Report Engine 14.
IIoT Services	Installs IIoT Services . **Preselected:** ▶ IIoT Services ▶ License Manager **Optional:** ▶ Engineering Studio ▶ Service Engine ▶ Report Engine ▶ GraphQL Interface ▶ Reporting Studio ▶ Smart Client ▶ Smart Server ▶ Device Management Interface components
Smart Client	Installs Smart Client. **Preselected:** ▶ **Smart Client** ▶ License Manager

Package	Components
	Optional: - Engineering Studio - Service Engine - Report Engine - GraphQL Interface - Reporting Studio - Smart Server - IIoT Services - Device Management Interface components
Smart Server	Installs Smart Server. **Preselected:** - Smart Server - License Manager **Optional:** - Engineering Studio - Service Engine - Report Engine - GraphQL Interface - Reporting Studio - Smart Client - IIoT Services - Device Management Interface components

4.1.3 Configuration and installation

All packages can be installed with the recommended components by means of a click. You also have the possibility to adjust the installation packages individually.

DEFAULT INSTALLATION

To install a package with the preselected components:

1. Select the desired package.
2. Click on the **Install** button.

   The installation is started. A restart may be necessary during installation. If the installation process is not automatically continued after a restart, restart the zenon setup.
3. Follow the instructions of the wizard.

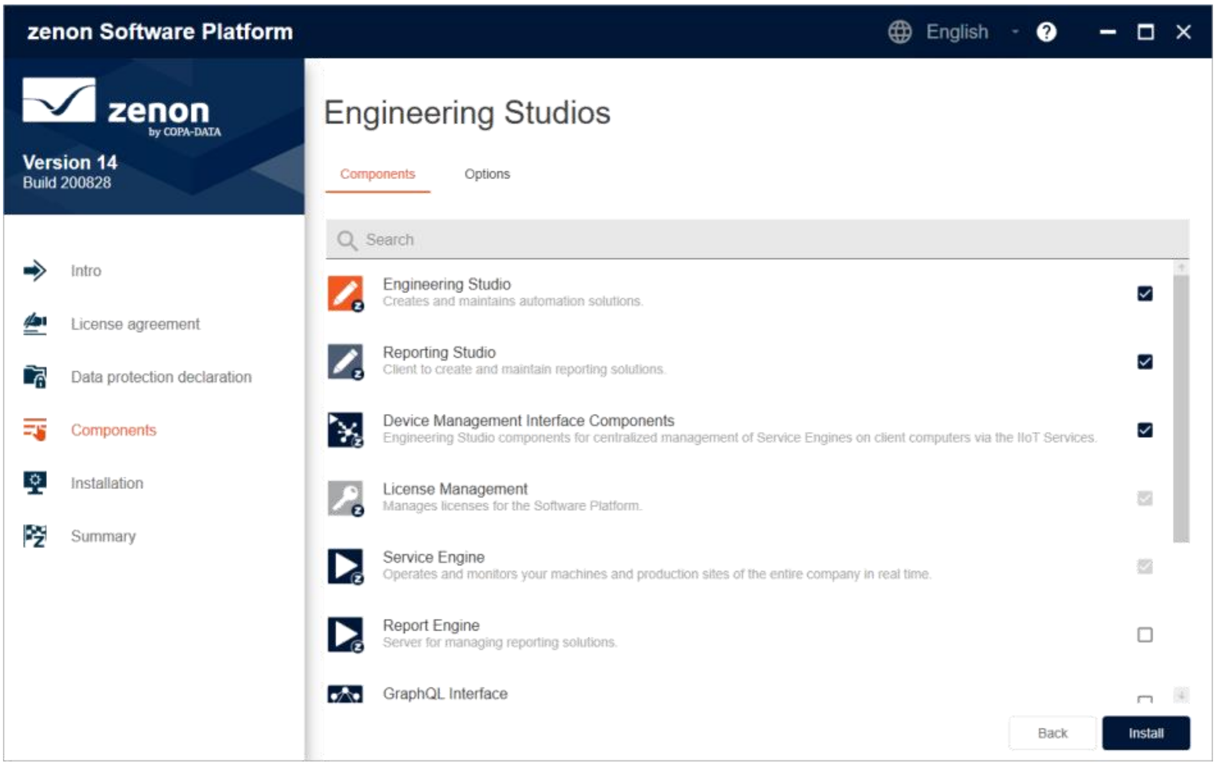
CUSTOMIZED INSTALLATION

You can adjust the packages individually.

To install a package with an adjusted installation:

1. Select the desired package.
2. Click on the **Customize** button.

   The dialog to adjust the installation is opened.
   The standard components have already been pre-selected.



Engineering Studio ☑
Creates and maintains automation solutions.

Reporting Studio ☑
Client to create and maintain reporting solutions.

Device Management Interface Components ☑
Engineering Studio components for centralized management of Service Engines on client computers via the IIoT Services.

License Management ☑
Manages licenses for the Software Platform.

Service Engine ☑
Operates and monitors your machines and production sites of the entire company in real time.

Report Engine ☐
Server for managing reporting solutions.

GraphQL Interface ☐

3. In the **Components** tab, select or deselect the desired components.
   Components can only be deselected if they are not required by another component.
   Already-installed components cannot be deselected.
   Information on the installation status and necessary requirements is available as a tool tip via the respective checkbox.

4. If necessary, configure the paths for installation in the **Options** tab.
   Paths can be selected for:
   - Engineering Studio
   - SQL Server databases of Engineering Studio and Report Engine

   Clicking on the button with the folder symbol opens the dialog for selecting the required folder. This configuration is only possible if no objects have been installed that require the installation location.
5. If necessary, select the **Harden installation** option in the **Options** tab.
   In this case, please note the additional required Configuration of the connections (on page 31) after installation.
   Note: You can only activate this option if there is no other version of the zenon Software Platform on the device.
6. Click on the **Install** button.

   The installation is started.
   A restart may be necessary during installation. If the installation process is not automatically continued after a restart, restart the zenon setup.
7. Follow the instructions of the wizard.

> ⚠**Attention**
>
> If there is already a version of IIoT Services on the system, it is strongly recommended that the data from this installation be backed up before an update. You can find information about this in the **Backup and Restore – Persistence Instance** section.

4.1.3.1 Hardening zenon

Activating the **Harden installation** option has the following effect:

- TCP/IP communication is restricted:
  - zenon only permits local access.
  - The SQL Server only permits local access.
  - The firewall rules only allow zenon services to have local access
- User permissions are restricted for the zenon project folders:
  - A new **SoftwarePlatformAccess** user group is created.
    Members of this group are:
    - the user currently logged in

- the **SYSTEM**
  user - the **LOCAL_SERVICE** user

- Access to the folders is restricted to members of the **SoftwarePlatformAccess** and **Administrators** groups.

Note: Setting permissions for files and folders takes effect immediately. The token for access by the user currently logged in is not updated. This means that the new permissions only take effect when the user logs in again. The setup requests a restart at the end of the installation to force the current user to log out and log in again.

Distributed engineering: Activating the **Harden installation** option also has an impact on **distributed engineering** in particular due to its restrictions. You must adjust the SQL Server, firewalls and connection to your requirements after installation. You can find information about the configuration of hardened systems in the Help in the **Security - distributed engineering with hardened systems** node.

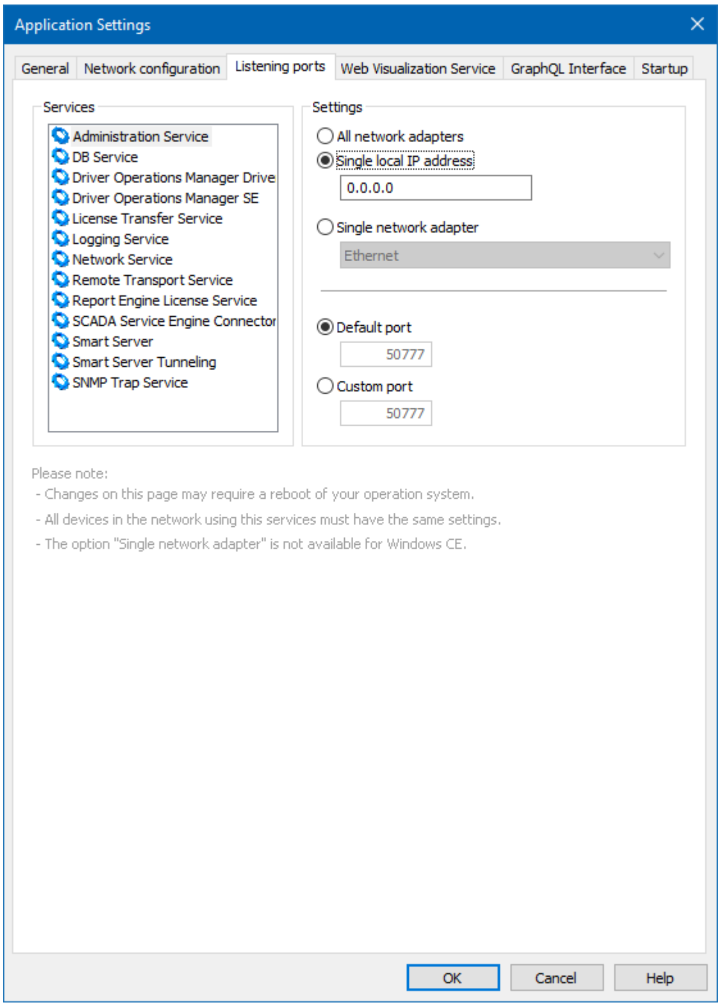
ADJUSTMENT IN THE STARTUP TOOL

When zenon is installed, exceptions are created in the Windows firewall by the setup. They are configured for applications and services that open a TCP Listening Port.

After installation, configure the exceptions in the Windows firewall to be more restrictive, suitable to their environment and the required apps and services.

On multi-homed systems with multiple network cards, zenon apps and services open the TCP Listening Port for all network cards present in the system by default. However communication throughout all network cards is often not necessary and not desirable.

After installing zenon, configure the TCP Listening Ports for the respective services and apps, according to their environment and requirements. Use the **Startup Tool** to do this. Only allow communication between the network card or IP address that is absolutely necessary for this. If you allocate a service to the Loopback adapter or the IP address *127.0.0.1*, you only allow local communication. This way, local Diagnosis Server access to local diagnosis clients can be limited, for example.



Services

- Administration Service
- DB Service
- Driver Operations Manager Drive
- Driver Operations Manager SE
- License Transfer Service
- Logging Service
- Network Service
- Remote Transport Service
- Report Engine License Service
- SCADA Service Engine Connector
- Smart Server
- Smart Server Tunneling
- SNMP Trap Service

Settings

○ All network adapters
● Single local IP address
0.0.0.0
○ Single network adapter
Ethernet

● Default port
50777
○ Custom port
50777

Please note:
- Changes on this page may require a reboot of your operation system.
- All devices in the network using this services must have the same settings.
- The option "Single network adapter" is not available for Windows CE.

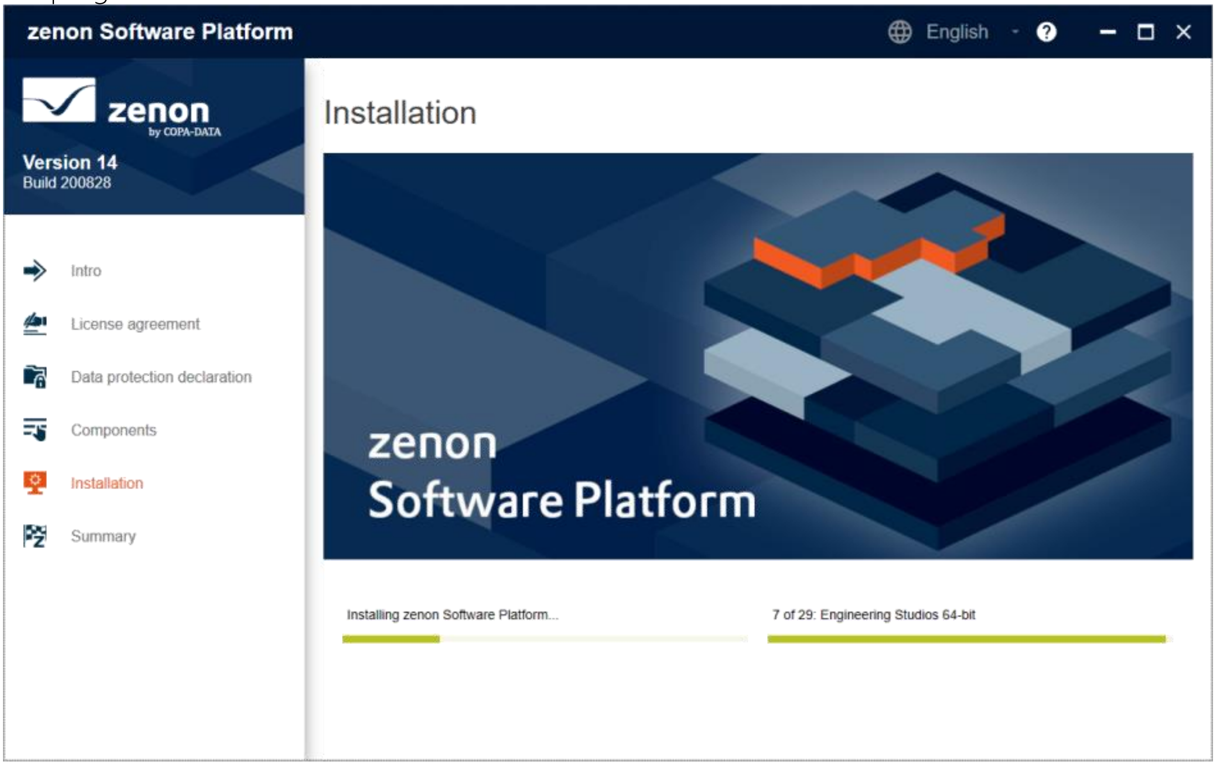
Note the following for platform functionalities that are independent of the configuration of the services:

Service	Platform functionality
Administration Service	Provides general functionalities for the operation of zenon.
DB Service	The database service is only needed by Engineering Studio. If you want to use **distributed engineering**, the service must be able to communicate via the network.
Driver Operations Manager	The **Driver Operations Manager** administers the local

Service	Platform functionality
	driver instances of Service Engine. It must be contactable from outside.
License Transfer Service	For **Remote Licensing** of zenon components, they must be able to communicate via the network. The service can also be limited again after concluding the licensing.
Logging Service	For **Remote Logging** via the **Diagnosis Viewer**, the service must be contactable via the network.
Network Service	This service must be reachable via the network for the use of : - zenon network - **Smart Client**
Remote Transport Service	Serves to manually transfer Service Engine files from an external instance of Engineering Studio. The service must be reachable via the network for this. With local instances of Engineering Studio, the service can only be operated locally.
Report Engine Licensing Service	For **Remote Licensing** of Report Engine, this service must be able to communicate via the network. The service can also be limited again after concluding the licensing.
SCADA Service Engine Connector	This service must be able to be contacted via the network if Report Engine, IIoT Services or remote Service Engine drivers: - are used with a remote instance - access online data of a Service Engine
Smart Server	This service is necessary for the operation of the **Smart Server**. Recommendation: Operate it as a separate instance, separate from Service Engine.
Smart Server Tunneling	This service is necessary for the operation of the **Smart Server**. Recommendation: Operate it as a separate instance, separate from Service Engine.
SNMP Trap Service	This service must be reachable via the network when using zenon SNMP services.

4.1.4 Installation and finishing

During installation, you are informed of the installation progress and the current installation stage in the progress bar.



Installing zenon Software Platform...

7 of 29: Engineering Studios 64-bit

The installation process may take some time. Do not turn your computer off in this time. Please also ensure that your computer is not automatically put into sleep mode.

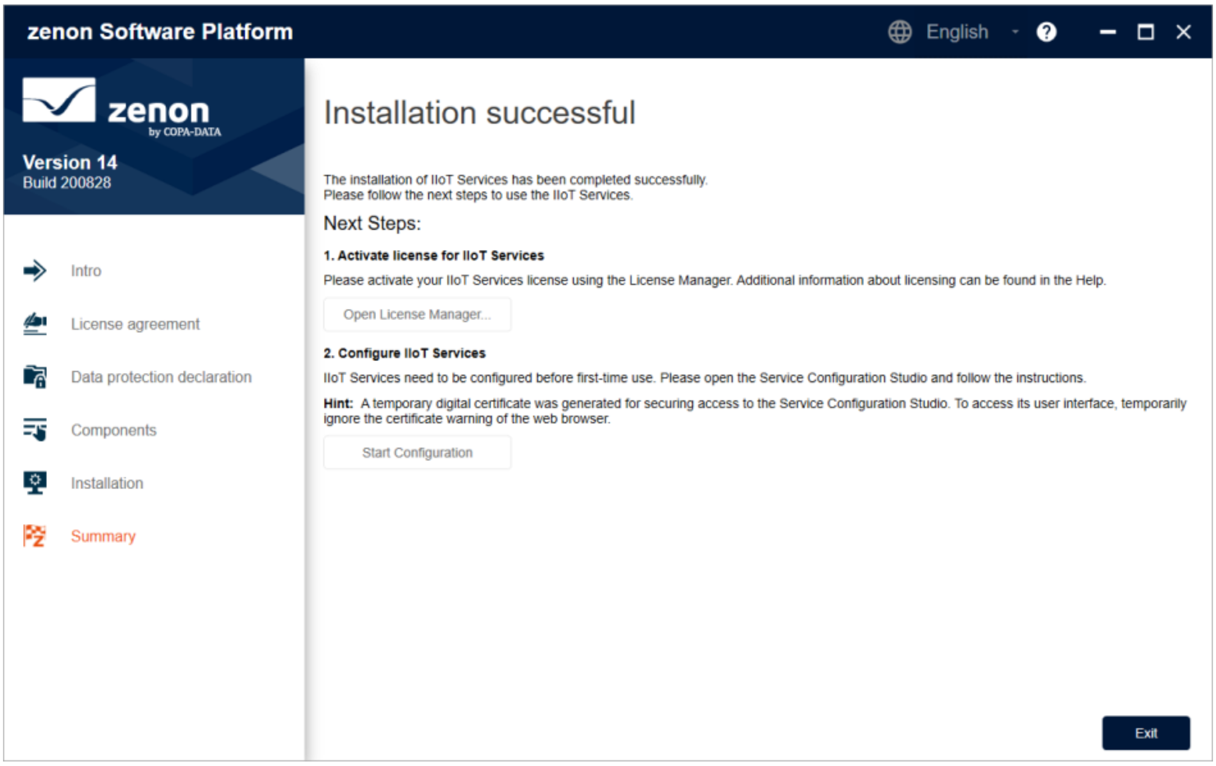
INSTALLATION IS COMPLETED

At the end of installation, you will receive a message about the success of the installation.

If IIoT Services has been reinstalled or updated:

1. Check or activate the license for the installed IIoT Services components using the **Open License Manager** button.

2. Configure the newly-installed IIoT Services using the **Start Configuration** button.
   Existing installations of IIoT Services can thus be adapted as required.



Installation successful

The installation of IIoT Services has been completed successfully.
Please follow the next steps to use the IIoT Services.

Next Steps:

1. Activate license for IIoT Services

Please activate your IIoT Services license using the License Manager. Additional information about licensing can be found in the Help.

2. Configure IIoT Services

IIoT Services need to be configured before first-time use. Please open the Service Configuration Studio and follow the instructions.

Hint: A temporary digital certificate was generated for securing access to the Service Configuration Studio. To access its user interface, temporarily ignore the certificate warning of the web browser.

CANCEL INSTALLATION

You can cancel the installation by clicking on the **Cancel** button.
Before canceling the installation there is a security query.
Possible actions:

- Yes: The installation is canceled.
  The dialog for an invalid installation is shown.
- No: The installation is continued.

CANCELED OR INCORRECT INSTALLATION

If an error occurs during installation or the installation was canceled, this is shown in a dialog.
You have the option of creating a detailed log with the **System Information Collector**. You can also send this log to COPA-DATA Support if needed.

Procedure:

1. Click on the **Collect system information for Support** button.

   The **System Information Collector** is started. Relevant data is collected.
   You will receive information on where the file was saved.
2. Use the information from the System Information Collector file for a reinstallation.

3. If necessary, send the file to your COPA-DATA support.
4. Attempt the installation again.

POSSIBLE CAUSES OF THE ERROR:

Important possible causes for a cancellation:

- A pending update to the Windows operating system.
- The SQL server required for Engineering Studio could not be installed.

4.1.5 Backup and Restore - Persistence Instance

The following applies for the Persistence Instance:

- The Persistence Service should be backed up before every update of the IIoT Services. This is a precaution.
- A restore of the backup is only necessary in rare cases. This is the case, for instance, if a problem occurs during an update.

You can find detailed information on this in the **Backup and Restore – Persistence Instance** node.

4.2 Installing additional components

You can install components of the zenon software platform that you have not yet installed at any time, via the Setup.

To do this:

1. Start the Setup for the zenon software platform.

   Because there are already components on the system, the page with the available components is opened immediately.
2. Select the desired components by clicking on the respective checkbox.
   If a further component is needed for a component, this is automatically selected too.
3. Click on **Install**.

The selected components are installed.

Note: You uninstall components using the Windows **Apps** app or the Control Panel.

4.3 Uninstalling components

To uninstall the zenon software platform or individual components:

Uninstall complete software platform:

- Open the Windows **Apps** settings.
- Click on the **zenon Platform Setup** entry.
- Click on **Uninstall**.
- Confirm the confirmation prompt.

All components of the zenon software platform are uninstalled.

Uninstall components:

- Open the Windows **Apps** settings.
- Click on the **zenon Platform Setup** entry.
- Click on **Change**.

  The dialog to select the components is opened.

  - Select the desired components.
  - Click on Uninstall.

The selected components are uninstalled.

BEHAVIOR WITH DIFFERENT VERSIONS

For components of the zenon Software Platform that were installed in different versions or upgraded to a higher version:
Components can only be changed in the version in which they were installed or to which they were upgraded.

Example:

Components of versions 11 and 12 are available on the system.

- The Report Engine was installed in version 11. An upgrade to version 12 has not been made. This means that it can only be removed via the version 11 installer.
- The Report Engine was installed in version 11. It was then upgraded to version 12. This means that it can only be removed by the version 12 installer.

When calling up the Windows functionality for modifying and uninstalling apps, only those components that have been installed with or updated to the selected version are offered for modification for the zenon Software Platform.

4.4 Logic Service for Windows (standalone installation)

On the installation medium, in the *%AdditionalSoftware%\COPA-DATA* Logic Service directory, you will find the installation packages for a standalone installation for **Logic Service for Windows**.

LOGIC SERVICE - CONTENTS OF THE INSTALLATION PACKAGE

With the standalone setup for **Logic Service for Windows** all components for operating the 61131-3 compliant Logic Service are installed on the target system.This includes among other things, components for licensing and diagnostics.No configuration components are installed.

INSTALLATION REQUIREMENTS

Keep in mind the general system requirements for installing the product. Pre-installation of **zenon Operator/Supervisor** or **Logic Service for Windows (Standalone)** is not permitted.

The product requires software already installed on the target system. You can also find them in the *%AdditionalSoftware%* folder of the installation medium. Therefore, if necessary, manually install the following packages:

- **Microsoft Visual Studio C++ Redistributables**
- **WIBU-SYSTEMS CodeMeter Runtime Kit**

After installation, execute the appropriate installation package (x86 or x64) for your target system.

> **Information**
>
> The installation does not include a license for the product. Therefore Logic Service starts in test mode. Licensing can be done using the general licensing tools.

UPDATE

To update an already installed version, uninstall it and perform a new installation.

4.5 Silent installation and uninstallation

zenon can also be installed and uninstalled silently (Silent Installation) and (Silent Remove).

As part of Silent Installation, it is possible to exclude certain standard components from the installation specifically:

- Firewall rules: The **CDPROP_INSTALLFIREWALL** parameter decides whether rules for the firewall are set.
- **Codemeter** Software: Can be configured using the *PREREQUISITES_* argument for the **ISFeatureInstall** parameter. Codemeter is not installed if the argument is not used.
  Attention: This argument is applicable for all *Prerequisites*. Other Prerequisites are also not installed in this case!

> **Information**
>
> All zenon versions from 7.10 on support silent installation and uninstalling.

PASSWORD CONVENTIONS FOR SA USERS

The random password created during a standard installation for the user *SA* on the SQL Server can be replaced with your own password. To do this, use an individual password for the argument **CDP_SQLADMINPW=** (Version 10 onwards) or **CDPROP_SQLADMINPASSWORD=** (before version 10).

Rules:

- Default length: 20 characters
- Permitted characters:
  - Letters: *A - Z, a - z*
  - Digits: *0 - 9*
  - Special characters: *!@$?#%&**
- Composition:
  - at least 1 lower case letter
  - at least 1 upper case letter
  - at least 1 number
  - at least 1 special character

> **Attention**
>
> A user-defined password is not validated.

4.5.1 As of version zenon 10

Initiation of silent *silent* installation can be carried out with the following parameters:

- **/***silent -silent /s -s*
- **/quiet -quiet /q -q**

For example:

- **SoftwarePlatform.exe** *-s* **CDP_WORKLOAD="***WlSmartClient*"
- **SoftwarePlatform.exe** */quiet* **CDP_WORKLOAD=**"*WlFullSoftwarePlatform*"
- **SoftwarePlatform.exe** */silent* **CDP_WORKLOAD="***WlEdgeServices*"

CDP_WORKLOAD does not make sense for installations that are not *silent*. The argument is therefore also not supported and is ignored.

Parameters	Arguments	Description
softwareplatform.exe		Call-up of the installation.
/silent		Silent installation.
CDP_WORKLOAD=	Workload to be installed. Must correspond to the ID from **WorkloadsSetup.config**. Example: *WlSmartClient* for a SmartClient installation. **Attention:** Workloads only, no features! Incorrect IDs or IDs that do not exist lead to the installation being aborted.	Entry is mandatory for *silent*. Is ignored with non-*silent*. Name and path of the configuration file can be defined individually with the **CDP_WORKLOADCONFIG_PATH=** parameter.
CDP_WORKLOADCONFIG_PATH=	Path to the storage location. Example: **CDP_WORKLOADCONFIG_PATH=**" *C:\Users\User\Desktop\My Workload\workload.config*"	Optional parameter. Specifies an individual storage location and name of the workload configuration file. If the parameter is not specified or the file does not exist in the specified path, the default file **WorkloadSetup.config** is used.
CDP_INSTALLFIREWALL=	▸ *1*: is installed ▸ *0*: is not installed Default:*1*	Information on whether firewall rules are installed.
CDP_INSTALLDEMO=	▸ *1*: is installed ▸ *0*: is not installed Default:*1*	Information on whether the demo project is also to be installed. Is only transferred to MSI.

Parameters	Arguments	Description
CDP_LANGUAGE=	- *1031*: German - *1033*: English - *1034*: Spanish - *1036*: French - *1040*: Italian - *1041*: Japanese - *1042*: Korean - *1049*: Russian Default:*1033*	Selection of the language. Is only transferred to MSI.
CDP_SQLPATH=	Path to SQL. *Empty*: Standard path Default: *Empty*	Path for SQL installation, as in GUI.
CDP_INSTALLDIR=	Path to 64-bit zenon installation folder. Default: default installation path	zenon software platform installation folder for 64-bit components
CDP_SQLADMINPW=	Any desired password. Must comply with SQL guidelines	SQL administrator password.
CDP_POSTINSTALLEXE=	Whether **PostInstall.exe** is executed after installation: - *1*: is executed - *0*: is not executed Default:*0*	
CDP_POSTINSTALLARGS =	Default:*empty*	Arguments for **PostInstall.exe**.
CDP_SERVICEHUB_PW=	Any desired password. *ServiceGrid*	Password for **Certificate Management**.
CDP_SQLADMINPW=	Any desired password.	Password for SQL server instance of Service Engine.
CDP_SQLADMINPW_REPORTING=	Any desired password.	Password for SQL server instance of Report Engine .

Parameters	Arguments	Description
CDP_NOREBOOT=	<u>**Values:**</u> - *1*: Restart is not performed. After installation, **SucesssReboot (3010)** is output. The new start must be carried out manually. - *0*: Restart is performed.	Configuration of whether the computer is restarted. Is used when updating IIoT Services.
UPGRADEIIOTSERVICES	Example: **BuildUpdate /silent /iacceptlicenseagreement /upgradeiiotservices**	Performs update of IIoT Services during build update.

SILENT UNINSTALLATION

Uninstallation must be carried out using the same **SoftwarePlatform.exe** that was used for installation. Because this is saved in a folder with an execution-specific GUID, the following lines are also logged with each successful installation.

"For uninstalling of the currently installed producty via CommandLine use:"
"For silent uninstall C:\\ProgramData\\Package Cache\\{bundleProviderGuid}\\SoftwarePlatform.exe /silent /uninstall"
"For uninstall via UI C:\\ProgramData\\Package Cache\\{bundleProviderGuid}\\SoftwarePlatform.exe /uninstall"

In doing so, **{bundleProviderGuid}** is always replaced with the execution-specific GUID. The full path to the EXE is thus given in the LOG file.

In principle, all actions are documented in the log. Certain queries that are displayed as GUI feedback during normal installation are written here in the LOG file.

4.5.2 zenon 7.20

Instigation of silent installation for version 7.20.

Syntax: **scada.exe /silent /language:***[number]* **CDPROP_EDITION=***[edition]* **CDPROP_TYPE=***[type]* **ISFeatureInstall=***[features]*

<u>**Examples:**</u>

- Installation of Engineering Studio, German, Energy Edition:
  scada.exe /silent /language:1031 CDPROP_EDITION=ENERGY CDPROP_TYPE=ED ISFeatureInstall=PREREQUISITES_EDITOR,SCADA

- Installation of Service Engine, English, Supervisor Edition:
  scada.exe /silent /language:1033 CDPROP_EDITION=SUPERVISOR CDPROP_TYPE=RT ISFeatureInstall=PREREQUISITES_RUNTIME,SCADA
- Installation of **Smart Server**, German:
  scada.exe /silent /language:1031 ISFeatureInstall=PREREQUISITES_WEBSERVER,WEBSERVER
- Installation of **Smart Client**, German:
  scada.exe /silent /language:1031 ISFeatureInstall=PREREQUISITES_WEBCLIENT,WEBCLIENT

PARAMETERS

Parameters	Arguments	Description
scada.exe		Call-up of the installation.
/silent		Silent installation.
/language:	*1031*: German *1033*: English *1034*: Spanish *1036*: French *1040*: Italian *1041*: Japanese *1042*: Korean *1049*: Russian	Selection of the language. English Example: **language:** *1033*
CDPROP_EDITION=	*ENERGY* *SUPERVISOR* *OPERATOR* *PHARMA*	Selection of the edition. Example: Energy Edition: **CDPROP_EDITION=** *ENERGY* Is not needed for **Smart Server** and **Smart Client**.
CDPROP_TYPE=	*ED*: Engineering Studio and Service Engine *RT*: Service Engine	Selection Engineering Studio or Service Engine. Example Service Engine: **CDPROP_TYPE=***RT* Is not needed for **Smart Server** and **Smart Client**.
CDPROP_INSTALLFIREWALL	*0* or *1*	Denotes whether rules for the firewall have been created:

Parameters	Arguments	Description
		- *0*: Rules are not created - *1*: Rules are created
CDPROP_SQLADMINPASSWORD=	User-defined password You can find further information on passwords in the **Password conventions for SA users** node.	Password for the *SA* user in SQL Server. This password is created for the *SA* user and used during installation. Example: **CDPROP_SQLADMINPASSWORD=***H1342DFAhzgs§*464578* If no password is transferred, a random password is generated during installation. **Attention:** User-defined passwords are not validated for validity and compliance with password rules!
ISFeatureInstall=	- *PREREQUISITES_EDITOR,SCADA*: Engineering Studio - *PREREQUISITES_RUNTIME,SCADA*: Service Engine - *PREREQUISITES_WEBSERVER,WEBSERVER*: Web Server - *PREREQUISITES_WEBCLIENT,WEBCLIENT*: Web Client	Selection of features to be installed. Arguments: - *PREREQUISITES_*: Decides whether Prerequisites are installed. The reasons why Prerequisites are installed is given after the underscore. E.g.: **EDITOR**. **Codemeter** is not installed if the argument is left out. - *SCADA*: Installs Engineering Studio and/or Service Engine, depending on the parameters for **CDPROP_TYPE=**. - *WEBSERVER*: Installs the Web Server.

Parameters	Arguments	Description
		- *WEBCLIENT*: Installs the Web Client. Examples: - Service Engine with Prerequistes: **ISFeatureInstall**=*PREREQUISITES_RUNTIME,SCADA* - Service Engine without Prerequistes: **ISFeatureInstall**=*SCADA*

SILENT UNINSTALLATION AS OF ZENON 7.20

The **GUID** is part of the path and depends on the version. The attendant version is visible in the file properties of a **GUID**.

Without LOG file:

- Path: **%ProgramFiles(x86)%\InstallShield Installation Information\{GUID}**
  Example: **C:\Program Files (x86)\InstallShield Installation Information\{9BE6EDFE-3465-486F-87EE-1C439DE5EA9A}**
- Syntax: **SCADA.exe /remove /silent**

With LOG file:

- Path: **%ProgramFiles(x86)%\InstallShield Installation Information\{GUID}**
  Example: **C:\Program Files (x86)\InstallShield Installation Information\{9BE6EDFE-3465-486F-87EE-1C439DE5EA9A}**
- Syntax: **SCADA.exe /remove /silent /log"%TEMP%"**

4.5.3 zenon 7.10 and 7.11

Syntax: **scada.exe /silent /language:[number] CDPROP_EDITION=[edition] CDPROP_TYPE=[type] ISFeatureInstall=[features]**

Parameters	Arguments	Description
scada.exe		Call-up of the installation.
/silent		Silent installation.
/language:	- **1031**: German	Selection of the language.

Parameters	Arguments	Description
	- **1033**: English - **1034**: Spanish - **1036**: French - **1040**: Italian	Example: English: **language:1033**
CDPROP_EDITION=	- **ENERGY** - **SUPERVISOR** - **OPERATOR** - **PHARMA**	Selection of the edition. Example: Energy Edition: **CDPROP_EDITION=ENERGY** Is not required for **Smart Server**.
CDPROP_TYPE=	- **ED**: Engineering Studio and Service Engine - **RT**: Service Engine	Selection Engineering Studio or Service Engine. Example Service Engine: **CDPROP_TYPE=RT** Is not required for **Smart Server**.
ISFeatureInstall=	- **WIBU,SCADA,MS,MSALL,SQL,COMMON**: Engineering Studio - **WIBU,SCADA,MS,MSALL,COMMON**: Service Engine - **WIBU,WS,MSALL,COMMON**: Web Server	Selection of features to be installed. Example Engineering Studio: **ISFeatureInstall=WIBU,SCADA,MS,MSALL,SQL,COMMON**

Examples:

- Installation of Engineering Studio, German, Energy Edition:
  scada.exe /silent /language:1031 CDPROP_EDITION=ENERGY CDPROP_TYPE=ED ISFeatureInstall=WIBU,SCADA,MS,MSALL,SQL,COMMON
- Installation of Service Engine, English, Supervisor Edition:
  scada.exe /silent /language:1033 CDPROP_EDITION=SUPERVISOR CDPROP_TYPE=RT ISFeatureInstall=WIBU,SCADA,MS,MSALL,COMMON
- Installation of **Smart Server**, German:
  scada.exe /silent /language:1031 ISFeatureInstall=WIBU,WS,MSALL,COMMON

SILENT UNINSTALLATION IN ZENON VERSION 7.10 AND 7.11

ZENON 7.10

Path: **C:\Program Files (x86)\InstallShield Installation Information\{860C41F0-6034-4822-BCF1-88D4849AE897}**

Syntax: **SCADA.exe /remove /silent**

ZENON 7.11

Path: **C:\Program Files (x86)\InstallShield Installation Information\{ED00D319-77B8-4C58-8D67-2DA2D48E90DB}**

Syntax: **SCADA.exe /remove /silent**

4.6 Error treatment

CHECK BEFORE INSTALLATION:

The system requirements are checked before installation. If the requirements are not met, you are shown these on a separate page with notices on how to rectify this.

ERROR DURING INSTALLATION

You will receive an error message if there are errors during installation.

If you need help from the Technical Consulting department of COPA-DATA:

1. If possible, create a screenshot of the error message
2. Navigate to the *%Temp%/SoftwarePlatform* folder.
3. Here you can find the log files of the installation.
4. Create a ZIP file with the content of the folder.
5. Forward the file and the screenshot to support@copadata.com

If you have already closed the error message window, you can find the log files with the installation information for the SQL Server in the folder:
C:\Program Files\Microsoft SQL Server\150\Setup Bootstrap\LOG

Tip: The file **summary.txt** provides information for troubleshooting.

> **Information**
>
> **Firewalls:** zenon automatically configures the firewall installed with Windows during installation. Firewalls from other providers must be properly configured by the user

FREQUENT SOURCES OF ERROR DURING INSTALLATION:

- The virus scanner is active and blocks the installtion because the scanner thinks it´s a virus. Solution: Seperate the system from the network, disable the virus scanner, execute the installation again.
- The firewall was not configured correctly. Solution: Seperate the system from the network, disable the firewall, execute the installation again.
- Erroneous SQL-installation on the system. Solution: Create project backups; if possible, deinstall SQL server, rename the SQL folder, and restart installation.

4.7 Windows Updates

Attention: Automatic Windows updates influence the installation

If an update of the Windows operating system is carried out while the zenon setup is running, it can cause problems during the zenon installation.

To prevent this:

- Deactivate the automatic Windows update during the time of installation.
- carry out the Windows update before starting the zenon installation

4.8 Virus scan

Anti-virus software can slow down or even prevent the installation of zenon.

Note: If the anti-virus software you use leads to problems during installation, deactivate the anti-virus software for the duration of the installation. The computers concerned can be exposed to higher risks during this time. Activate your anti-virus software immediately after the installation of zenon.

4.9 File Structure

The special file structure is created or extended during the installation.

The zenon program files are stored in a folder that is specified during installation.

Additionally the installation asks for a folder for the SQL databases of the projects. The storage medium for project archiving (SQL, screens etc.) must have enough free space, because all current and future project data is stored there.

Folder	Path
Program folder	32-bit system: *%ProgramFiles(x86)%\COPA-DATA\zenon Software Platform 14* *%CommonProgramFiles(x86)%\COPA-DATA* 64-bit system: *%ProgramFiles%\COPA-DATA\zenon Software Platform 14* *%CommonProgramFiles%\COPA-DATA*
Program data folder, for example global symbols, print templates, LOG files, etc.	*%ProgramData%\COPA-DATA\zenon1400*
Database folder (SQL)	*%ProgramData%\COPA-DATA\SQL2022*
System folder	*%ProgramData%\COPA-DATA\System*
Settings Engineering Studio and profiles	*%Users%\UserName\AppData\Local\COPA-DATA\zenon*\Engineering Studio
Settings for Diagnosis Viewer	*%Users%\UserName\AppData\Local\COPA-DATA\zenon\DiagView*

DEFINITION SERVICE ENGINE FOLDER AND DATA FOLDER

SERVICE ENGINE FOLDER

Engineering Studio generates Service Engine files in the Service Engine folder, or they are transferred to this folder by means of Remote Transport. The Service Engine folder is created or updated when compiling a project in Engineering Studio. This folder can be configured in Engineering Studio with the **Service Engine folder** project property. With remote transfer, the Service Engine folder is defined in the Remote Transport settings.

DATA FOLDER

Service Engine saves all data files that were created in Service Engine such as alarm files, archive files etc. in the data folder. The data folder is created as a subfolder of the Service Engine folder by default. The folder is automatically assigned the name of the computer that Service Engine is running on. You can change this save location in the project properties (**General**/**Data folder**).

Hint: Never set the data folder to a removable device such as an USB stick or a network device. It is recommended that the data is recorded locally and backed up externally.

> ⚠**Attention**
>
> If the defined path does not exist or is not available, no more data is written from Service Engine. This means a complete loss of data. Service Engine can still be operated but must be restarted as soon as the path is available again. The availability of the folder can be checked via the system driver variable **[Systeminformation] Service Engine folder not available**.

4.10 Free ports

zenon and zenon Logic require certain communication ports to communication in the network. If other apps, for example an already-installed SQL Server, occupy these ports, communication from zenon can be affected. Many ports in zenon can be changed using the **Startup Tool** or properties in Engineering Studio.

This is how you check the port assignments:

1. Enter *netstat -a -n -o* in the command line.

   You can reach the command line in Windows:

   - by pressing the Windows-key and R
   - Enter *cmd* and confirm with **OK**
   - A DOS input window pops up
   - enter the command *netstat*
2. A list of all currently used TCP/IP and UDP ports will pop up.
3. Check the listening ports (states: *LISTEN*) if the process ID (PID) from the ports needed by zenon and zenon Logic corresponds with the apps from zenon and zenon Logic.

These PIDs can be read in the **Windows Task Manager**. To do this, open the **Windows Task Manager** and switch to the **Services** tab.

4. If another software uses these ports, reconfigure this software.
You can see which ports zenon and zenon Logic use in the **Table of port assignments by** zenon **and** zenon Logic. Here you can also see if ports can be customized in these apps.

PORT SETTINGS BY ZENON AND ZENON LOGIC

App	Description	Ports	Transport log
stratonrt[k].exe	Logic Service polling communication and Logic Studio.	▸ *1200-1210*	**TCP**
stratonrt[k].exe		▸ *4500-4510*	**TCP**
stratonrt[k].exe	zenon Logic Redundancy.	▸ *7000-7010*	**TCP**
stratonrt[k].exe	Logic Service - spontaneous communication.	▸ *9000-9010*	**TCP**
zennetsrv.exe	zenon Network service.	▸ *1100-1100*	**TCP**
zensyssrv.exe	zenon Transport service.	▸ *1101*	**TCP**
zendbsrv.exe	zenon Database service.	▸ *1103*	**TCP**
zenAdminsrv.exe	zenon Management service.	▸ *50777*	**TCP**
zenLogSrv.exe	zenon Logging service.	▸ *50780*	**TCP**
CodeMeter.exe	Code Meter Dongle service.	▸ *22350* (changeable but must not be changed)	**TCP**
WkSvW32.exe	WibuKey Network service.	▸ *22347* (fixed)	**TCP**
Zenrt32.exe	**Message Control** with Voice over IP.	▸ *5060*: SIP ▸ *4000*: RTP ▸ *4001*: RTCP (fixed) SIP and RTP can be configured using Engineering Studio. RTCP is automatically set by the	**UDP**

App	Description	Ports	Transport log
		system.	
zenHelpService.exe	zenon Help provisioning service	- *50790*	**TCP**

4.11 Install older version, if zenon 14 is already installed (64-bit operating system)

If, on a 64-bit operating system, after installation of zenon 14, a version of zenon with a version number lower than 7.10 is installed, the 64-bit services must be registered again after the installation via the command line. Registration can be carried out using a batch file or manually.

Registration with a batch file:

1. Copy the file named **Register.bat** from the zenon installation medium.
2. You can find this in the following folder: ...*\AdditionalSoftware\Register Admin Service and Log Service (x64)*
3. Execute the file on the respective computer as an administrator.

manual registration:

1. Run the command line with administrative rights
2. Go to the folder *%ProgramFiles%\Common Files\COPA-DATA\zenAdminSrv*
3. Start the service **zenAdminSrv.exe** with the parameter **-service**
4. Go to the folder *%ProgramFiles%\Common Files\COPA-DATA\zenLogSrv*
5. Start the service **zenLogSrv.exe** with the parameter **-service**

Example

zenAdminSrv:

- Folder: *C:\Program Files\Common Files\COPA-DATA\zenAdminSrv*
- Command: **zenAdminSrv.exe -service**

zenLogSrv:

- Folder: *C:\Program Files\Common Files\COPA-DATA\zenLogSrv*
- Command: **zenLogSrv.exe -service**

4.12 Installation of version 7.x and version 6.51 on the same computer

If a version 7.x is installed on a system that already has zenon 6.51 installed, the **Multiple Network Protocol Driver** must be reinstalled after a reboot.

FOR X64 SYSTEMS

For new installation:

1. Restart the system.
2. On the installation medium, open the following path: *AdditionalSoftware\COPA-DATA Multiple Network Protocol Driver*.
3. Execute the file called **MNDPx64Setup.bat**.

This means that the driver is reinstalled and properly linked to the network adapters.

FOR X86 SYSTEMS

For new installation:

1. Restart the system.
2. On the installation medium, open the following path: *AdditionalSoftware\COPA-DATA Multiple Network Protocol Driver*
3. Execute the file called **MNDPx86Setup.bat**

This means that the driver is reinstalled and properly linked to the network adapters.

4.13 Licensing preview versions

The following time-limited licenses are available for zenon preview programs. They have a fixed expiry date.
If a preview version is installed, it needs an internet connection. The licensing will fail if this is not present.
In this case, the timestamp can be updated manually.
To do this:

1. Open the command line.
2. Enter: *%programfiles(x86)%\CodeMeter\Runtime\bin\cmu32.exe --time-update*

The timestamp is updated and the license is valid.

4.14 Software and paths for installation and operation

Paths for zenon:

- Installation
- Engineering Studio
- Service Engine

> **Info**
>
> You can display many default paths with the help of the **set** command:
>
> - start the command line (enter **cmd** in the Windows start area)
> - enter command **set**
> - By pressing the **Enter** key, the default folder for Windows and zenon are displayed

Note: Because absolute paths can differ in different operating systems, the paths are always displayed as Windows environment variables in this node. For example *%ProgramData%* instead of *C:\ProgramData*.

INSTALLATION

During installation, paths are set for:

- Engineering Studio
- zenon SQL folder

Only the paths for Engineering Studio and zenon SQL folder can be adjusted.
The setup needs administrator rights. This is also true for changing the installation paths.

REQUIREMENTS

The installation paths of the required third-party software match the standard paths of the respective manufacturer and cannot be changed during setup.

The additional software packages that need to be installed depend on the type of installation:

- Engineering Studio
- Service Engine
- **Smart Client**
- Logic Service

Requirements	Engineering Studio	Service Engine	Smart Client	Logic Service
Microsoft .NET 8.0	X	X	X	--
Microsoft SQL Server 2022 Express (Bei Report Engine Standard Edition)	X	--	--	--
Microsoft Visual C++ 2022 Redistributable	X	X	X	X
Microsoft .NET Core Hosting Bundle 6.0.3	X	X	X	X
CodeMeter Runtime Kit 8	X	X	--	X
Microsoft Web Deploy 3.6	--	--	--	--
Visual Basic for Applications | VBA 7.1	X	X	X	--
Visual Basic for Applications Language Pack | VBA 7.1	X	X	X	--
COPA-DATA Multiple Network Protocol Driver	X	X	--	X

Key:

X: supported

--: not supported

Note: **Microsoft Visual Studio Tools for Applications** (VSTA 2.0) is not required and is not installed during setup. It can be installed separately using the **Startup Tool**. To do this, the **VSTA_AddOn.exe** file must be on the system.

ZENON

The installation of Engineering Studio sets two paths:

- Engineering Studio:
  - 32-bit systems: *%ProgramFiles(x86)%\COPA-DATA\zenon Software Platform [Version]*
  - 64-bit systems: *%ProgramFiles%\COPA-DATA\zenon Software Platform [Version]*

  These paths can be customized during the installation.
- zenon SQL folder:
  - Version 14 and higher: %ProgramData%\COPA-DATA\SQL2022\

  These paths can be customized manually via zenDB.ini.

ENGINEERING STUDIO

In Engineering Studio, the following paths are used by default:

Object	Path
Workspace	*%CD_USERDATA%* For example: *C:\Users\Public\Documents\zenon_Projects*
Projects	*%CD_USERDATA%* For example: *C:\Users\Public\Documents\zenon_Projects* **Hint for shortcuts:** highlight the project -> **Ctrl+Alt+D**
SQL folder of the project	*%ProgramData%\COPA-DATA\[SQL folder]\[GUID]\FILES* **Hint for shortcuts:** highlight the project -> **Ctrl+Alt+E**
project.ini	*%ProgramData%\COPA-DATA\[SQL folder]\[GUID]\FILES\zenon\system*
zenon6.ini	*%ProgramData%\COPA-DATA\System* For example: *C:\ProgramData\COPA-DATA\System*
Backup	*%ProgramData%\COPA-DATA\[SQL folder]\GUID]\BACKUP* *%ProgramData%\COPA-DATA\[SQL folder\GUID]\FILES\[Projects]*
Compiled files	*%CD_USERDATA%\[Workspace]\[Project]\RT*
External files	*%CD_USERDATA%\[Workspace]\[Project]\RT\FILES\...* **Note:** Can be set in the project using the file storage property.
System files	Windows system folder.

SERVICE ENGINE

In Service Engine, the following paths are used by default:

Object	Path
Projects	*%CD_USERDATA%\[Workspace]\[Project]\RT*
External files	*%CD_USERDATA%\[Workspace]\[Project]\RT\FILES\...*

Object	Path
Exported archives, Chronological Event List and Alarm Message List	*%CD_USERDATA%\[Workspace]\[Project]\Export* **Note:** Is created at the first export.
System files	Windows system folder.

ZENON LOGIC

Paths for zenon Logic are created equal to the zenon paths.

5 Version change and build updates

In zenon, you can change to new versions for example, from 10 to 11. Or you can install updates within a version.

- New versions can be installed in parallel with existing versions. They mainly offer new features.
- Build updates modify a previously installed version. They mainly fix bugs.

With each setup for new build or new versions, you receive a link to the COPA-DATA website with the changes between the previous version and the newly-installed one.

BUILD UPDATE

An update only updates those files which are more current than the previously installed files. All projects and individual settings will remain unchanged. Build updates can contain changes for all installed components. When calling up the setup, the components that have been changed and the version to which they have been changed are shown.

Note: Build updates may have lower quality assurance standards than release versions.

Note: Build updates can only be installed locally. Installation on network paths (UNC) is not possible.

VERSION CHANGE

If you want to install a new version of zenon, start the installation routine. The new version is being installed parallel to the old one. All projects and individual settings will remain unchanged. Projects aren´t converted to the new version during installation. The respective project is converted when it´s being opened for the first time in Engineering Studio. A dialog box notifies you about this procedure.

The old version is automatically backed up. If you want to use only the most up-to-date version, use the Windows control panel software deinstallation routine to remove the old version.

> ⚠**Attention**
>
> If an installation involves changing the SQL server (for example, from zenon 10 to zenon 11 or higher), all projects or the workspace must be backed up before the installation. This backup is read back after the installation. For details see also section **Multi-user projects/Update with change of SQL servers**.

If you want to use multiple versions of zenon simultaneously, you have to manage them using the **Startup Tool**. You can start only one version at a time. You can select which version you want to run using the **Startup Tool** that automatically adjusts all necessary settings. You can find details in the **Startup Tool** node.

> **Information**
>
> New versions always bring about structural changes. Projects and settings remain untouched during installation. If you open Engineering Studio for the first time, projects are converted to the new version. Simultaneously, an automatic backup of the old version is created.
>
> Converted projects cannot be edited in legacy versions. From version 6.2 on, Engineering Studio is able to generate projects for different Service Engine versions.
>
> Important tips for converting projects can be found in the revision text and in the **Project conversion** node.

MULTI-USER PROJECTS

To ensure a change to a new zenon version in multi-user projects without data loss:

1. Check in all checked out elements on all Clients by clicking *Apply changes*.
   Nothing must be checked out. This is true for all projects.
2. Install the new zenon version on the server computer.
3. Convert all server projects to the new version.
   To do this, load each project on the server computer into Engineering Studio and accept the conversion.
4. Install the new zenon version on the client computers.
5. Load the projects to the clients.

CHANGE THE SQL SERVER

If an installation involves changing the SQL Server (e.g. from zenon 8.20 to zenon 14), additional steps are needed.
These steps are carried out:

- after all projects are checked in
- before the new version is installed

Procedure when changing the SQL Server:

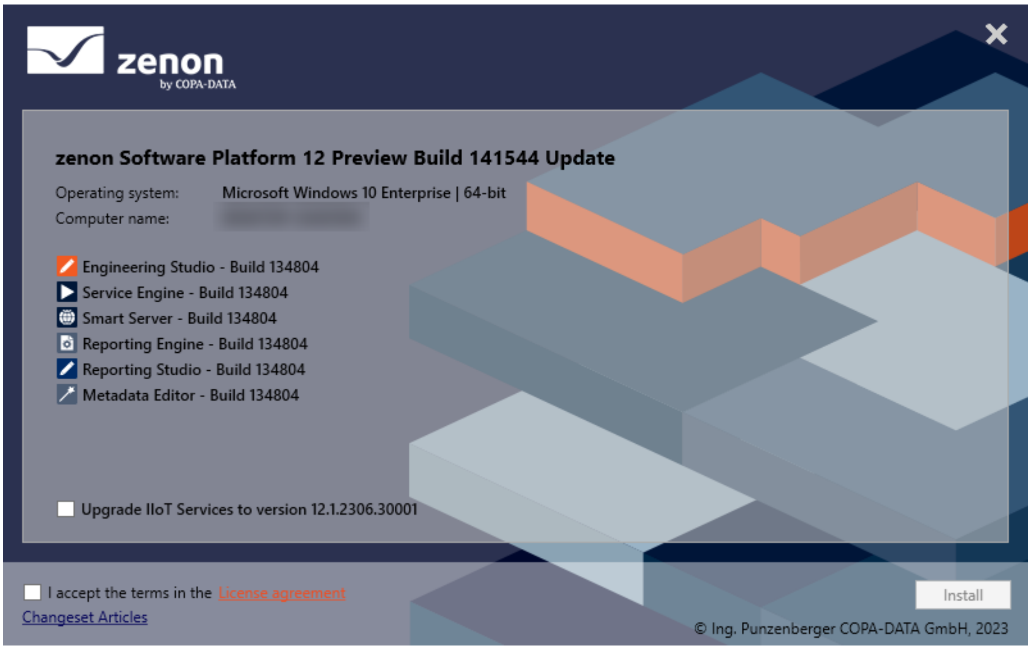
1. Check in all checked out elements on all Clients -> *Apply changes*.
2. On the multi-user server, open Engineering Studio in the original version.
3. Create backups of all projects which you want to edit or open with the new version:
   - either as single project backups
   - or as backup of the complete workspace
4. Install the new version on the Server.
5. Convert all Server projects to the new version by loading the previously created project backups one time in Engineering Studio.
6. Install the update on every Client.
7. Transfer the projects from the multi-user Server to the Clients
   Keep in mind the new name of the SQL instance.

The projects are converted, synchronized and ready for use.

Attention: Make sure that the settings of the firewall allows the data traffic between the multi-user Server and the Clients.

5.1 Build update IIoT Services

As of version 12, IIoT Services versions can also be updated when installing a build update.
In this case, an option for updating the IIoT Services version to the current version is offered when the setup is called.



zenon Software Platform 12 Preview Build 141544 Update

Operating system: Microsoft Windows 10 Enterprise | 64-bit
Computer name: [redacted]

Engineering Studio - Build 134804
Service Engine - Build 134804
Smart Server - Build 134804
Reporting Engine - Build 134804
Reporting Studio - Build 134804
Metadata Editor - Build 134804

□ Upgrade IIoT Services to version 12.1.2306.30001

□ I accept the terms in the License agreement
Changeset Articles

© Ing. Punzenberger COPA-DATA GmbH, 2023

Procedure:

- Checkbox for upgrading IIoT Services is deactivated:
  IIoT Services are not updated. They remain in the version already installed on the system.
- Checkbox for upgrading IIoT Services is activated:
  - IIoT Services are updated to the current version.
  - The task performs a silent installation of the IIoT Services setup without rebooting.
  - If the installation is successful, a button for manual reboot is offered at the end.
    Do a reboot before starting IIoT Services. This ensures the correct operation of all services.

5.2 Compatibility

Compatibility in zenon concerns:

- Service Engine: Cooperation of different Service Engine versions.
- Engineering Studio: Up-converting existing projects to new Engineering Studio versions.

- Engineering Studio: Creating Service Engine files for different Service Engine versions in Engineering Studio.

SERVICE ENGINE

Service Engine online compatibility enables Service Engine systems to work together in the zenon network, as well as via **Smart Clients**.

The following is applicable here: The version of the client Service Engine must be the same or higher than the version of the server Service Engine.
For example:

- An 8.20 client can work together with an 8.10 server.
- An 8.00 client does not work together with an 8.10 server. In this case, the Service Engine client must be upgraded to version 8.10 or higher.

Note: When using the server and standby server, the same zenon version must be used on both of them.

The current Service Engine can load projects of the following versions:

- *6.20 SP4*
- *6.21 SP0*
- *6.21 SP1*
- *6.22 SP0*
- *6.22 SP1*
- *6.50 SP0*
- *6.51 SP0*
- *7.00 SP0*
- *7.10 SP0*
- *7.11 SP0*
- *7.20 SP0*
- *7.20 SP0[current build no.]*
- *7.50 SP0*
- *7.60 SP0*
- *8.00 SP0*
- *8.10 SP0*
- *8.20 SP0*
- *10.00*

- *11.00*
- *12.00*
- *14.00*

Due to the multi-project administration, projects from different versions can be loaded. For example, the integration project can be version 8.20, a subproject from version 8.10 and another subproject from version 7.60.

ENGINEERING STUDIO

Engineering Studio can open projects from the previous versions in each new version. These can be edited further in the new version. If adjustments are required by the user, information can be found in the current revision text and in the **Project conversion** node. When opening a project with a lower version number in a higher Engineering Studio version:

- the project is automatically converted
- a backup of the project is automatically created

> ⚠**Attention**
>
> There is no backward compatibility between Engineering Studio versions. Backward compatibility is only ensured between Engineering Studio and Service Engine.
>
> That means:
>
> - Converted projects can no longer be opened in a Engineering Studio with a lower version number.
> - The project backup created during conversion can still be opened and edited.
>
> Also avoid transferring projects via XML import from newer to older versions. This can lead to undesirable results. Drivers in particular can perform differently than expected.

COMPATIBILITY BETWEEN ENGINEERING STUDIO AND SERVICE ENGINE

With Engineering Studio, Service Engine files can be created for different versions of Service Engine. The Service Engine version therefore does not need to correspond to the Engineering Studio version. This backward compatibility is particularly suited for use of mixed systems.

For example: A project that has been configured with Engineering Studio version 10.00 and compiled for 8.00, can also be started with Service Engine8.00.

⚠Attention

If possible, the same version of Engineering Studio and Service Engine should always be used. Configurations of properties that are not available in older versions can lead to unwanted results in older versions of Service Engine.

CREATE SERVICE ENGINE FILES

To create Service Engine files for earlier versions in Engineering Studio:

1. Select the project in the project tree.
2. Navigate to the **General** section in project properties.
3. Open the **Create Service Engine files for** property drop-down list.
4. Select the desired version from the drop-down list

Attention: In order to ensure consistency of Service Engine files, all Service Engine files must be newly created each time this property is changed. The configurations for all drivers are converted. Settings that do not exist in the configured version are set to the default setting.

ERROR CREATING SERVICE ENGINE FILES AND MICROSOFT OFFICE 365

In certain configurations, an error may occur when creating Service Engine files:

- You can create Service Engine files on a computer with:
  - Windows 10
    operating system and
  - Office 365.
- The creation of Service Engine files has failed and is ended with an error message.

This is caused when an incorrect version of a program library by VBA is loaded.

Solution:

1. Go to the folder: *%AppData%\Microsoft\FORMS*.
2. Delete the file **zenone32.box**.
3. This file is created new automatically by the zenon.

The creation of Service Engine files is possible again.

XML

Data exported in XML is then available for import into later Engineering Studio versions.

Exception: If data from the RGM is saved in Service Engine directly as an XML file using the export function, this cannot be reimported.

Recommendation: Avoid transferring projects via XML import from newer to older versions. This can lead to undesired events in Engineering Studio and Service Engine.

6 Installation of Service Engine

The procedure for putting Service Engine into operation for the first time depends on whether the device you are operating is a turn-key device with Service Engine installed or a device on which Service Engine must be installed. If Service Engine is already installed on the device, only the desired project needs to be copied to the Service Engine device.

To do this:

- Save your project files on an external data medium.
- Use Drag&Drop or Copy&Paste to transfer these to your Service Engine device.

or

- Get the project from the server on the Service Engine device.
- Create a Client using the network topology.

Note: This is also possible by means of remote transport.

If Service Engine is not yet installed on the device, it must first be installed. You can read how this works in the zenon standard installation (on page 24) node.

A license is required after installation of Service Engine in order to put it into operation. The license is transferred with the License Manager.

Note: The storage medium of the device you are operating may stop working after many years of use in a harsh industrial environment. To avoid loss of data, it may be a good idea to ensure that there is a redundant environment, for example working with both a Server and a Standby Server.

6.1 Installation

For the installation of Service Engine, start the installation process from your zenon installation medium.

This process automatically starts when the zenon installation medium is connected and guides you through the whole installation. Alternatively, it is possible to start the installation by executing the **START.exe** file in the root folder of your zenon installation medium.

Information

You can find further information in the Installation and updates (on page 6) node in the zenon standard installation (on page 24) node.

6.2 System requirements and operating systems

A detailed overview of supported operating systems and the required software and hardware can be found in the installation and updates (on page 6) node.

6.2.1 System requirements when using DirectX

The following minimum requirements must be met when using *DirectX hardware* or *DirectX software*:

Note: For extensive projects or several projects loaded at the same time you will need accordingly faster/stronger hardware. The minimum requirements can increase as a result of this.

Parameter	Minimum requirements	Recommended
CPU:	Single core with SSE2 support.	Quad Core or more cores
Graphics adapter: (DirectX hardware only)	DirectX 11 mainstream graphics card. **Note:** When an integrated graphics chip is used in particular, it is possible, depending on the driver used, that there are impairments to the display quality.	Dedicated DirectX 11 AMD or nVidia high-end graphics card
Graphics memory: (DirectX hardware only)	1 GB VRAM **Note:** The size that is actually needed depends on the number of screens called up and the elements displayed.	2 GB VRAM
Graphics card driver: (DirectX hardware only)	The graphics card manufacturer's most recent driver.	
Operating system:	*DirectX Hardware* and *DirectX Software* only work on operating systems with *DirectX 11.1* support.	

Parameter	Minimum requirements	Recommended
	If the system does not support *DirectX 11.1*, it is automatically switched to *Windows based* . The current *DirectX*-Service Engine must be installed. For zenon it is installed together with the setup. It must be manually installed for **Smart Client**.	

You can check the DirectX hardware compatibility of the graphics card and the driver with the **dxdiag.exe** of the Windows operating system.
As of Windows 8: All supported versions of DirectX are displayed in the **Display** tab under **Feature Levels**. For example, DirectX 11 is displayed as *11.0*.

6.2.2 File Structure

The special file structure is created or extended during the installation.

The zenon program files are stored in a folder that is specified during installation.

Additionally the installation asks for a folder for the SQL databases of the projects. The storage medium for project archiving (SQL, screens etc.) must have enough free space, because all current and future project data is stored there.

Folder	Path
Program folder	32-bit system: *%ProgramFiles(x86)%\COPA-DATA\zenon Software Platform 14* *%CommonProgramFiles(x86)%\COPA-DATA* 64-bit system: *%ProgramFiles%\COPA-DATA\zenon Software Platform 14* *%CommonProgramFiles%\COPA-DATA*
Program data folder, for example global symbols, print templates, LOG files, etc.	*%ProgramData%\COPA-DATA\zenon1400*

Folder	Path
Database folder (SQL)	*%ProgramData%\COPA-DATA\SQL2022*
System folder	*%ProgramData%\COPA-DATA\System*
Settings Engineering Studio and profiles	*%Users%\UserName\AppData\Local\COPA-DATA\zenon\*Engineering Studio
Settings for Diagnosis Viewer	*%Users%\UserName\AppData\Local\COPA-DATA\zenon\DiagView*

DEFINITION SERVICE ENGINE FOLDER AND DATA FOLDER

SERVICE ENGINE FOLDER

Engineering Studio generates Service Engine files in the Service Engine folder, or they are transferred to this folder by means of Remote Transport. The Service Engine folder is created or updated when compiling a project in Engineering Studio. This folder can be configured in Engineering Studio with the **Service Engine folder** project property. With remote transfer, the Service Engine folder is defined in the Remote Transport settings.

DATA FOLDER

Service Engine saves all data files that were created in Service Engine such as alarm files, archive files etc. in the data folder. The data folder is created as a subfolder of the Service Engine folder by default. The folder is automatically assigned the name of the computer that Service Engine is running on. You can change this save location in the project properties (**General**/**Data folder**).

Hint: Never set the data folder to a removable device such as an USB stick or a network device. It is recommended that the data is recorded locally and backed up externally.

> ⚠**Attention**
>
> If the defined path does not exist or is not available, no more data is written from Service Engine. This means a complete loss of data. Service Engine can still be operated but must be restarted as soon as the path is available again. The availability of the folder can be checked via the system driver variable **[Systeminformation] Service Engine folder not available**.

6.2.3 Free ports

zenon and zenon Logic require certain communication ports to communication in the network. If other apps, for example an already-installed SQL Server, occupy these ports, communication from zenon can be affected. Many ports in zenon can be changed using the **Startup Tool** or properties in Engineering Studio.

This is how you check the port assignments:

1. Enter *netstat -a -n -o* in the command line.

   You can reach the command line in Windows:

   - by pressing the Windows-key and R
   - Enter *cmd* and confirm with **OK**
   - A DOS input window pops up
   - enter the command *netstat*

2. A list of all currently used TCP/IP and UDP ports will pop up.
3. Check the listening ports (states: *LISTEN*) if the process ID (PID) from the ports needed by zenon and zenon Logic corresponds with the apps from zenon and zenon Logic.

   These PIDs can be read in the **Windows Task Manager**. To do this, open the **Windows Task Manager** and switch to the **Services** tab.

4. If another software uses these ports, reconfigure this software.
   You can see which ports zenon and zenon Logic use in the **Table of port assignments by** zenon **and** zenon Logic. Here you can also see if ports can be customized in these apps.

PORT SETTINGS BY ZENON AND ZENON LOGIC

App	Description	Ports	Transport log
stratonrt[k].exe	Logic Service polling communication and Logic Studio.	- *1200-1210*	**TCP**
stratonrt[k].exe		- *4500-4510*	**TCP**
stratonrt[k].exe	zenon Logic Redundancy.	- *7000-7010*	**TCP**
stratonrt[k].exe	Logic Service - spontaneous communication.	- *9000-9010*	**TCP**

App	Description	Ports	Transport log
zennetsrv.exe	zenon Network service.	- *1100-1100*	**TCP**
zensyssrv.exe	zenon Transport service.	- *1101*	**TCP**
zendbsrv.exe	zenon Database service.	- *1103*	**TCP**
zenAdminsrv.exe	zenon Management service.	- *50777*	**TCP**
zenLogSrv.exe	zenon Logging service.	- *50780*	**TCP**
CodeMeter.exe	Code Meter Dongle service.	- *22350* (changeable but must not be changed)	**TCP**
WkSvW32.exe	WibuKey Network service.	- *22347* (fixed)	**TCP**
Zenrt32.exe	**Message Control** with Voice over IP.	- *5060*: SIP - *4000*: RTP - *4001*: RTCP (fixed) SIP and RTP can be configured using Engineering Studio. RTCP is automatically set by the system.	**UDP**
zenHelpService.exe	zenon Help provisioning service	- *50790*	**TCP**

6.2.4 Exceptions for anti-virus

zenon needs a range of services and operations that anti-virus programs could categorize as potentially dangerous.

Examples of this:

- Communication with CodeMeter --> USB dongle
- Loading VBA or VSTA code
- Embedding ActiveX elements in screens
- Frequent file access, for example for archiving, CEL, AML...

Note: There are compatibility problems with *Ahnlabs V3* virus scanner. If zenon refuses to start and displays an error message, put V3 into *Game-Mode* or uninstall V3.

6.2.5 firewall setup

zenon uses a number of ports. These must be activated in security measures, such as a firewall, and accessible during operation.

PORTS USED

For communication within zenon, only TCP ports are used; no UDP ports are used. zenon requires the following ports in the network:

Service	File	Task	TCP port
Network service	**zenNetSrv.exe**	Service Engine communication.	*1100*
Transport service	**zenSysSrv.exe**	Data transfer by means of Remote Transport (Engineering Studio).	*1101*
Smart Server	**zenWebSrv.exe**	On-site logging machine between **Smart Client** and Service Engine.	*1102*

Port numbers can be changed individually via the **Listening ports** tab in the **Startup Tool**. In this case, the measuring range must be adapted manually.

Furthermore, zenon and Report Engine services use a range of default ports.

DEFAULT PORTS

ZENON

Application	Startup Tool naming	Standard port
Administration Service **Note:** Only local connections are permitted, regardless of the settings of the network card or IP address.	*zenAdminSrv.exe*	*50777*
DB Service	*zenDBSrv.exe*	*1103*
Driver Operations Manager Driver **Note:** Only local connections are permitted, regardless of the settings of the network	*zenDrvOpsManager.exe*	*50787*

Application	Startup Tool naming	Standard port
card or IP address.		
Driver Operations Manager SE **Note:** Only local connections are permitted, regardless of the settings of the network card or IP address.	*zenDrvOpsManager.exe*	*50786*
License Transfer Service	*zenLicTransfer.exe*	*50784*
Logging Service	*zenLogSrv.exe*	*50780*
Network Service	*zenNetSrv.exe*	*1100*
Remote Transport Service	*zenSysSrv.exe*	*1101*
Report Engine License Service	*zrsLicSrv.exe*	*50779*
SCADA Service Engine Connector	*zrsConnector.exe*	*50778*
Smart Server	*zenWebSrv.exe*	*1102*
Smart Server Tunneling	*zenWebSrv.exe*	*8080*
SNMP Trap Service	*zenSnmpTrapSrv.exe*	*50782*
Help Service	*zenHelpService.exe*	*50790*
SQL Browser Service, (for distributed engineering in Engineering Studio)	*sqlbrowser.exe*	*1434*

ZENON LOGIC

Application	Standard port
Assigned port for zenon Logic or straton depends on the project and service. e.g.: First zenon Logic project used *1200* and *9000*, second project *1201* and *9001* etc.	*1200 - 1210* *4500 - 4510* *7000 - 7010* *9000 - 9010*

REPORT ENGINE

Application	Standard port
Administration Service	*50777*

Application	Standard port
Analyzer Connector Service	*50778*
Analyzer License Service	*50779*
Reporting Studio	*50781*

DRIVERS

Application	Standard port
Driver Simulation	*6000 - 6020*
Process Gateway OPC Server	*135*
Process Gateway SNMP	*161*
Process Gateway Modbus	*502*
Process Gateway IEC60870-5 104 slave	*2404*
Process Gateway DEC	*5555*
Process Gateway DNP3 Slave	*20000*

IIOT SERVICES

Application	Standard port
IIoT API	*9400*
Certificate Management (general)	*9410*
Certificate Management (dedicated port for connection to **Data Hub**)	*9412*
Data Hub	*9411*
Configuration Backend	*9420*
Identity Service	*9430*
Identity Management	*9431*

Note: zenon drivers that communicate by means of Ethernet use TCP and may therefore require authorizations in the firewall in this case, depending on the port used. Therefore you must ensure that you can share the necessary ports.

6.3 Hardware requirements

One of the requirements for putting Service Engine into operation is that the setup (on page 55) has been completed successfully. A valid license is also necessary. If this license is missing, Service Engine can only be started for a limited time period in demo mode (on page 76).

The minimum requirements are listed in the following table. These are based on a complete installation of Service Engine. For extensive projects or several projects loaded at the same time you will need accordingly faster/stronger hardware. The minimum requirements can increase as a result of this.

Hardware	Minimum requirements	Recommended
CPU	Single core with SSE2 support.	Quad Core
Memory	Windows 8: from 512 MB. **Note:** Projects with large amounts of data, network projects, several parallel projects and projects in redundant operation require more memory.	Windows 8: 4096 MB
Storage medium	2 GB free space for Service Engine installation plus additional space for the projects. **Attention:** If you log historical data (e.g. archived data or alarm/CEL data), there will need to be sufficient memory available or the configuration has to ensure that the historical data is evacuated or deleted.	
Monitor resolution	VGA with 1280 x 800 pixels.	
Graphics adapter	64 MB dedicated memory. Cards with shared memory can lead to performance loss. Note the System requirements when using DirectX (on page 14) node in relation to this.	
Input devices	Keyboard and/or mouse. Operation via touchscreen is also possible. Many individual, customizable touch keyboards for the touchscreen are available for you.	
USB interface **(optional)**	▸ For installation. Installation also possible via network or other storage media. ▸ For dongle. Network dongle also available.	

Hardware	Minimum requirements	Recommended
Network connection (optional)	64 kBits/s for standard Client/Server projects. 100 Mbit/s full duplex for redundant operation.	100 Mbits/s full duplex for standard Client/Server projects.
Remote connection (optional)	Minimum requirements: Dial-up modem with 9600 Bit/s.	1 Mbit/s full duplex.
WAN connection (optional)	Any desired conncetion via router, e.g. per ISDN or DSL. Data transfer is slower in a WAN than in a local network for technical reasons. Be sure to check the possible data transfer rates of your WAN technology already at the time when you create the project.	
Message Control (optional)	Please refer to the Message Control node for the requirements.	
Interfaces (optional)	The necessary interfaces depend on the requirements of the PLC and/or the bus connection, for example serial RS232 or RS422/485 interfaces, ISA/PCI slots, etc.	

6.3.1 Paths for setup and operation

Paths for zenon:

- Setup
- Service Engine

> ### Information
>
> You can display many standard paths with the help of the **set** command:
>
> - start the command line (enter **cmd** in the Windows start area)
> - enter command **set**
> - Press the **Enter key** to display standard folders for Windows and zenon

Note: As absolute paths differ in different operating system, the paths are displayed as Windows environment variables in this node. For example *%ProgramData%* instead of *C:\ProgramData*.

SETUP

During setup, paths are set for required third-party software and the Service Engine folder.
Setup requires administrator rights, which also applies to changing the installation paths. You can read more information about prerequisites in the **Installation and updates** (on page 6) node, in the **Software and paths for installation and operation** (on page 55) node.

SERVICE ENGINE

Service Engine uses the path that is set in zenon6.ini for projects. This path can be set using the Startup Tool as well as using the Remote Transport from a Engineering Studio.

Object	Path
Project	*[user-defined path]\[project]\RT*
External files	*[User-defined path]\[project]\RT\FILES\...*
Exported archives, Chronological Event List and Alarm Message List	*[User-defined path]\[project]\Export* **Note:** Is created at the first export.
System files	Windows system folder.

ZENON LOGIC

Paths for zenon Logic are created in the same way as zenon paths.

6.3.2 Demo licenses and Demo mode

Each installation contains:

- **Demo licenses**: Allows you to test a component for a certain time period.
  Demo licenses follow the rules for licenses with a time quota.
  Note: Engineering Studio can be started the first 30 times with a duration of 30 minutes each. It switches to demo mode after this.
- **Demo mode**: Allows you to start a component and use it for a certain period of time. It is closed again afterwards.
  Engineering Studio duration: 10 minutes
  Service Engine duration: 30 minutes

Each installation contains at least one demo license. This has a pre-defined duration or number of permitted starts. If these are used up, the component can continue to be used in demo mode, but it is automatically ended after running for a defined duration however. You can find out what usage period is available in the **License Manager**. To do this, open the details of the license in the **License Manager** tab.

COPA-DATA **components:**

- Engineering Studio with Logic Studio as a service
  Demo license:
  - 30 Starts, with a duration of 30 minutes each
  - 10 minutes, if the number of demo starts has been used up
- Service Engine und Logic Service:
  Runtime period for demo licenses:
  - 43200 minutes on hardware computers
  - 1440 minutes for virtual machines
  - 30 minutes if the time allocation has been used up
- Logic Studio (as an integrated solution in Engineering Studio)
  Term for demo licenses:
  - 256 TAGs
  - 120 minutes
- **Process Gateway** and OPC DA Server
- **Smart Server** and **Smart Client**
- **HTML Web Engine**
- Report Engine
- Certificate Management
- IIoT API
- Identity Service

Demo licenses have their own software dongles. These cannot be used for other licenses. The last existing demo license cannot be deactivated using the **License Manager**.

The remaining duration and the number of the starts that are still possible for demo licenses and demo mode are shown in the start screen and in the version and license information.

To administer existing demo licenses:

1. Open the **License usage** tab.
2. Select the desired demo license.
3. Select the desired component.
4. Click on the button at the far right of the list.
   The button turns green in color and the license is now being used.

MISSING DEMO LICENSE

If no demo license is available on your computer, then:

1. Open the **License Manager**.

   The activation of a demo license is offered to you on the start screen. You only find demo licenses in this area. Component licenses that have already been obtained are not displayed here.

2. Select the desired demo license.

   The license is activated and can be used immediately by clicking on it. Only one demo license can be activated at the same time. If a new demo license is activated, the previous one is deactivated.

Note: If there is also no demo license available in **License Manager**, contact your COPA-DATA sales partner.

> ### Hint
>
> You can use demo licenses to test components and editions. To do this, place the demo license at the first location of the license list in the **License Manager**. This license is used the next time the component is started.

6.3.3 Service Engine under Windows Embedded Standard

The minimum requirements relate to an installation of Service Engine adapted to the Windows Embedded Standard 7 SP1 operating system with platform update. The hardware must be accordingly more powerful for extensive projects.

This table only states the figures that are different to the standard installation. The other parameters correspond to the figures described in the Hardware requirements for Service Engine (on page 18) node.

Hardware	Minimum requirement	Recommended
Memory	▸ 512 MB. **Note:** Projects with large amounts of data, network projects, several parallel projects and projects in redundant operation require more memory.	▸ 2048 MB
Storage medium	▸ 2 GB of free memory on C:\ drive before the installation of .NET Framework 800 MB of free memory after the installation of the	▸ 80 GB

Hardware	Minimum requirement	Recommended
	.NET Framework - plus memory space for the projects, archives, etc.	

Attention: If you log historical data (e.g. archived data or alarm/CEL data), there will need to be sufficient memory available or the configuration has to ensure that the historical data is evacuated or deleted.

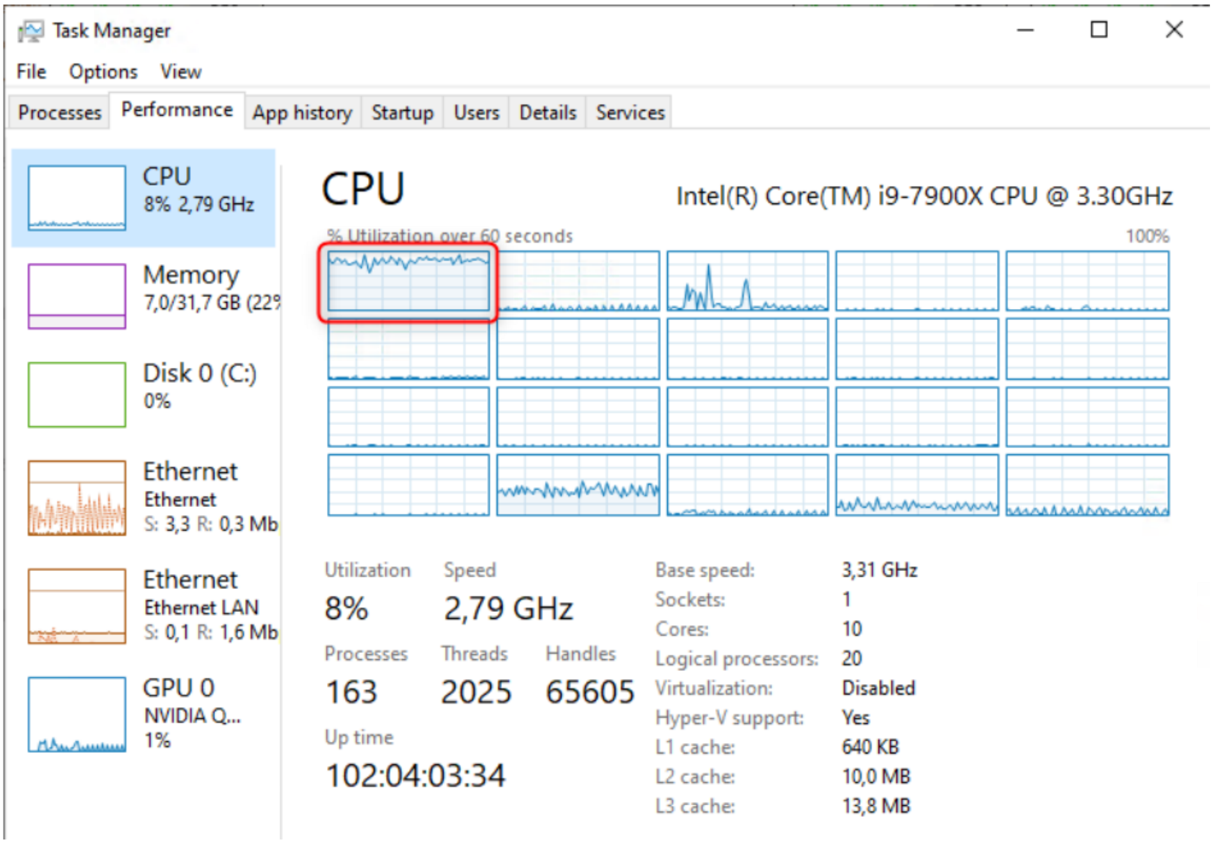
6.4 How do you establish whether the Service Engine computer has reached its limits?

The system is at the limit of its CPU load if the Windows **Task Manager** reports:

- That a logical processor has a lasting load of more than 80% and
- The CPU speed is close to the maximum

This is also the case if the stated overall CPU load is low.

Example: The illustration shows a situation in which a logical processor is constantly under a high load of 80%. However, because the CPU speed has not achieved peak values (2.8 GHz of a nominal 3.30 GHz), this system can be seen as just about powerful enough.



Intel(R) Core(TM) i9-7900X CPU @ 3.30GHz

Utilization 8%
Speed 2,79 GHz
Processes 163
Threads 2025
Handles 65605
Up time 102:04:03:34

Base speed: 3,31 GHz
Sockets: 1
Cores: 10
Logical processors: 20
Virtualization: Disabled
Hyper-V support: Yes
L1 cache: 640 KB
L2 cache: 10,0 MB
L3 cache: 13,8 MB

If the computers continually work with a high load, it is recommended that they are operated in an air-conditioned environment.

Note: In the zenon network, the Standby Server needs the highest CPU capacities, because it must process both the tasks of the server and those of the client. If a CPU load that is too high is detected, the zenon project should be scaled to several projects and distributed to further servers. You can find more information in the **Multi-Project Administration** node.

6.5 Uninstall Service Engine

To uninstall Service Engine:

1. Open the **Control Panel**.
2. Click on **Programs**.
3. Click on **Programs and Features**.
4. Select zenon 14 in the programs.

5. Click on **Uninstall**.
6. Follow the uninstall wizard.

7 Report Engine

This section provides information for installing the Report Engine.

7.1 Installation and updates

The installation of Report Engine consists of several components:

- Report Engine server: Central SQL server.
- Reporting Studio: Application to administer Report Engine and to create reports. It must be installed on the engineering computer.
- Additional applications

> ⚠**Attention**
>
> Ensure that you have the appropriate licenses.
>
> **Note:** For each user, up to three different devices at the same time per license are permitted.

You can find the hardware and software requirements in the Prerequisites node.

NOTES

Note the following before installation:

Theme	Description
User authorizations:	Local administrator rights are required for the installation of Report Engine. Ensure that, after installation, there is at least one user who can log into Report Engine. **Attention:** The user who carries out an installation is automatically created as the first user for Report Engine. After installation, only this user can login to Report Engine and add further users. For example: User 1 is the local administrator and carries out the

Theme	Description
	installation. Report Engine is used by User 2 however. User 1 must add User 2 in the Reporting Studio after installation.
ISOs and restart:	The content of ISO images must be copied to a local storage medium before the installation and setup can be started from here. The inclusion of an ISO image and the installation of Report Engine by the mounted driver cannot be completed successfully if a restart is required during installation.
Licensing:	Licensing is carried out using the **License Manager**. This can be started from Reporting Studio or from the operating system directly.
.NET Framework 4.8:	.NET Framework 4.8 must already be installed and running on the target computer in order to carry out the installation successfully. Otherwise, an error notification from the Report Engine setup will show up and the installation process will be canceled.
Remote installation:	The installation medium must be on the local computer. Network drives may not be available punctually for a reboot during installation. For remote installations and virtual installations, copy the content of the installation medium to a temporary folder on the computer and start the setup.
Server:	The Report Engine server and the **Domain Controller** must not be installed on the same computer.
IIoT Services	Some components need a connection to the IIoT Services in order to connect with the Service Engine. You can configure this with the **IIoT Services Connection Wizard**.
SQL Server Management Studio:	If the SQL Server Management Studio (SSMS) is to be available, it must be installed manually.
Wizards	Several wizards are available for Report Engine. They work with different Report Engine versions and zenon versions. For details, read the **Report Engine wizard compatibility** node.

PERFORMING THE INSTALLATION

To install Report Engine components:

1. Connect the installation medium to the computer or copy its contents to a local folder.
   If Autorun does not automatically start the setup, use the file named **start.exe**.

   The zenon Software Platform setup is opened.
2. Select the desired language from the drop-down list.

3. Accept the license conditions.
4. Click on the **Next** button.
5. Accept the data protection agreement.
6. Click on the **Next** button.
7. Select the desired components.

   Note: For the installation of the Report Engine Server, a **Data Hub** for IIoT Services must also be installed. This is regardless of whether there is already a **Data Hub** in the system. The Report Engine server uses the **Data Hub** to establish the connection to IIoT Services.
   In an additional dialog, you are requested to issue a user name and password for access to the **Data Hub**. **Attention:** Note the password in a secure place. It cannot be displayed or recovered later.
8. Click on the **Next** button.

   The installation or the update will start.
9. Follow the instructions given to you by the installation wizard.
10. After successful installation, configure the connections to IIoT Services.

Notes on update: When switching version, the version of the assemblies contained in the database is checked before the update of the structure. If the version to be installed is more recent, the SQL elements contained are updated.

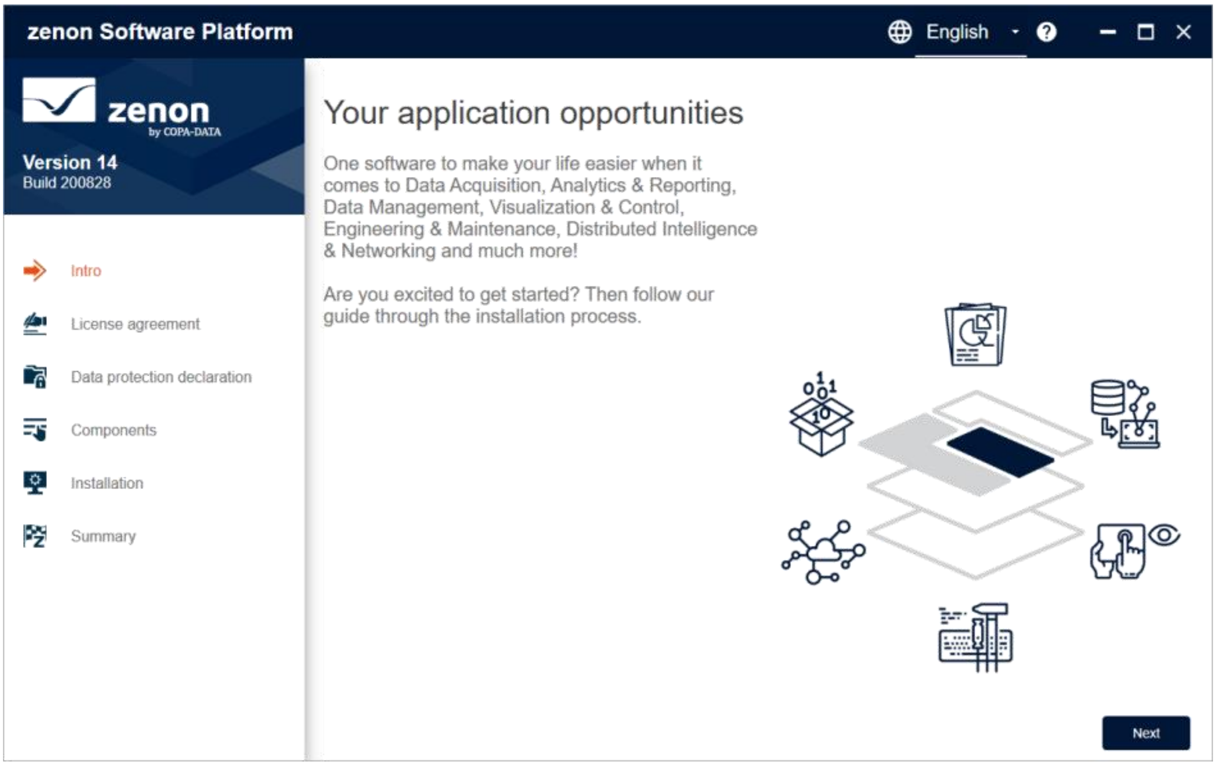
INSTALLATION ON THE CLIENT

Only a current browser is needed on the client. The language that is set in the browser determines the language for Report Launcher. The language for Reporting Studio is stipulated in the options in Reporting Studio.

7.1.1 Start window

You are given general information about the zenon Software Platform in the start window.

The information in the left window shows you the current status of the installation process. You switch to the next respective window with the **Next** button.
You can get help on installation by clicking on the Help symbol at the top right.



Your application opportunities

One software to make your life easier when it comes to Data Acquisition, Analytics & Reporting, Data Management, Visualization & Control, Engineering & Maintenance, Distributed Intelligence & Networking and much more!

Are you excited to get started? Then follow our guide through the installation process.

1. From the drop-down list at the top right, select the desired language for installation.
   The following languages are available:
   - German
   - English
   - Italian
   - French
   - Spanish
   - Czech
   - Japanese
   - Korean
   - Chinese
   - Russian

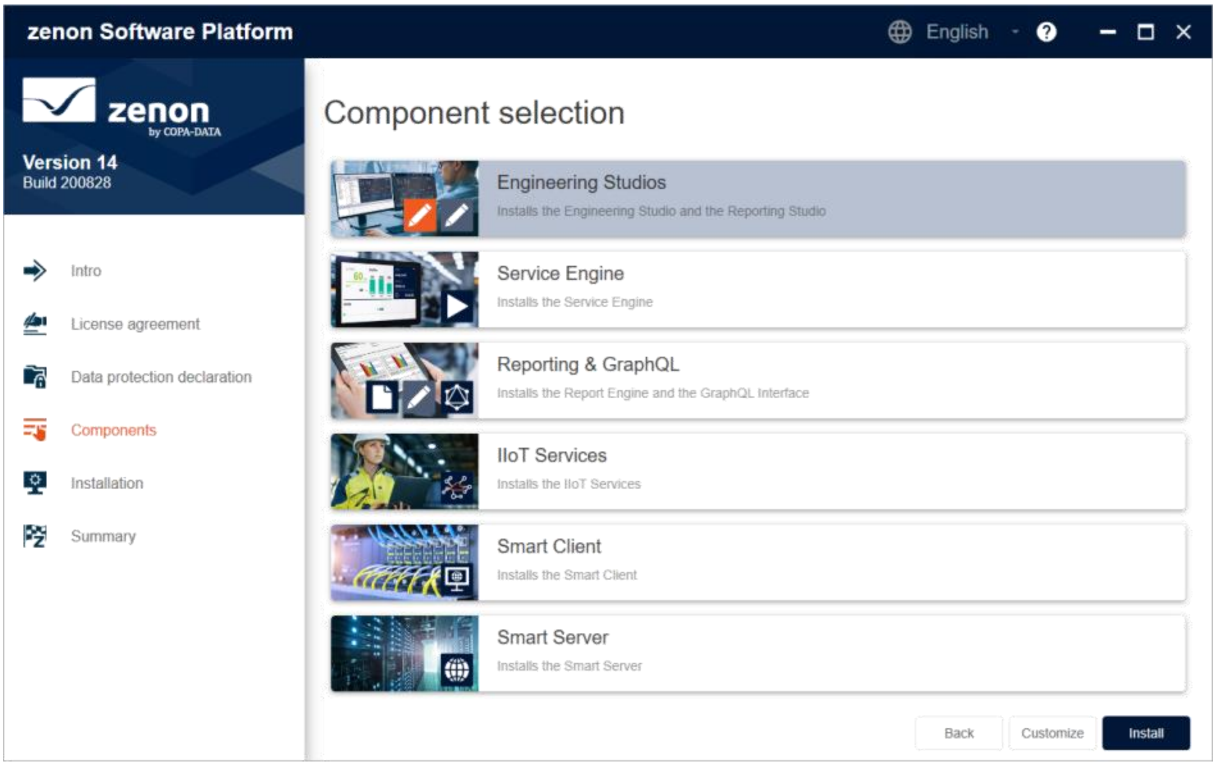
   Note: The language can only be changed on this page. In the following steps, the language is shown but can no longer be amended.
2. Clicking on the **Next** button opens the window with the license conditions.

3. Confirm the license conditions by activating the corresponding checkbox.
   If you do not accept the license conditions, you cannot install the product.

   You can also print the license conditions out by clicking on the **Print** button.
4. Clicking on the **Next** button opens the privacy policy.
   Read the privacy policy carefully.
   You can print out the privacy policy by clicking on the **Print** button.
5. Activate the checkbox for the privacy policy.
   This will confirm that you have read this. If you do not accept the privacy policy, the product cannot be installed.
6. Clicking on the **Next** button opens the window to select the desired product.
   Note: The **Next** button is only available if you have agreed to the license conditions by activating the checkbox.

7.1.2 zenon Standard installation

Select the desired components. It is only possible to select components that have not already been installed. If you want to carry out a reinstallation, you must first uninstall the previously-installed components using the Windows **Apps** application or the Control Panel.



Component selection

Engineering Studios
Installs the Engineering Studio and the Reporting Studio

Service Engine
Installs the Service Engine

Reporting & GraphQL
Installs the Report Engine and the GraphQL Interface

IIoT Services
Installs the IIoT Services

Smart Client
Installs the Smart Client

Smart Server
Installs the Smart Server

INSTALLATION PACKAGES

There are 6 collections of packages available for installation. By default, only the pre-selected components are installed, as well as components that you choose to be installed. You can individually add or remove further components before installation by using the **Customize** button.

Note: If an exclamation mark is shown for a package, this indicates incompatibility for the installation. For example, already-installed components that do not permit the installation of a further version. In this case, select the adjusted installation.

Package	Components
Engineering Studios	Installs Engineering Studio a Reporting Studio. **Preselected:** ▸ Engineering Studio ▸ Reporting Studio ▸ Device Management Interface components ▸ License Manager **Optional:** ▸ Service Engine (pre-selected) ▸ Report Engine ▸ GraphQL Interface ▸ Smart Client ▸ Smart Server ▸ IIoT Services
Service Engine	Installs Service Engine. **Preselected:** ▸ Service Engine ▸ Device Management Interface components ▸ License Manager **Optional:** ▸ Engineering Studio ▸ Report Engine ▸ GraphQL Interface

Package	Components
	- Reporting Studio - Smart Client - Smart Server - IIoT Services
Reporting & GraphQL	Installs Report Engine and the **GraphQL Interface**. **Preselected:** - Report Engine - IIoT Services - License Manager **Optional:** - Engineering Studio - Service Engine - GraphQL Interface - Reporting Studio - Smart Client - Smart Server - Device Management Interface components **Attention:** The Report Engine 14 cannot be installed in parallel with a Report Engine version 10, 11 or 12 on the same computer. If one of these versions already exists on the system, a corresponding message is displayed when you select it in the setup. The installation of Report Engine 14 is only possible after the uninstallation of Report Engine 10, 11 or 12. Versions of **Analyzer 3.40** or older can be installed in parallel with Report Engine 14.
IIoT Services	Installs IIoT Services . **Preselected:** - IIoT Services - License Manager **Optional:** - Engineering Studio

Package	Components
	‣ Service Engine ‣ Report Engine ‣ GraphQL Interface ‣ Reporting Studio ‣ Smart Client ‣ Smart Server ‣ Device Management Interface components
Smart Client	Installs Smart Client. **Preselected:** ‣ **Smart Client** ‣ License Manager **Optional:** ‣ Engineering Studio ‣ Service Engine ‣ Report Engine ‣ GraphQL Interface ‣ Reporting Studio ‣ Smart Server ‣ IIoT Services ‣ Device Management Interface components
Smart Server	Installs Smart Server. **Preselected:** ‣ Smart Server ‣ License Manager **Optional:** ‣ Engineering Studio ‣ Service Engine ‣ Report Engine ‣ GraphQL Interface

Package	Components
	- Reporting Studio - Smart Client - IIoT Services - Device Management Interface components

7.1.3 Selection and installation

Report Engine is automatically installed durig a complete installation of the components for **Reporting & GraphQL**.

To install the Report Engine only:

1. Click on **Customize** in **Reporting & GraphQL**.

   The dialog to customize the installation is opened.
   The standard packages have already been pre-selected.

2. Deselect Service Engine
3. Select Report Engine.
   All other required packages are selected automatically.

To install the Reporting Studio only:

1. Deselect all default entries.
2. Activate the Reporting Studio , Metadata Editor and License Management.

Notes:

1. Packages can only be deselected if they are not required by any other package.
   Already-installed packages cannot be deselected.
2. If necessary, configure the paths for installation in the **Options** tab.
3. The computer may be restarted automatically during installation.

7.1.4 Updates

In zenon, you can change to new versions for example, from 10 to 11. Or you can install updates within a version.

- New versions can be installed in parallel with existing versions. They mainly offer new features.
- Build updates modify a previously installed version. They mainly fix bugs.

With each setup for new build or new versions, you receive a link to the COPA-DATA website with the changes between the previous version and the newly-installed one.

BUILD UPDATE

An update only updates those files which are more current than the previously installed files. All projects and individual settings will remain unchanged. Build updates can contain changes for all installed components. When calling up the setup, the components that have been changed and the version to which they have been changed are shown.

Note: Build updates may have lower quality assurance standards than release versions.

Note: Build updates can only be installed locally. Installation on network paths (UNC) is not possible.

7.1.5 Upgrade information

Note: An upgrade is a higher-level configuration or version. For example, if **Analyzer 3.40** and Report Engine 10 are installed side-by-side, this is not an upgrade.

UPGRADE WITHOUT SQL SERVER SWAP

The upgrade can be done through the setup.

Note: Create a backup before upgrading.

UPGRADE WITH SQL SERVER SWAP

To perform an upgrade:

1. Create a backup of the Report Engine version you are using.
2. Reinstall the backup file on a newly installed system with Report Engine 10 or higher.
3. If you want more than just an upgrade (**Example:** linked servers, users, permissions,...), the **Migration Tool** can be used.

7.1.5.1 Database structure

The database structure has been changed to version 12.

8 IIoT Services

In this section you will find information for installing IIoT Services and for the initial setup on Windows or Docker.

8.1 Installation

It is recommended to always install the latest IIoT Services release. Existing installations can be upgraded within the recommended update paths (on page 97).

INSTALLATION VARIANTS OF THE IIOT SERVICES

IIoT Services offer the same range of functions in all installation options.

Please note the following differences:

	IIoT Services (Docker on Windows)	IIoT Services (Docker on Linux)	IIoT Services (Windows native)
Application area	▸ Test environments	▸ Test environments ▸ Productive environments	▸ Test environments ▸ Productive environments
Host operating system	Windows	Linux	Windows
Method of installation	Configuration files	Configuration files	As native Windows application via a .ISO file The installation of the IIoT Services is integrated in the **Setup of the software platform** (on page 24).
Internal services	Docker services as a Linux container	Docker services as a Linux container	Windows Services
Monitoring of internal	▸ **Windows**	▸ With Shell	**Windows Management Console**.

	IIoT Services (Docker on Windows)	IIoT Services (Docker on Linux)	IIoT Services (Windows native)
services	**PowerShell** - With GUI via **Docker Dashboard**	No GUI available.	
Minimum number of computers for test environment	- 1 computer for the IIoT Services and all clients (Windows host OS)	- 1 computer for the IIoT Services (Linux host OS) - 1 computer for clients (Windows host OS)	1 computer for the IIoT Services and all clients (Windows host OS)
Minimum number of computers for produtive environment	- 1 dedicated computer for the IIoT Services. - Separate computers for clients.	- 1 dedicated computer for the IIoT Services. - Separate computers for clients.	- 1 dedicated computer for the IIoT Services. - Separate computers for clients.

Note: The installation options of IIoT Services are basically the same for the administration in the Service Configuration Studio.

⚠Attention: Fixed user context in "Docker on Windows"

A IIoT Services installation in **Docker Desktop for Windows** **Docker on Windows** is started in a **fixed user context**.

Example: User A installs IIoT Services using Docker Desktop for Windows.

- **User A** has access to IIoT Services through their user account.
- **User B** does not have access to IIoT Services through their user account on the same computer.

Hint: You can get around this limitation on a test system by using a shared user account.

8.1.1 Installation: Standalone vs. parallel vs. virtual machine

In general, it is recommended to install IIoT Services as standalone applications on a dedicated computer.

STANDALONE INSTALLATION

With a standalone installation, other than IIoT Services, no further zenon services are installed.

Standalone installation is recommended for:

- All installation options of IIoT Services (Docker and Windows native)
- All computer types (physical computer and VMs)
- All uses (test systems and productive systems)

Standalone installation ensures a clear separation of connected communication partners in IIoT Services networks.

PARALLEL INSTALLATION

In a parallel installation, both IIoT Services as well as other zenon services are installed on the same computer.

Parallel installation is only recommended for separately documented cases. An example of this is the test environment in the Getting Started Guide for the IIoT Services (Docker on Windows) installation option.

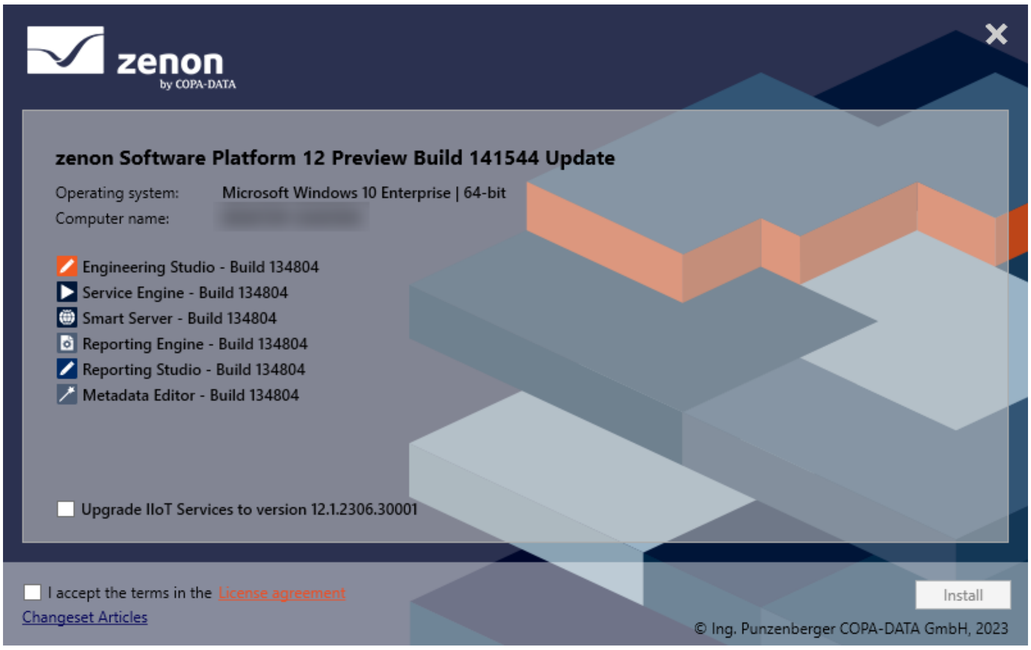
INSTALLATION ON A VIRTUAL MACHINE

⚠**Attention**

If you run **IIoT Services** on a virtual machine with Docker containers: First check whether AVX commands are supported by the underlying hypervisor.

8.1.1.1 Build update IIoT Services

As of version 12, IIoT Services versions can also be updated when installing a build update.
In this case, an option for updating the IIoT Services version to the current version is offered when the setup is called.



zenon Software Platform 12 Preview Build 141544 Update

Operating system: Microsoft Windows 10 Enterprise | 64-bit
Computer name: [redacted]

- Engineering Studio - Build 134804
- Service Engine - Build 134804
- Smart Server - Build 134804
- Reporting Engine - Build 134804
- Reporting Studio - Build 134804
- Metadata Editor - Build 134804

□ Upgrade IIoT Services to version 12.1.2306.30001

□ I accept the terms in the License agreement

Changeset Articles

© Ing. Punzenberger COPA-DATA GmbH, 2023

Procedure:

- Checkbox for upgrading IIoT Services is deactivated:
  IIoT Services are not updated. They remain in the version already installed on the system.
- Checkbox for upgrading IIoT Services is activated:
  - IIoT Services are updated to the current version.
  - The task performs a silent installation of the IIoT Services setup without rebooting.
  - If the installation is successful, a button for manual reboot is offered at the end.
    Do a reboot before starting IIoT Services. This ensures the correct operation of all services.

8.1.2 Kubernetes

In the Docker installation option, the IIoT Services services are installed in Linux containers. This meets the requirements for operating IIoT Services in a Kubernetes cluster.

In particular, you should note:

- The configuration files *docker-compose.yml* and *.env* provided with IIoT Services can be used as a foundation for creating Kubernetes configuration files.
- You must create customized Kubernetes configuration files for your specific environment.

The following application scenarios are possible:

- self-hosted Kubernetes cluster (*on-premise*)
- third-party hosted Kubernetes cluster of cloud providers such as Amazon *(Amazon Kubernetes Cluster)*, Google (*Google Kubernetes Engine*) or Microsoft (*Azure Kubernetes Service*)

By using the Kubernetes container management, you can provide, scale and manage containers automatically.

> 👍 **Hint**
>
> The use of IIoT Services in a Kubernetes cluster requires relevant prior knowledge and is generally recommended only for enterprise environments.

8.1.3 Update paths

The following update paths are recommended for IIoT Services:

- Version 2.0 to 2.1
- Version 2.x to 10.0
- Version 10.0 to 10.x
- 10.x to higher versions

Available configurations will be automatically carried over during the update. It is recommended to perform a backup of **Persistence Instance** before every version update.

8.1.4 Compatibility

The IIoT Services typically connect zenon applications and third-party applications to one another. It must be ensured that all connected applications are compatible with one another.

The following applies for a IIoT Services installation:

- Install the most recent version of the IIoT Services.
- The IIoT Services version must be at least just as high as the version of the connected zenon applications.

- All connected zenon applications must be compatible with the installed IIoT Services version.

> **Information**
>
> Backward compatibility:
>
> The IIoT Services support components of the zenon software platform from version 10 and higher.

8.1.4.1 Renaming of Service Grid to IIoT Services

With version 12, the **Service Grid** and its Services and components were renamed to **IIoT Services**. The graphical user interface of the **IIoT Services** and apps (programs) and the help have been adjusted accordingly.

Previous versions up to and including 11.2	From version 12
Service Grid	**IIoT Services**
Service Grid Hub Controller	**Certificate Management**
Service Grid API	**IIoT API**
Service Grid Persistence	**Persistence Service**
Service Grid Studio	**Service Configuration Studio**
Service Grid Proxy	**Proxy Service**
Service Grid Gateway	**IIoT Services Gateway**
Service Grid Egress Connector	**Data Hub Driver**
Service Grid Ingress Connector	**IIoT Services Gateway** components.

8.1.4.2 Compatibility of version 12 with previous versions

With version 12, the connection between Engineering Studio project configurations to IIoT Services and communication between the individual services was simplified.

For existing project configurations in Engineering Studio, it is expressly recommended that the connections to the IIoT Services and the configuration for individual services are reconfigured.

You can find detailed information on the configuration in the **IIoT Services - configuration in Engineering Studio** section.

8.1.4.3 Compatibility of version 11.2 with previous versions

From version 11.2 and higher, the individual services of IIoT Services are addressed using a central URL. The addressing of services was by means of port numbers in previous versions. The default port number for IIoT Services is *9443*. This port number can be adjusted by means of configuration.

If the central IIoT Services URL is entered in a web browser, Service Configuration Studio starts.

COMPATIBILITY NOTE - IIOT SERVICES 11.2

The following is applicable in order to work seamlessly with a zenon installation with IIoT Services:

- Service Engine or Engineering Studio in version 11 communicate with IIoT Services version 11.2 or higher:
  Build *111398* or higher for Service Engine or Engineering Studio must be installed.
  The following configurations must also be updated:
  - **Service Node Configuration Tool**
    When entering a **Connection** setting, add the */hub-controller* sub-path to the central URL.
    Example: *https://hostname.local/hub-controller*
    Attention: From version 12, the **Service Node Configuration Tool** has been replaced with the **IIoT Services Connection Wizard**.
  - Configuration in Engineering Studio
    It is not necessary to specify a subpath for configurations in Engineering Studio. The same URL is used for **Identity Service** and **Data Storage**. Entering the central URL is sufficient. The URL of the corresponding properties must be configured as URL + port.
    Examples:
    URL for **Identity Service** (**Network** property group, **Identity Service**, property: **URL**)
    URL for **Data Storage** (**Network** property group, **Data Storage**, property: **URL**)
    hostname.local:9443

8.1.4.4 IIoT Services update

You can migrate a IIoT Services installation to a higher version with little effort.

How to perform an update:

1. Back up the existing **Persistence Instance**.
2. Stop the IIoT Services.
3. Reinstall the IIoT Services in accordance with the installation option.

- IIoT Services (Docker): The *.env* file for the new version must be configured with the necessary values. Afterwards, the IIoT Services are initialized with the *docker-compose.yml*.
- IIoT Services (Windows-native): Carry out the setup of the new version.

4. Restart the IIoT Services.

You have now updated the IIoT Services.

> 👍 **Hint**
>
> Existing certificate bundles of an old IIoT Services version are usually compatible with the new version and do not need to be issued again.
>
> The **Persistence Instance** with all configurations of the IIoT Services is migrated automatically.
>
> The backup of the **Persistence Instance** is a security precaution. This means that a restore is possible in the event of an error.

8.1.4.5 Update MongoDB

From version 12, the update of MongoDB has been made much simpler for the user. MongoDB persistence instances can thus be updated to the most recent version that is used by the IIoT Services. Subsequent change is not possible.

Please also note the **Backup and restore – persistence instance** node in order to back up your data before an update.

UPDATE FOR WINDOWS (NATIVE)

The updating of MongoDB for Windows operating systems is fully integrated into the setup. No additional manual steps are necessary.

UPDATE FOR DOCKER

The following requirements are necessary for updating MongoDB in a Docker environment:

- The tool **CopaData.ServiceGrid.Tools.PersistenceManagementCli.exe** is installed on the computer running Docker Desktop for Windows.
- The installation is done by running *PersistenceManagementCli.x64.msi*.
  The data are stored in the following folder:
  %programfiles%\zenon\zenon Platform 14\IIoT Services\PersistenceManagementCli.
- The **MongoDB Command Line Database Tools** are installed.

- The PATH environment variable has been extended with the path to the MongoDB Command Line database tools (see previous step), e.g.:
  C:\Tools\mongodb-database-tools-windows-x86_64-100.7.0\bin
- The current version of the IIoT Services is installed and running.
- The .ENV file with the current settings and the docker-compose .YML file for the new version are available in their own Windows folder.
- Port *27017* is available on the computer for connecting to the MongoDB database.

RUN UPDATE

In the Docker environment, do the following:

1. Open an elevated PowerShell.
2. Navigate to the storage location of the CLI, e.g. (default path): *%programfiles%\zenon\zenon Platform 14\IIoT Services\PersistenceManagementCli*.
3. Enter the following command:
   CopaData.ServiceGrid.Tools.PersistenceManagementCli.exe docker upgrade

- The tool starts and guides you through the update process step by step. Necessary parameters are queried. The update process is continued after the necessary parameters are entered. In addition, information and a log are displayed directly in the tool.

8.1.4.6 Login to the Identity Service after an update or upgrade

After an update or upgrade of the IIoT Services version, it may happen that logging in to the Identity Service in the web browser is no longer possible.

In this case, clear the cache of your web browser. In addition, you can call up the login screen in a private window of the web browser.

8.1.4.7 IIoT Services Gateway

The **IIoT Services Gateway** connects IIoT Services to zenon applications. It also ensures compatibility between different release versions.

> **Information**
>
> Recommendation: Generally speaking, you should always use the version of **IIoT Services Gateway** that corresponds to the installed version of IIoT Services.

VERSION CHECK

Different versions of IIoT Services and **IIoT Services Gateway** can communicate with one another on the basis of a common protocol.

For communication, a check is carried out to see which version of the protocol is used by the individual components. The check is successful if all components use the same major version of the protocol. Minor versions can be different. If the major version is different for a component, communication is no longer possible.

Example:

- Components with *20.00* can communicate with *20.10*.
- Components with *20.10* cannot communicate with *30.10*

> **Information**
>
> Valid major version for IIoT Services 14: *20*

Note: For versions of the zenon Software Platform prior to version 14, all components must correspond exactly. This also includes the minor version.

8.1.4.7.1 Installation

For the connection between zenon applications and IIoT Services, the appropriate version of the **IIoT Services Gateway** must be selected and installed.

The following applications use the **IIoT Services Gateway**:

- Service Engine
- Engineering Studio
- Report Engine
- Reporting Studio

Important: You must always execute both installers (x86 and x64) on each client. This way, you ensure that these clients can connect to the IIoT Services.

Tip

Check the installation:

Under **Apps and features**, the Windows operating system shows a separate entry for each installed version of the **IIoT Services Gateway**.

8.1.4.7.2Configuration

PRIOR CONSIDERATIONS

Several versions of the **IIoT Services Gateway** can be installed on a computer at the same time. The system cannot use these versions at the same time however. Only one version of the **IIoT Services Gateway** can ever be centrally configured and used.

These processes can install a **IIoT Services Gateway**:

- Installation of zenon applications via the platform setup.
- Build update of installed zenon applications.
- Installation of the **IIoT Services Gateway** via two separate installers (x64 and x86).

In practice, several versions of the **IIoT Services Gateway** are typically installed on a computer at the same time.

Hint

Use several versions alternately:

You can use several versions of the **IIoT Services Gateway** alternately on one computer. For each change, you must configure the respective required version of the **IIoT Services Gateway** manually in *zenon6.ini*.

DEFAULT CONFIGURATION

By default, zenon applications always connect to IIoT Services via the most-recently-installed version of the **IIoT Services Gateway**.

The default configuration covers the usual application purposes and therefore does not generally need to be adjusted manually by the user.

MANUAL CONFIGURATION

Manual configuration of the **IIoT Services Gateway** by the user is only required in a few cases.

You can use manual configuration to stipulate to the system which version of the **IIoT Services Gateway** zenon applications connect to the IIoT Services.

General requirement:

Several versions of the **IIoT Services Gateway** are installed on the computer.

Manual configuration can, for example, be necessary in the following cases:

- Subsequent downgrade of a zenon installation
- Parallel installations of different zenon versions on one computer
- Connection from zenon applications to different versions of the IIoT Services
- The zenon version used does not support the latest version of the **IIoT Services Gateway**.
- The connection should be established with a current **IIoT Services Gateway** to the IIoT Services of an older version.

> **Hint**
>
> Restart applications and services:
>
> You must restart the following components after manual configuration of the **IIoT Services Gateway**:
>
> - All zenon apps connected to the IIoT Services: Service Engine, Engineering Studio and Reporting Studio
> - The Windows service for the Report Engine service node (if you are using Report Engine): *zanMQTTClientxxxx*
>
> The new configuration is only effective after restarting these applications and services.

8.1.4.7.3 Configuration in zenon6.ini

The **IIoT Services Gateway** is configured centrally in the *%cd_system%\zenon6.ini* file. This setting is applicable for all zenon applications installed on the computer.

The default configuration is as follows:

[ServiceGridGateway]

Version=LAST

In this configuration, zenon applications connect to the most recent version of the **IIoT Services Gateway** that is installed on the computer.

Example of configuration for version 11.0:

[ServiceGridGateway]

Version=11_0

In this example, zenon applications connect to **IIoT Services Gateway 11**.

SYNTAX

The syntax for manual configuration of the version is "MM_N". The first two figures "MM" define the version number of the major release. The last figure "N" defines the minor release.

> 👍 **Hint**
>
> Configure the figure for the minor release:
>
> The last figure must always be given, including for major releases, such as **IIoT Services Gateway** 11 for example. In this case, you must configure the value "*11_0*".

8.2 Getting Started

In this node, you can find detailed information on putting IIoT Services into operation in Windows and Docker environments.

8.2.1 Welcome to COPA-DATA help

GENERAL HELP

If you cannot find any information you require in this help chapter or can think of anything that you would like added, please send an email to documentation@copadata.com.

LICENSES AND SERVICES

If you find that you need other zenon services or licenses, our staff will be happy to help you. Email sales@copadata.com.

PROJECT SUPPORT

You can receive support for any real project you may have from our customer service team, which you can contact via email at support@copadata.com.

8.2.2 Getting Started Guide (Windows)

In this node, you find out how you install IIoT Services on a Windows computer and initially configure it.

> **Information**
>
> These instructions were written on an operating system in English.

8.2.2.1 System requirements

Note the following system requirements to install IIoT Services:

- General

  You can find information on installation via Setup in the **Installation and update** (on page 6) section in the **Installation** (on page 92) chapter.

- Operating system

  You can find information on supported operating systems in the**Installation and update** (on page 6) section in the **Windows operating systems (2/2)** (on page 11) node.

  Update your Windows operating system to the latest version.

- Browser

  The following browsers are supported:

  - Google Chrome
  - Mozilla Firefox
  - Microsoft Edge
  - Apple Safari

  Note: Always use the most recent version of the respective browser.

- Storage space

  For the installation of IIoT Services, at least 6 GB of free storage space is required on the storage medium.

  You can find the required storage space for further zenon components in the **Installation and update** (on page 6) section in the **Engineering Studio** (on page 17) node.

- Requirements

  Ensure that the following points have been met:

  - Sufficient resources for the smooth operation of all installed applications (CPU, RAM, storage space).
  - Working internet connection.
  - There must be Windows administrator rights.

8.2.2.2 Further requirements

To check your Windows installation, the following requirements must be met:

- An installation of Service Engine and Engineering Studio.
- Make sure that the version of **IIoT Services Gateway** is installed that corresponds to the version of IIoT Services to be installed.
- Ensure that this installation is licensed accordingly.

> **Info**
>
> This installation can be present on your own computer or on the same computer as the Windows installation.

8.2.2.3 Configure IIoT Services

In this node, you can find further information on the following topics:

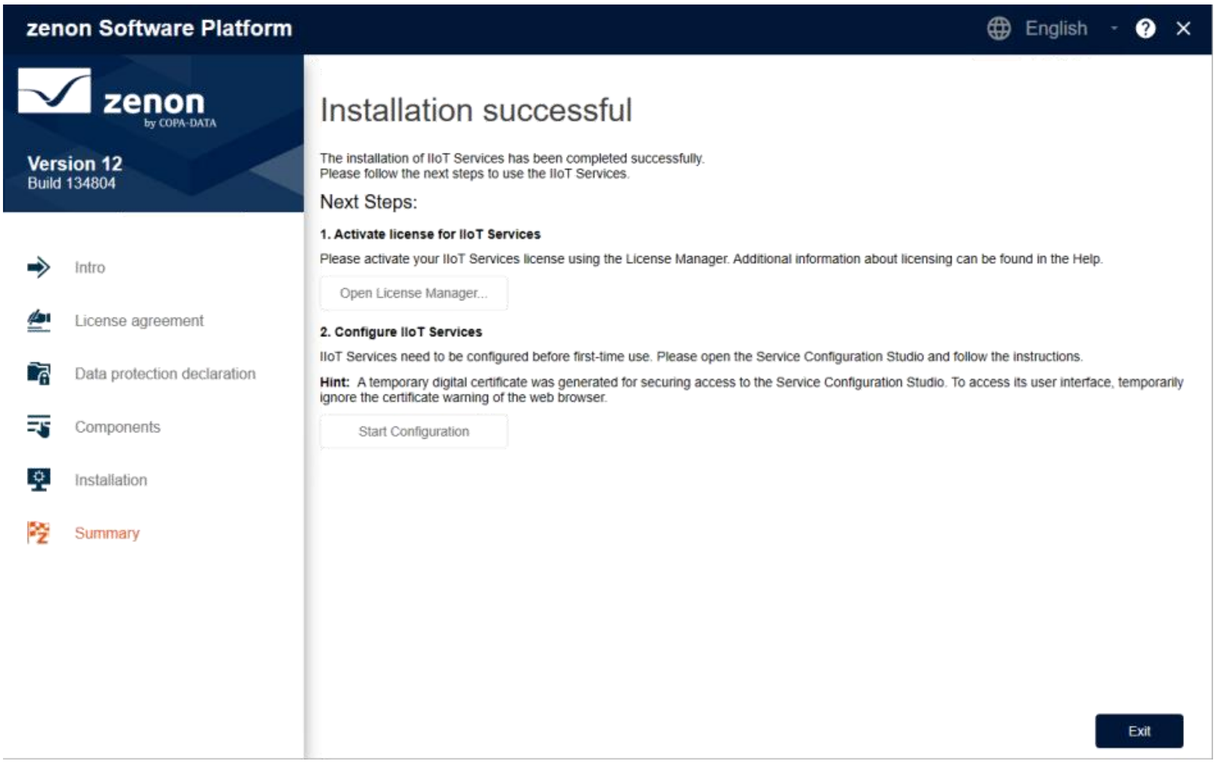
1. Install zenon (on page 109)
2. Activate the licenses for IIoT Services (on page 110)
3. Configure IIoT Services (on page 111)
4. Configure HTTPS trust setting (on page 112)
5. Summary and next steps (on page 114)

8.2.2.3.1Install zenon

Carry out the following steps to install zenon:

1. Double-click on the ISO file.
2. Double-click, in the mounted drive, on the file named START.exe. Setup will start.
3. Carry out the further installation steps. You can find further information in the **zenon Softwareplattform Standardinstallation** (on page 24) node in the **Konfiguration und Installation** (on page 29) node.

   Note: You must select, for the **Components** step, **IIoT Services.** The **Lizenzmanagement** component is already preselected and cannot be deselected. All other components are optional.
4. Ensure, after the necessary restarting of the computer, that you can activate (mount) the ISO file again. Otherwise the installation will not be carried out correctly.



Installation successful

The installation of IIoT Services has been completed successfully.
Please follow the next steps to use the IIoT Services.

Next Steps:

1. Activate license for IIoT Services

Please activate your IIoT Services license using the License Manager. Additional information about licensing can be found in the Help.

2. Configure IIoT Services

IIoT Services need to be configured before first-time use. Please open the Service Configuration Studio and follow the instructions.

Hint: A temporary digital certificate was generated for securing access to the Service Configuration Studio. To access its user interface, temporarily ignore the certificate warning of the web browser.

Note: Do not close the **Installation successful** window. You still need it to activate the licenses and to configure IIoT Services.

> ⚠**Attention**
>
> If you have nevertheless unintentionally closed the **Installation successful** window, you can continue configuration via the **Service Configuration Studio** home page and **zenon Lizenzmanagement**.

Name	Sample values	Description
Service Configuration Studio	*https://mycomputer.mydomain.com:9443* System-specific value*	You can continue the configuration of IIoT Services in **Service Configuration Studio**.

* Replace mycomputer.mydomain.com in the URLs with your computer's FQDN (on page 126).

8.2.2.3.2 Activate license for IIoT Services

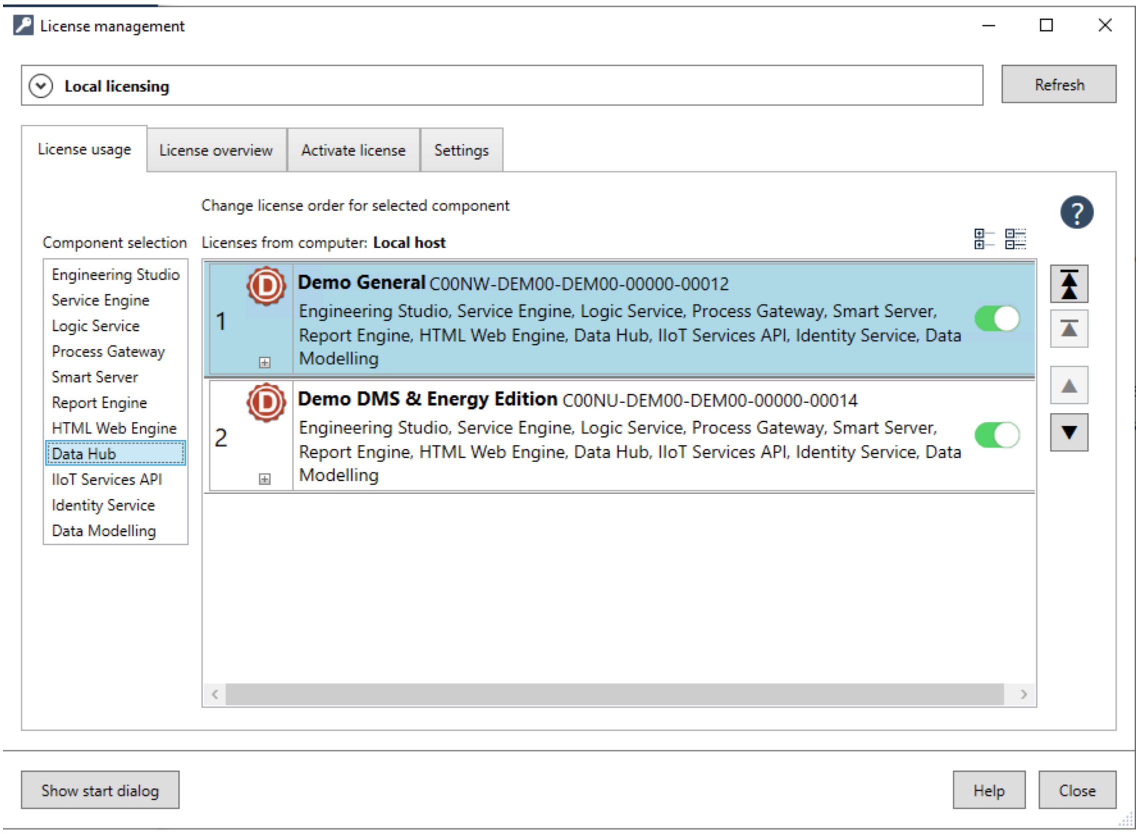
You have several options for licensing. Use either

- the supplied demo licenses or
- your own **zenon** licenses

To install the demo licenses:

1. Click, in the **Installation successful** window, in the **1. Activate license for IIoT Services** section, on the **Open License Manager** button.
2. In the **License management** window that is opened, click on **Advanced options.**
3. In the **License management** window that is opened, click on **Advanced license administration.**
4. Activate the necessary licenses for **Data Hub**, **IIoT Services** API, **Identity Service** and **Data Modeling**. To do this, move the slider to the right. The green background shows that the license has been activated.
5. Use the cursor buttons to move the activated licenses to the top.

6. Close the dialog by clicking on the **Close** button.



Change license order for selected component

Component selection

Engineering Studio
Service Engine
Logic Service
Process Gateway
Smart Server
Report Engine
HTML Web Engine
Data Hub
IIoT Services API
Identity Service
Data Modelling

Licenses from computer: **Local host**

1 **Demo General** C00NW-DEM00-DEM00-00000-00012
Engineering Studio, Service Engine, Logic Service, Process Gateway, Smart Server, Report Engine, HTML Web Engine, Data Hub, IIoT Services API, Identity Service, Data Modelling

2 **Demo DMS & Energy Edition** C00NU-DEM00-DEM00-00000-00014
Engineering Studio, Service Engine, Logic Service, Process Gateway, Smart Server, Report Engine, HTML Web Engine, Data Hub, IIoT Services API, Identity Service, Data Modelling

You can also use your own, pre-existing **zenon** licenses. You can find information about licensing in the Licensing node in the Licensing in a few steps node.

> **Attention**
>
> All **zenon** components that you use in **IIoT Services** must also be licensed.
>
> You can find information in relation to this in the Licensing node in the Licensing components - overview node.

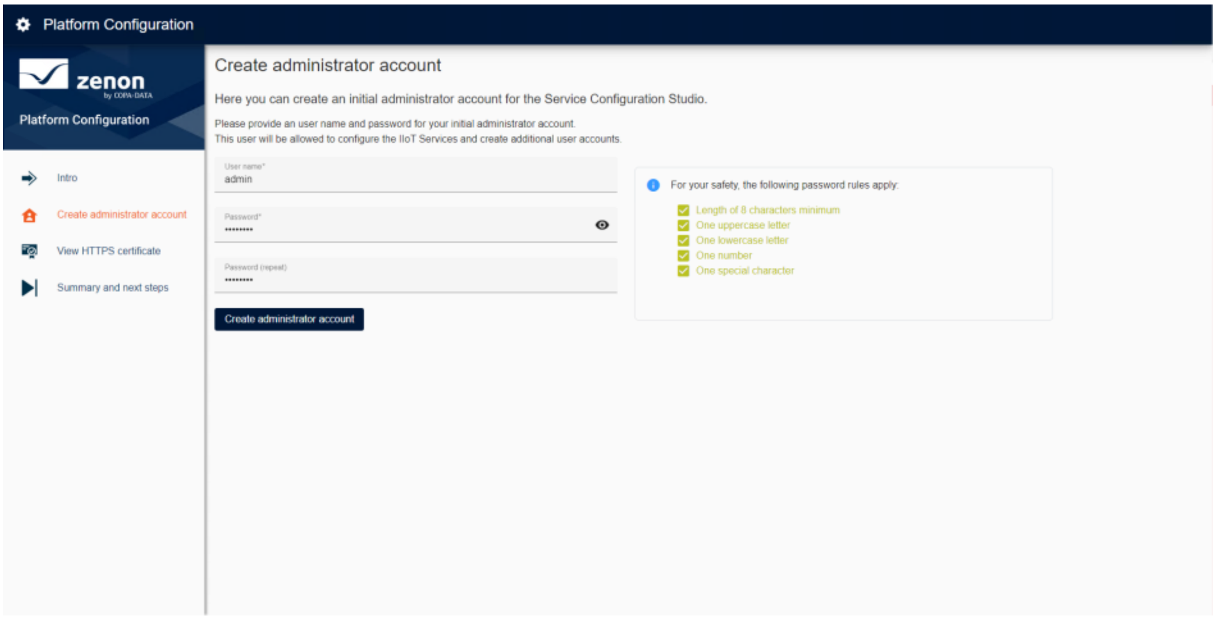
8.2.2.3.3 Configure IIoT Services

You configure the administrator account with the following steps.

1. Click, in the **Installation successful** window, in the **2. Configure IIoT Services** section, on the **Start Configuration** button.

   Note: In **Service Configuration Studio**, you can continue the configuration of the platform at a later point in time by clicking on the **Platform Configuration** button. The configuration is continued from the point where it was stopped.

2. In the **Platform Configuration** window, click on the **Get started** button**.**
3. Enter a user name in the **Create administrator account** window.
4. Enter a password. Note the given password criteria. If the password criteria are adhered to, the font color changes to green.
5. Enter the password again. If the two entries of the password match, the **Create administrator account** button is activated.
6. Click on this button. The creation of the administrator account is thus completed.



Create administrator account

Here you can create an initial administrator account for the Service Configuration Studio.

Please provide an user name and password for your initial administrator account.
This user will be allowed to configure the IIoT Services and create additional user accounts.

User name*
admin

Password*
••••••••

Password (repeat)
••••••••

For your safety, the following password rules apply:

☑ Length of 8 characters minimum
☑ One uppercase letter
☑ One lowercase letter
☑ One number
☑ One special character

Note: This user is also authorized to configure IIoT Services and to create further users

> ### ⚠Attention
>
> Note the password in a secure place. If the password is forgotten, there is no possibility to retrieve it.

8.2.2.3.4 View HTTPS certificate

IIoT Services use an HTTPS certificate for secure communication. To trust the HTTPS certificate, the root certificate must be trusted.

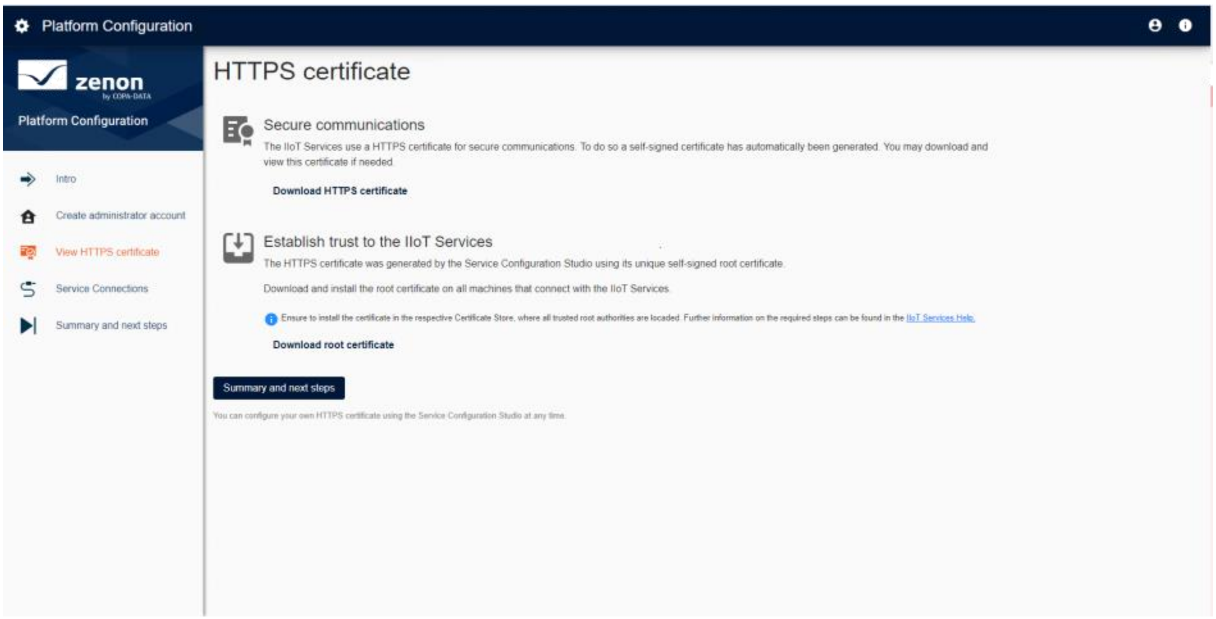
To install the root certificate, proceed as follows:

1. Click, in the **HTTPS certificate** window, on the **Download root certificate** button.

2. Open the downloaded certificate and install it in the **Trusted Root Certification Authorities Store**. You can find information on the procedure in the **HTTPS-Vertrauensstellung** node in the **Vertrauensstellung konfigurieren** node.
3. After successful installation of the root certificate, click on the **Summary and next steps** button.

> ## ⚠Attention
>
> When installing IIoT Services for the first time, your browser will issue a security warning. At this stage, you cannot verify the certificate yet. To complete the installation, you must ignore the security warnings this one time.

Note: Also install the root certificate on all clients that you want to connect to IIoT Services.



HTTPS certificate

Secure communications

The IIoT Services use a HTTPS certificate for secure communications. To do so a self-signed certificate has automatically been generated. You may download and view this certificate if needed.

Download HTTPS certificate

Establish trust to the IIoT Services

The HTTPS certificate was generated by the Service Configuration Studio using its unique self-signed root certificate.

Download and install the root certificate on all machines that connect with the IIoT Services.

Ensure to install the certificate in the respective Certificate Store, where all trusted root authorities are located. Further information on the required steps can be found in the IIoT Services Help.

Download root certificate

Summary and next steps

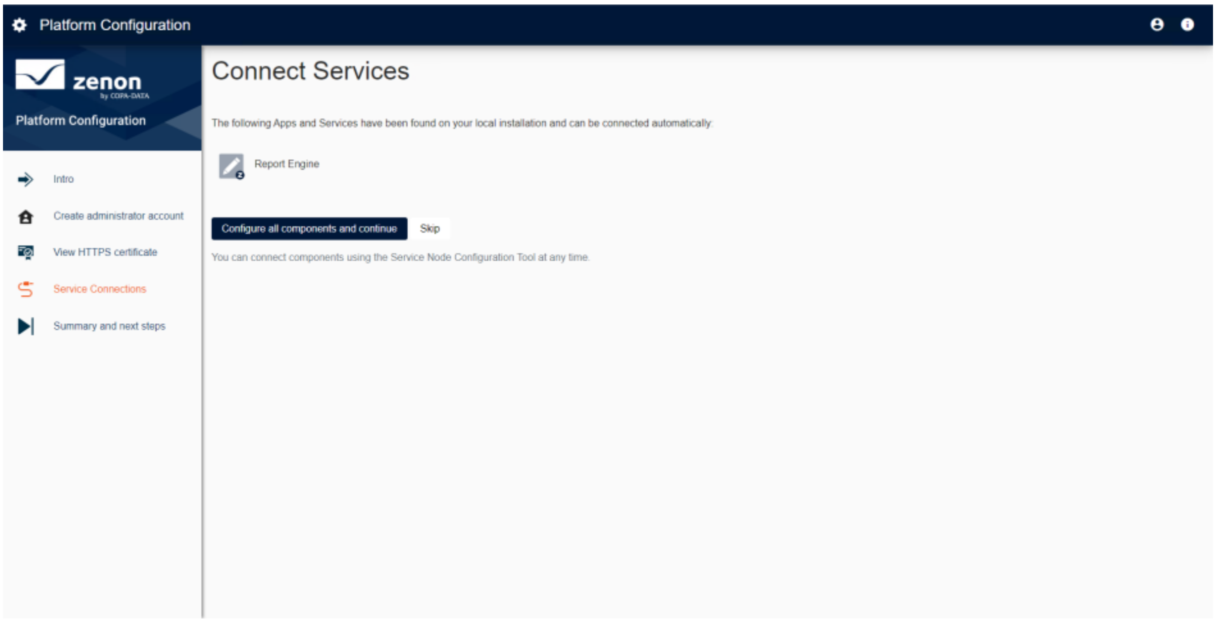
You can configure your own HTTPS certificate using the Service Configuration Studio at any time.

8.2.2.3.5 Service Connections

If you have installed additional **zenon** components, you can connect them to **IIoT Services**. Components are, for example, **Engineering Studio**, **Service Engine** or **Report Engine**.

Click, in the **Connect Services** window, on the **Configure all components and continue** button.

> ## Information
>
> In **Service Configuration Studio**, you have at all times the possibility to connect further **zenon** components to **IIoT Services**.



Platform Configuration

zenon by COPA-DATA

Platform Configuration

- Intro
- Create administrator account
- View HTTPS certificate
- Service Connections
- Summary and next steps

Connect Services

The following Apps and Services have been found on your local installation and can be connected automatically.

Report Engine

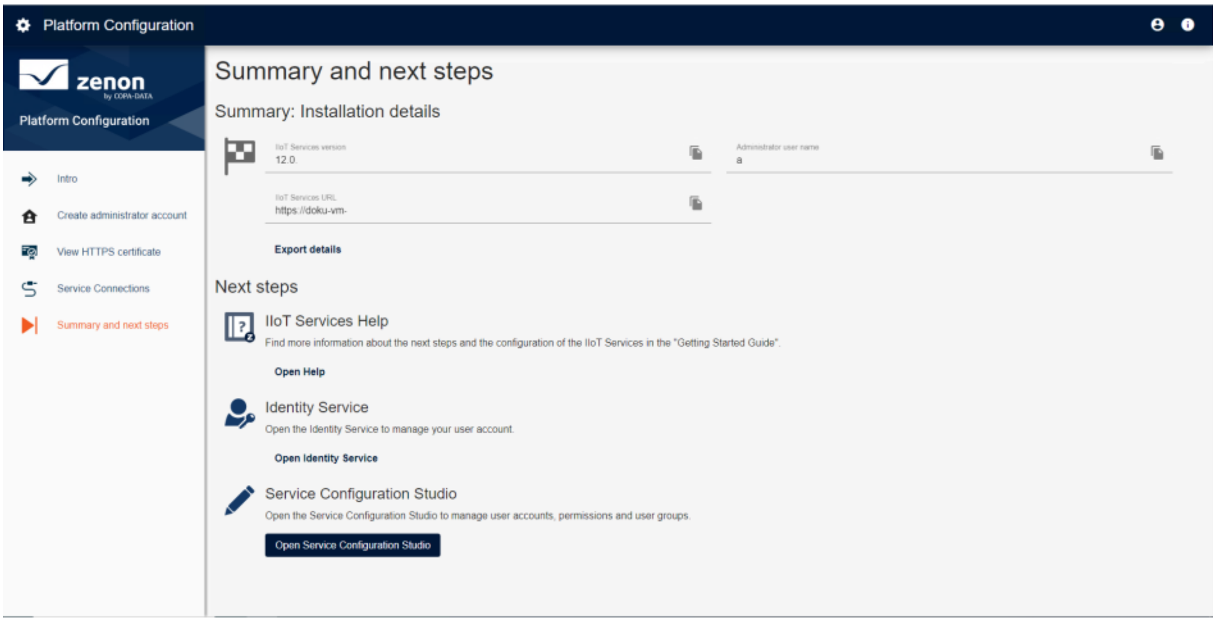
You can connect components using the Service Node Configuration Tool at any time.

8.2.2.3.6 Summary and next steps

Here you can find a summary of the installation details as well as information about the next steps.

You have the following possibilities:

1. Start the online help.
2. Edit your user account with the **Identity Service**.
3. You can administer users and authorizations In **Service Configuration Studio**, you can administer users, groups and permissions, among other things.



Platform Configuration

zenon by COPA-DATA

Platform Configuration

- Intro
- Create administrator account
- View HTTPS certificate
- Service Connections
- Summary and next steps

Summary and next steps

Summary: Installation details

IIoT Services version: 12.0.

Administrator user name: a

IIoT Services URL: https://doku-vm-

Next steps

IIoT Services Help

Find more information about the next steps and the configuration of the IIoT Services in the "Getting Started Guide".

Identity Service

Open the Identity Service to manage your user account.

Service Configuration Studio

Open the Service Configuration Studio to manage user accounts, permissions and user groups.

> 👍 **Tip**
>
> Create the link for **Service Configuration Studio** as a bookmark in your browser.

Name	Sample values	Description
Service Configuration Studio	*https://mycomputer.mydomain.com:9443* System-specific value*	In the **Service Configuration Studio**, only users with administrator rights can fully administer the IIoT Services.

* Replace mycomputer.mydomain.com in the URLs with your computer's FQDN (on page 126).

8.2.2.4Configuration

In this node, you can find information on setting up the following components:

1. Engineering Studio
2. Service Engine
3. IIoT API

8.2.2.4.1Engineering Studio

The connections must be configured beforehand in order for Engineering Studio and Service Engine to be able to communicate with IIoT Services.

8.2.2.4.2 Connection to IIoT Services

CREATE CONNECTION FOR A PROJECT

To do this, carry out the following steps:

1. Highlight the project in Engineering Studio.
2. Go to the **Network** node in the project properties
3. Go to property group **IIoT Services settings**.
4. Activate the **Activate IIoT Services** checkbox.
   This activates the configuration of the **Connection settings** property as well as the **... button**.

5. Click on the **... button** . The **IIoT Services Connection Wizard** is started.
6. Enter the URL of your IIoT Services installation and follow the instructions in the wizard.

   Skip the step for Report Engine.
7. Once the **IIoT Services Connection Wizard** has been successfully configured, you will find the used **IIoT Service URL** and the **Client-ID** in the input field of the connection settings.

Note: You can find further information on the **IIoT Services Connection Wizard** in the **IIoT Services Connection Wizard** node in the **Welcome** section.

8.2.2.4.3 Configure variables

In order to use variables in IIoT Services, they must be configured for it in Engineering Studio.

Only variables with **simple data type** are supported.

To configure variables:

1. Select the desired variable.
2. Open the **Authorization/eSignature** group in the properties.
3. Switch to the **IIoT Services settings** subgroup.
4. Configure the variable for use in IIoT Services.

Configurable properties:

Access permission

Access right of a variable in IIoT Services. Select from drop-down list:

- *None*: Variable is not available in IIoT Services.
- *Read*: IIoT Services has read access to this variable.
- *Read and write*: IIoT Services have read and write access to this variable.

Note: For reasons of security, access permission should only be set to the extent that they are actually required.

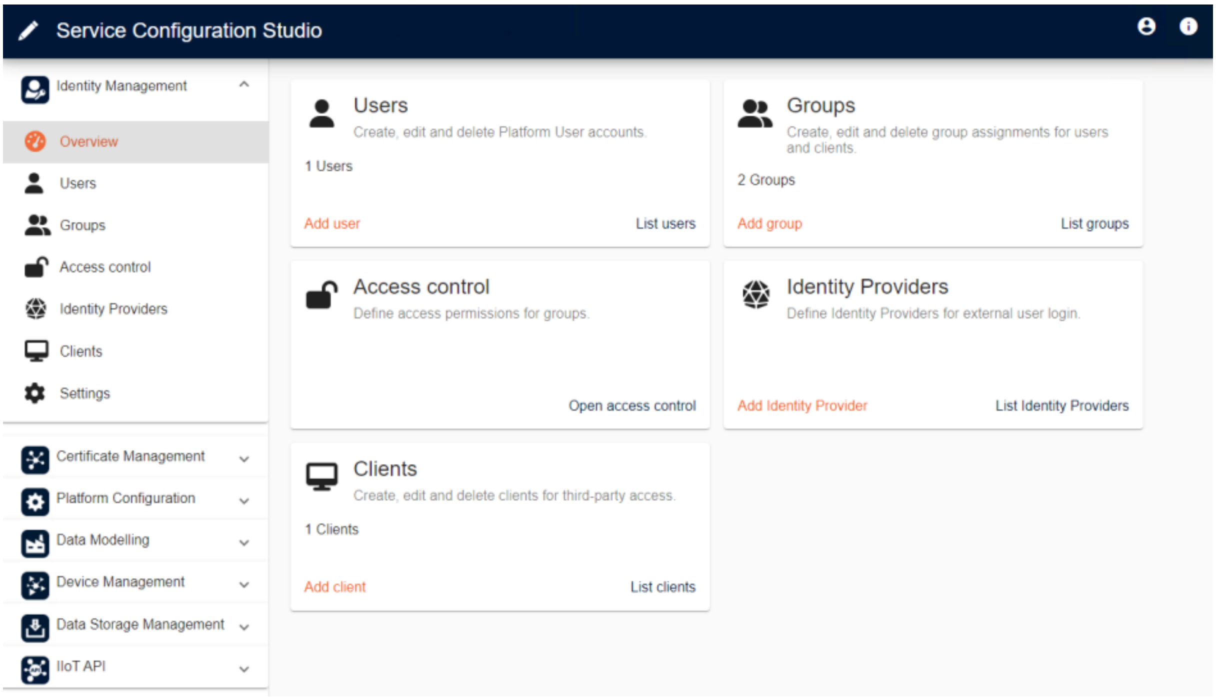
8.2.2.4.4 Starting Service Engine

Start **Service Engine** after configuration:

1. Save the project with all the changes.
2. Click on the **Geänderte Service Engine Dateien erzeugen** button.
3. Click on the **Service Engine starten** button.

8.2.2.4.5 Identity Management

In **Identity Management**, you administer users, groups, resources and privileges.



Create, edit and delete Platform User accounts.

Create, edit and delete group assignments for users and clients.

Define access permissions for groups.

Define Identity Providers for external user login.

Create, edit and delete clients for third-party access.

Assign the *Administrator* user the right to access Service Engine via IIoT Services.

To do this, follow the instructions in the following chapters.

8.2.2.4.6 Creating a group and adding users

To create a user group:

1. Navigate in **Service Configuration Studio** to the **Identity Management** menu item.
2. Click on the **Groups** submenu.
3. Click on the **Add Group** button.
4. Assign the group name *Users*.
5. Click on **Add**.

The *Users* user group has been created.

To add a user to the **Users** user group:

1. Select the **Users** group.
2. Click on the **Add user** button.
3. Select the *Administrator* user.

   Note: The *Administrator* user is displayed in the list as *admin admin* (first name; last name).
4. Click on **Add**.

You have thus added the *Administrator* user of the *Users* user group.

8.2.2.4.7 Add resource and add role

To add the Service Engine resource to the *Users* user group:

1. Ensure that Service Engine has been started.
2. Navigate in **Service Configuration Studio** to the **Identity Management** menu item.
3. Click on the **Access Control** submenu.
4. Select the *Users* user group under **Groups**.
5. Click on the **Add Resources** button.
6. Select your project in the overview.
7. Click on the **Add** button.

You have thus added the resource to the *Users* user group.

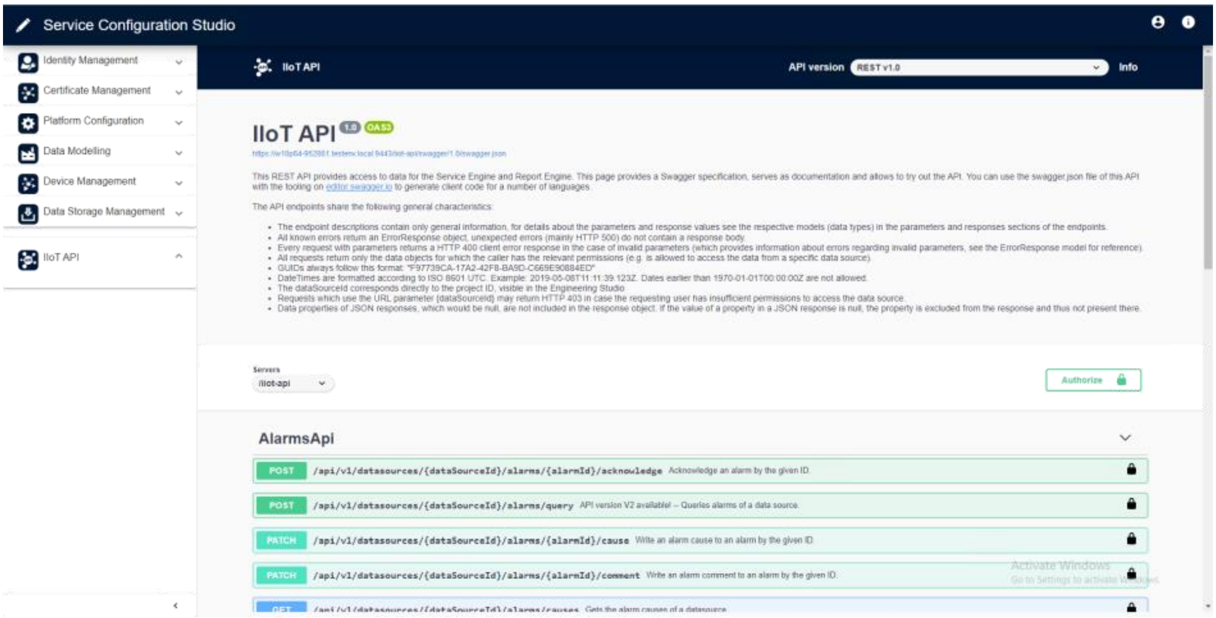
To assign the necessary role to the resource:

1. Click on the **... button** under **Assigned Resources .**
2. Select **Manage roles**.
3. Select the following permission: *Data Read*.
4. Click on the **Submit** button.

You have thus assigned the necessary role to the resource. Tthe *Administrator* user can access the released variables in Service Engine via IIoT API.

8.2.2.4.8 IIoT API

In **Service Configuration Studio**, you access the IIoT API manually as a user. With the IIoT API, you can retrieve data from the IIoT Services.



IIoT API

https://w10p64-952881.testenv.local:9443/iiot-api/swagger/1.0/swagger.json

This REST API provides access to data for the Service Engine and Report Engine. This page provides a Swagger specification, serves as documentation and allows to try out the API. You can use the swagger.json file of this API with the tooling on editor.swagger.io to generate client code for a number of languages.

The API endpoints share the following general characteristics:

- The endpoint descriptions contain only general information, for details about the parameters and response values see the respective models (data types) in the parameters and responses sections of the endpoints.
- All known errors return an ErrorResponse object, unexpected errors (mainly HTTP 500) do not contain a response body.
- Every request with parameters returns a HTTP 400 client error response in the case of invalid parameters (which provides information about errors regarding invalid parameters, see the ErrorResponse model for reference).
- All requests return only the data objects for which the caller has the relevant permissions (e.g. is allowed to access the data from a specific data source).
- GUIDs always follow this format: "F97739CA-17A2-42F8-BA9D-C669E90884ED"
- DateTimes are formatted according to ISO 8601 UTC. Example: 2019-05-08T11:11:39.123Z. Dates earlier than 1970-01-01T00:00:00Z are not allowed.
- The dataSourceId corresponds directly to the project ID, visible in the Engineering Studio
- Requests which use the URL parameter {dataSourceId} may return HTTP 403 in case the requesting user has insufficient permissions to access the data source.
- Data properties of JSON responses, which would be null, are not included in the response object. If the value of a property in a JSON response is null, the property is excluded from the response and thus not present there.

AlarmsApi

POST /api/v1/datasources/{dataSourceId}/alarms/{alarmId}/acknowledge Acknowledge an alarm by the given ID.

POST /api/v1/datasources/{dataSourceId}/alarms/query API version V2 available! – Queries alarms of a data source.

PATCH /api/v1/datasources/{dataSourceId}/alarms/{alarmId}/cause Write an alarm cause to an alarm by the given ID.

PATCH /api/v1/datasources/{dataSourceId}/alarms/{alarmId}/comment Write an alarm comment to an alarm by the given ID.

GET /api/v1/datasources/{dataSourceId}/alarms/causes Gets the alarm causes of a datasource.

There are two possibilities:

- For test purposes, you access the IIoT API manually in **Service Configuration Studio**.
- In a productive environment, a client application automatically accesses the IIoT API. To do this, you need an accordingly programmed 3rd party application.

8.2.2.4.9 User authorization

For a manual query using the IIoT API, you must authorize yourself. To authorize a user in the IIoT API:

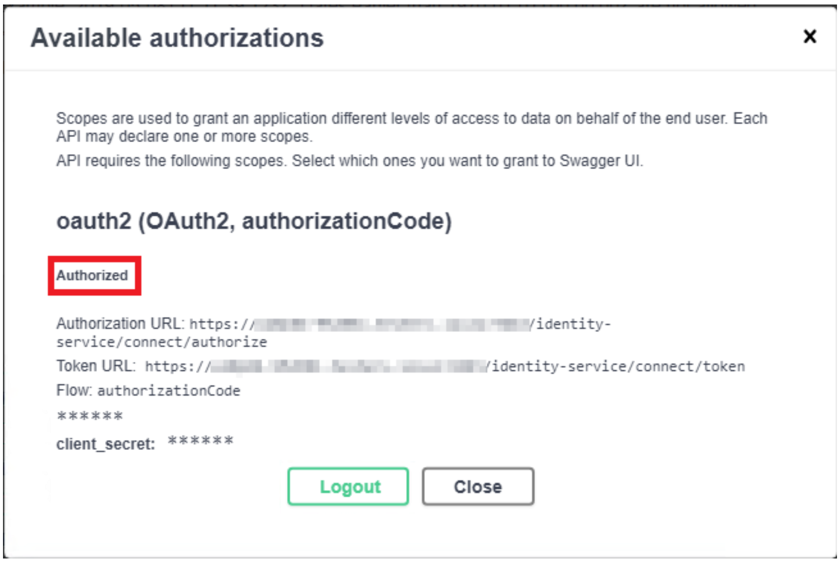
1. Ensure that Service Engine has been started.
2. Open the **Service Configuration Studio**.
3. Go to the **IIoT API** button.
4. Click on the green **Authorize** button. A window opens.
   Note: You are not authorized by default. The icon shows an opened lock.
5. Ensure that the value for the **client_id** field is set to *swagger_demo_api*.
6. Activate the following checkboxes:

- **iiotServicesAPI**
- **dataStorageAPI**

  Note: You thus determine the scope of the application.

7. Click on the **Authorize** button.
8. After successful authorization, the system shows the message *Authorized*.
9. Click on the **Close** button. Authorization remains active.

   Note: If you are authorized, you will see the locked icon.


Available authorizations

Scopes are used to grant an application different levels of access to data on behalf of the end user. Each API may declare one or more scopes.

API requires the following scopes. Select which ones you want to grant to Swagger UI.

oauth2 (OAuth2, authorizationCode)

Authorized

Authorization URL: https://[redacted]/identity-service/connect/authorize

Token URL: https://[redacted]/identity-service/connect/token

Flow: authorizationCode

******

client_secret: ******

> **Info**
>
> You can find the complete list of IIoT API error codes in the Troubleshooting node in the IIoT API error codes node.

8.2.2.4.10 Test 1: Query available project

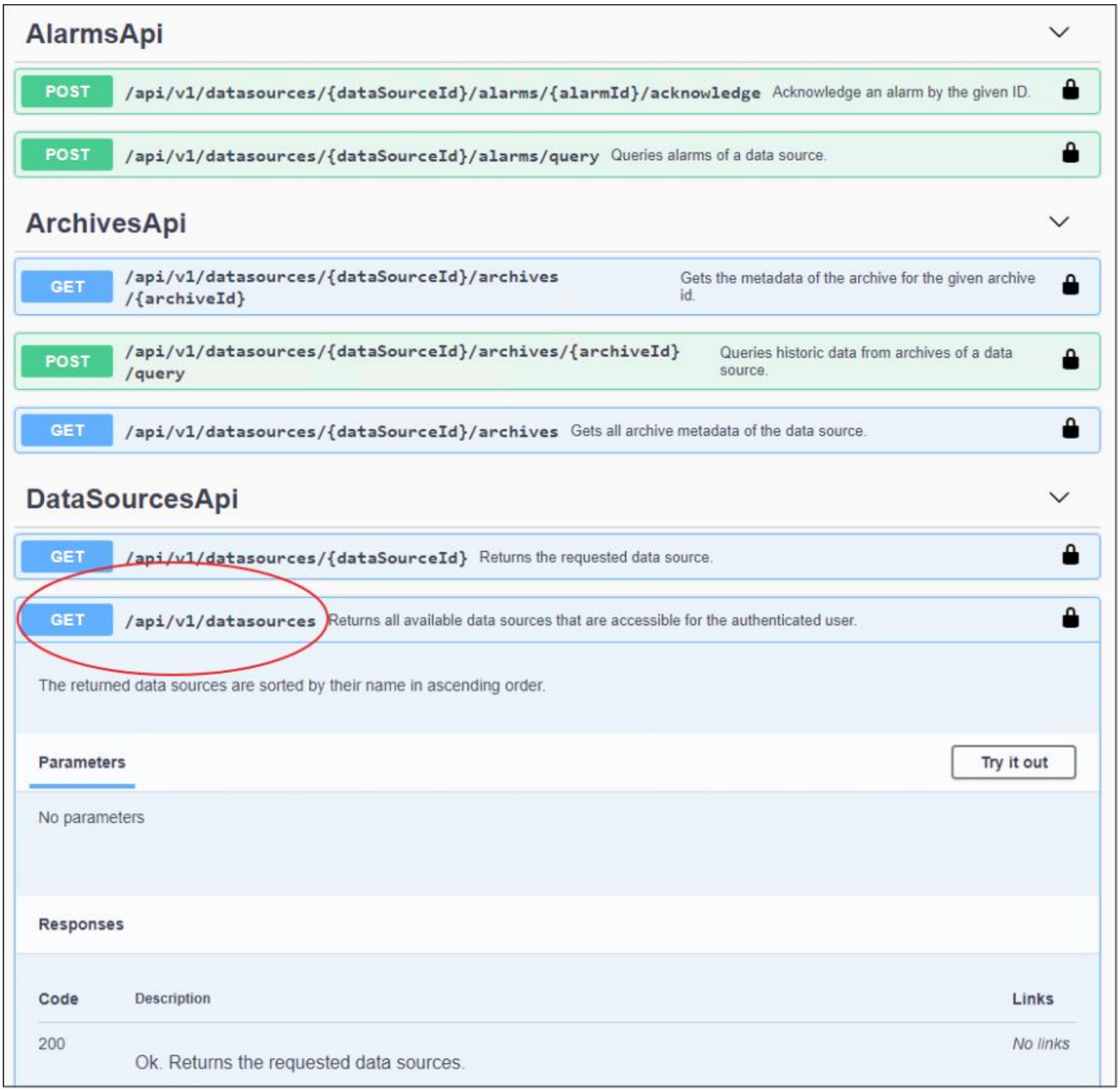
With this test, you check to see which projects you can use in Service Engine.

SELECT ENDPOINT

1. Ensure that Service Engine has been started.
2. Start **Service Configuration Studio**.
3. Go to the **IIoT API** button.
4. Ensure that the user authorization for the IIoT API (on page 119) has been carried out.

5. Check whether the value *REST v1.0* is set as **API version** in the header.
6. Go to the **DataSourcesApi** category.
7. Go within the category to the line with the */api/v1/datasources* endpoint.

You must configure this endpoint for the following query.



AlarmsApi

POST /api/v1/datasources/{dataSourceId}/alarms/{alarmId}/acknowledge Acknowledge an alarm by the given ID.

POST /api/v1/datasources/{dataSourceId}/alarms/query Queries alarms of a data source.

ArchivesApi

GET /api/v1/datasources/{dataSourceId}/archives/{archiveId} Gets the metadata of the archive for the given archive id.

POST /api/v1/datasources/{dataSourceId}/archives/{archiveId}/query Queries historic data from archives of a data source.

GET /api/v1/datasources/{dataSourceId}/archives Gets all archive metadata of the data source.

DataSourcesApi

GET /api/v1/datasources/{dataSourceId} Returns the requested data source.

GET /api/v1/datasources Returns all available data sources that are accessible for the authenticated user.

The returned data sources are sorted by their name in ascending order.

No parameters

Responses

Code	Description	Links
200	Ok. Returns the requested data sources.	*No links*

QUERY PROJECT

1. Click on the blue **GET** button in the line. This expands the endpoint.
2. Click on the **Try it out** button.
3. Click on the **Execute** button.

4. Copy the *dataSourceId* into a text file. You need this value for the following test.
   Note: It is identical to the project ID of your project.

RESULT

The query shows the available project.

Note: Ensure that the project is in the **Online** state.

Note for programmers

Code sample: Response body

```
{
"dataSources": [
{
"name": "ZENON14_DEMO",
"dataSourceId": "d3058681-c6a8-4b2e-908d-610676fce605",
"state": "Online"
}
]
}
```

8.2.2.4.11 Test 2: Query available variables and variable values

With this test, you will access the variables and variable values enabled in the zenon project via IIoT Services.

OPEN ENDPOINT

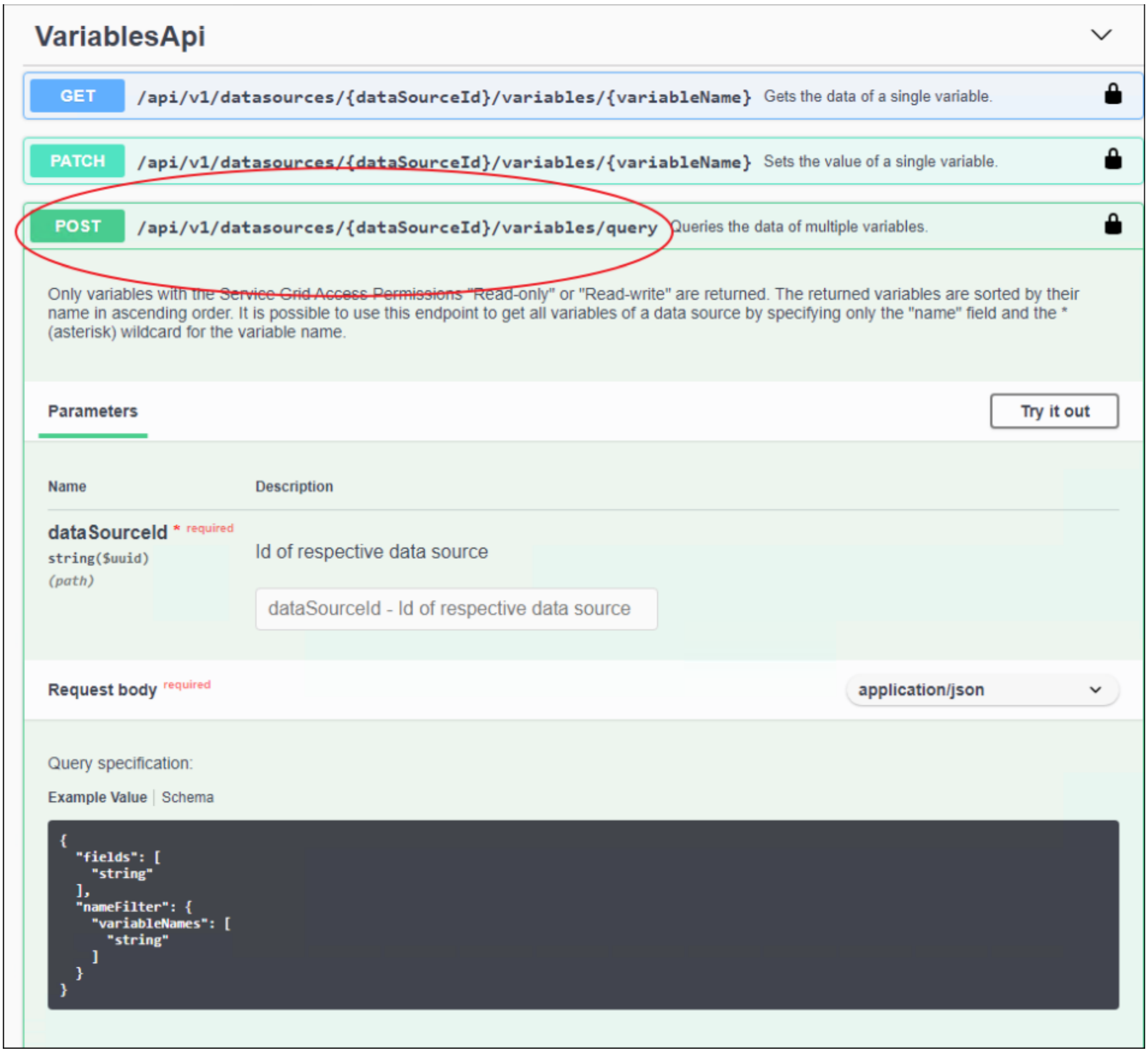
1. Ensure that Service Engine is running.
2. Ensure that the user authorization for the IIoT API (on page 119) has been carried out.
3. Start **Service Configuration Studio**.
4. Go to the **IIoT API** menu item.
5. Check whether the value *REST v1.0* is set as **API version** in the header.
6. Go to the **Variables API** category.
7. Go to the the line with the */api/v1/datasources/{dataSourceId}/variables/query}* endpoint.

You must configure the query in this endpoint.

CONFIGURE QUERY

1. Click on the green **Post** button.
2. Click on the **Try it out** button. You have thus activated the input field for the **dataSourceId**.
3. Enter the **dataSourceId** (identical to the zenon project ID).
   Note: You have thus defined the target project for the query. (Example: Initial query (on page 125))
4. Change the following points in the **Query specification**:
   a) *fields*: Replace the predefined "string" with "name", "value".
   You thus define the data fields for the query.
   b) *nameFilter*: Replace the predefined "string" with "*".
   You use this placeholder to query all values unfiltered. (Example: custom query (on page 125))
5. Then click on **Execute** to perform the query.
6. The query is acknowledged as follows: "Code 200" "Ok. Returns the queried variables.”
7. The "Response body" section shows the query result. (Example: query result (on page 126)).

The query result shows the released variables and their variable values from the specified zenon project.



VariablesApi

GET /api/v1/datasources/{dataSourceId}/variables/{variableName} Gets the data of a single variable.

PATCH /api/v1/datasources/{dataSourceId}/variables/{variableName} Sets the value of a single variable.

POST /api/v1/datasources/{dataSourceId}/variables/query Queries the data of multiple variables.

Only variables with the Service Grid Access Permissions "Read-only" or "Read-write" are returned. The returned variables are sorted by their name in ascending order. It is possible to use this endpoint to get all variables of a data source by specifying only the "name" field and the * (asterisk) wildcard for the variable name.

Name	Description
dataSourceId * required string($uuid) (path)	Id of respective data source dataSourceId - Id of respective data source

Request body required

Query specification:

```
{
  "fields": [
    "string"
  ],
  "nameFilter": {
    "variableNames": [
      "string"
    ]
  }
}
```

8.2.2.4.12 Query specifications

You can find the query specifications in this section.

8.2.2.4.13 Initial query

Code Sample:

```
{
"fields": [
"string"
],
"nameFilter": {
"variableNames": [
"string"
]
}}
```

Initial query

8.2.2.4.14 Custom query

Code Sample:

```
{
"fields": [
"name", "value"
],
"nameFilter": {
"variableNames": [
"*"
]
}}
```

Query of variables and variable values

8.2.2.4.15 Query result

Code Sample:

```
{
"variables": [
{
"name": "MY_VARIABLE",
"value": "1"
}
]
}
```

The shared variable and the variable value are in the "Response body" section.

8.2.2.5 Appendix

In this node, you can find further information on the following topics:

1. Determine FQDN (on page 126)
2. Monitoring services (on page 126)
3. Test environment vs. productive environment (on page 127)

8.2.2.5.1 Determine FQDN (Fully Qualified Domain Name)

To determine the **FQDN** of the Windows computer:

1. Open the command line using the **Windows + R** keyboard shortcut.
2. Enter **cmd.exe**.
3. Enter the **ping localhost** command.
4. The **Command Line Interface** shows your computer’s FQDN, among other things.
5. Convert the FQDN to lower-case letters.

You have now determined the FQDN that you need for use in the IIoT Services.

8.2.2.5.2 Monitor services

All services in IIoT Services are automatically started by the operating system. You can check the status of IIoT Services services with the following steps:

1. Open the command line using the **Windows + R** shortcut.

2. Enter **services.msc**.
3. Confirm the entry with **Enter**. This then opens the console for the administration of services.
4. You can find the services at: **zenon** <servicename>
5. All **zenon** services must in principle be in *running* state.
   Note: Data Storage can also be in **exited** state (because it is not used).

8.2.2.5.3 Test environment vs. productive environment

The test environment described in this guide is quicker and easier to set up than a typical productive environment.

The fundamental differences are:

	Test environment	Productive environment
Installation option	▸ IIoT Services (Windows native)	▸ IIoT Services (Windows native) ▸ IIoT Services (Docker on Linux)
Number of computers	▸ A computer for IIoT Services and all clients	▸ A computer for IIoT Services ▸ Dedicated computers for clients
Network topology	All applications run on the same computer.	The applications run on different computers. The computers can be distributed over different remote locations.
Passwords	It is possible to use predefined passwords in a protected test environment.	For all logins, it is essential that you assign your own secure passwords.

8.2.3 Welcome to COPA-DATA help

GENERAL HELP

If you cannot find any information you require in this help chapter or can think of anything that you would like added, please send an email to documentation@copadata.com.

LICENSES AND SERVICES

If you find that you need other zenon services or licenses, our staff will be happy to help you. Email sales@copadata.com.

PROJECT SUPPORT

You can receive support for any real project you may have from our customer service team, which you can contact via email at support@copadata.com.

8.2.4 Getting Started Guide (Docker)

This guide describes how you install and initially configure the IIoT Services on the Docker containerization software. It explains the installation of IIoT Services on Docker for Windows. It can also be used for Docker for Linux and other container solutions such as Podman or Kubernetes.

The container images supplied by COPA-DATA are based on Linux and can be run on Docker for Windows as well as on Docker for Linux and other container platforms.

The target group is users who have little or no experience with Docker.

> **Info**
>
> These instructions were written on an operating system in English.

8.2.4.1 System requirements

Note the following system requirements to install IIoT Services:

- General

  Docker, Engineering Studio and Service Engine are installed on one computer.

- Browser

  The following browsers are supported:

  - Google Chrome
  - Mozilla Firefox
  - Microsoft Edge
  - Apple Safari

  Note: Always use the most recent version of the respective browser.

- Storage space

  For the installation of IIoT Services, at least 6 GB of free storage space is required on the storage medium.

  You can find the required storage space for further zenon components in the **Installation and update** (on page 6) section in the **Engineering Studio** (on page 17) node.

- Requirements

  Ensure that the following points have been met:

  - Sufficient resources for the smooth operation of all installed applications (CPU, RAM, storage space).
  - The CPU must support hardware virtualization.
  - CPU hardware virtualization must be activated in the BIOS.
  - Working internet connection.
  - Ensure that you have Windows administrator rights on the computer.

8.2.4.2 Further requirements

To check your Docker installation, the following requirements must be met:

- An installation of Service Engine and Engineering Studio.
- Ensure that the version of IIoT Services Gateway is installed that corresponds to the version of IIoT Services to be installed.
- Ensure that this installation is licensed accordingly.

> **Information**
>
> In this guide, the entire installation is carried out on one computer.

Attention

If you run IIoT Services on a virtual machine with Docker containers:

First check whether AVX commands are supported by the underlying hypervisor.

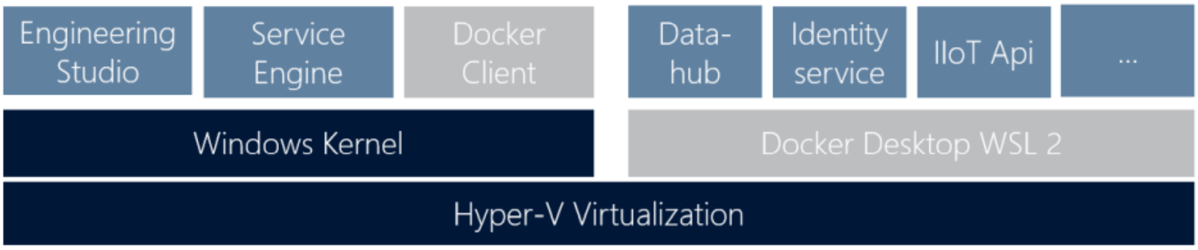
8.2.4.3 System architecture

In this guide, the entire installation is carried out on one computer. This includes Service Engine and Engineering Studio as a native Windows installation and at the same time IIoT Services as a Docker installation.

Note: You can obtain the container images from Docker Hub (https://hub.docker.com/u/copadata).

SHORT DESCRIPTION OF DOCKER

Docker is a technology for the container-based execution of software applications. The applications run in containers independently of the host operating system. The container images supplied by COPA-DATA are based on the Linux operating system. Thanks to the Windows Subsystem for Linux 2 (WSL 2), such Linux containers can also be executed on a Windows host operating system.



The Hyper-V hardware visualization layer from Microsoft hosts the Windows operating system and the Windows Subsystem for Linux 2 (WSL 2).

8.2.4.3.1Installation of Docker for Windows

Before you can install the **IIoT Services** for Docker on a Windows host system, you must first install Docker for Windows and the Windows Subsystem for Linux (WSL) 2 from Microsoft. Proceed in the following way:

1. Ensure that hardware virtualization (on page 151) is activated for the CPU.
2. Configure an *elevated PowerShell (on page 152)*. The is a PowerShell with administrator rights. You can use it to subsequently initialize and administer **IIoT Services**.

3. Download the current version of **Docker for Windows** from the Docker manufacturer’s web site (docker.com) (https://www.docker.com/).
4. Install **Docker for Windows** with the **WSL2 engine**.
5. Follow the link shown to *https://aka.ms/wsl2kernel*.
6. Download the *WSL2 Linux kernel update package for x64 machines*.
7. Install the update package
8. Start **Docker for Windows**.
9. Check whether Docker has been configured for the use of Linux containers. This is the default setting of Docker.

8.2.4.3.2 Installation of Docker for Linux

In this section, you can find brief instructions for installing **Docker für Ubuntu**.

1. Install Docker.
2. To do this, follow the detailed installation instructions on the provider's website (https://docs.docker.com/engine/install/ubuntu/).
   - Information about other distributions is also available on this website.
   - Keep in mind the requirements for installation.

- Configure your computer. To do this, follow the further instructions in this **Getting Started Guide**.

8.2.4.4Basic configuration of ENV file

In order to install IIoT Services, you must adapt the ENV file supplied by COPA-DATA. You will find this file in a package that you can download from the COPA-DATA website.

Carry out the following steps to configure the *ENV* file:

You can download the configuration files for IIoT Services from the COPA-DATA (https://www.copadata.com/en/downloads/product-downloads/) website.

The download contains:

- IIoT Services configuration files: *.env* and *docker-compose.yml*
- One PDF file: **IIoT Services**

To download the configuration files:

- Go to the COPA-DATA (https://www.copadata.com/en/downloads/product-downloads/) website.

- You must log in to the COPA-DATA website with your user account for this download. Registration is free.
- Then filter for IIoT Services in **Select Category**
- Download the *ZIP* file with the appropriate version of **IIoT Services** (Docker).

1. Create a working directory. This is used to store the Docker configuration files.

   Note: Under Windows, you can use *C:\iiot-services*as the working directory.

   Note: Under Linux, you can use */home/<user>/iiot-services*as the working directory.
2. Unzip the *ZIP* file into your working directory. There you will find the *ENV* file.
3. Open the *ENV* file with a text editor.
4. Enter the values for the corresponding configuration entries in the *ENV*.
5. Save the changes.
6. Check whether the *.env* file still contains the leading period (**"."**).
   Note: Some file operations in the Windows operating system can remove the dot. In this case, rename the file from **"env"** to **".env"**.

CONFIGURE ENV FILE

The following entries are necessary for the configuration of the *ENV* file.

Entry	Sample values	Description
Datenbank		
Persistence_Username=	*iiot_user*	You can choose the usernames yourself.
Persistence_Password=	*iiot_Changeme123!*	You can define the password yourself. **Note**: Note the minimum password requirements!
Persistence_Uri=		Optional entry; is not needed
Machine settings		
MACHINE_HOSTNAME=	*mycomputer.mydomain.com* System-specific value: - Determine the FQDN host name of your Windows computer. (To	Frequent configuration errors on MACHINE HOSTNAME are: - Use of capital letters

Entry	Sample values	Description
	do this, use the command line command **ping localhost**) - FQDN must be entered in continuous lowercase letters.	

8.2.4.5Commissioning

Once you have entered the configuration values into the *ENV* file, you can initialize IIoT Services.

To do this, carry out the following steps:

1. Download Docker images (on page 134)
2. Initialize IIoT Services (on page 134)
3. Start and monitor services

8.2.4.5.1Download Docker images

Carry out the following steps to download the Docker images:

1. Start Docker.
2. Ensure that the *ENV* file (on page 132) has been fully configured.
3. Open the *elevated PowerShell*. (Windows) or a terminal (Linux).
4. Change to your working directory where you have saved the Docker configuration files.
5. Download the containers with the command.
   docker compose pull

   This downloads the images from *hub.docker.com* (*https://hub.docker.com/u/copadata*).

8.2.4.5.2 Initialize IIoT Services

Then carry out the following steps to install IIoT Services:

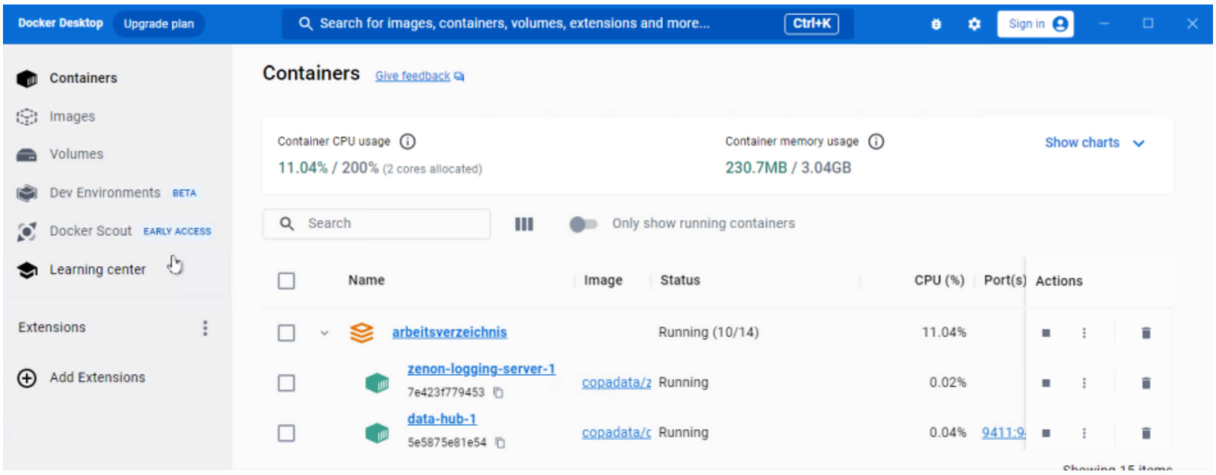
1. Start the containers with the command

   docker compose up -d

2. Windows only: If necessary, confirm the Windows firewall enable for IIoT Services.
   Note: The firewall blocks the services until they are approved. This can lead to timeouts. The initialization will fail as a result. In this case, you must restart the initialization.
3. Check if all containers are in the *Running* state.

   docker ps -a

   Note: Under Windows, you can also check the state of the container in the **Docker Dashboard**.



Container CPU usage
11.04% / 200% (2 cores allocated)

Container memory usage
230.7MB / 3.04GB

	Name	Image	Status	CPU (%)	Port(s)	Actions
□	arbeitsverzeichnis		Running (10/14)	11.04%		■ ⋮ 🗑
□	zenon-logging-server-1 7e423f779453	copadata/z	Running	0.02%		■ ⋮ 🗑
□	data-hub-1 5e5875e81e54	copadata/c	Running	0.04%	9411:9	■ ⋮ 🗑

8.2.4.6Configure IIoT Services

In this section, you will find information on the following topics:

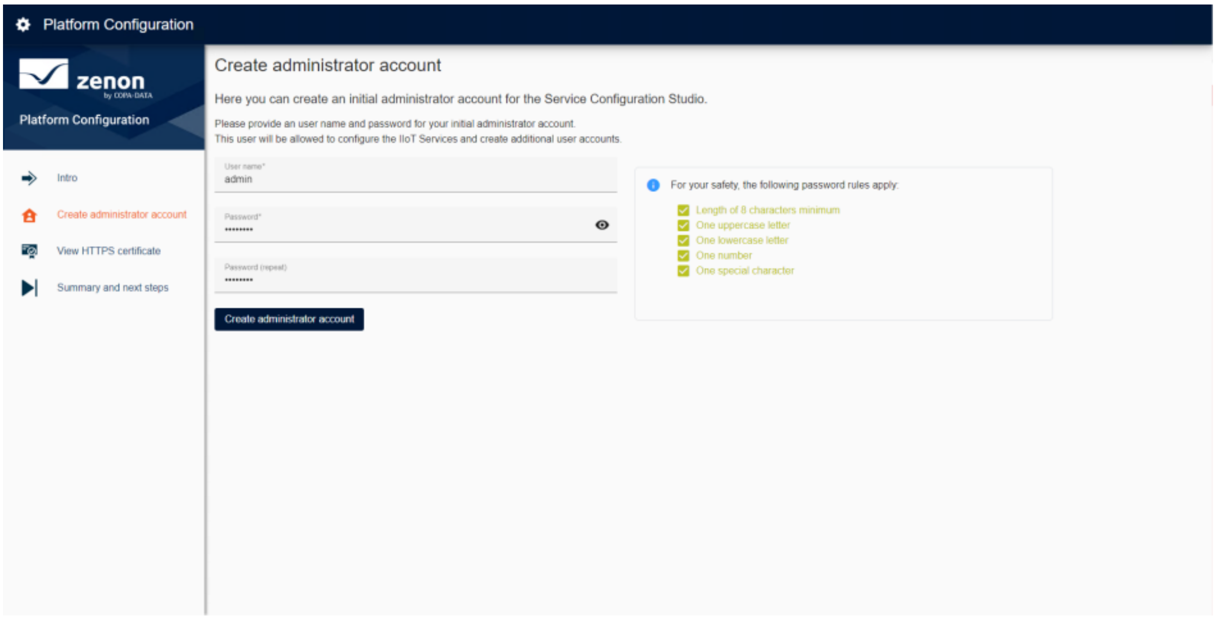
8.2.4.6.1Create administrator account

After the first installation of the IIoT Services, use the following steps to configure the administrator account.

1. Open the following address in the browser *https://mycomputer.mydomain.com:9443* and follow the steps set out.
2. In the **Platform Configuration** window, click on the **Get started** button**.**
3. Enter a user name in the **Create administrator account** window.
4. Enter a password. Note the given password criteria. If the password criteria are adhered to, the font color changes to green.

5. Enter the password again. If the two entries of the password match, the **Create administrator account** button is activated.

6. Click on this button. The creation of the administrator account is thus completed.



Create administrator account

Here you can create an initial administrator account for the Service Configuration Studio.

Please provide an user name and password for your initial administrator account.
This user will be allowed to configure the IIoT Services and create additional user accounts.

User name*
admin

Password*
••••••••

Password (repeat)
••••••••

Create administrator account

For your safety, the following password rules apply:

☑ Length of 8 characters minimum
☑ One uppercase letter
☑ One lowercase letter
☑ One number
☑ One special character

Note: This user is also authorized to configure IIoT Services and to create further users

> **⚠Attention**
>
> Note the password in a secure place. If the password is forgotten, there is no possibility to retrieve it.

8.2.4.6.2 HTTPS certificate - Creating a trust relationship

IIoT Services use an HTTPS certificate for secure communication. To trust the HTTPS certificate, the root certificate must be trusted.

To install the root certificate, proceed as follows:

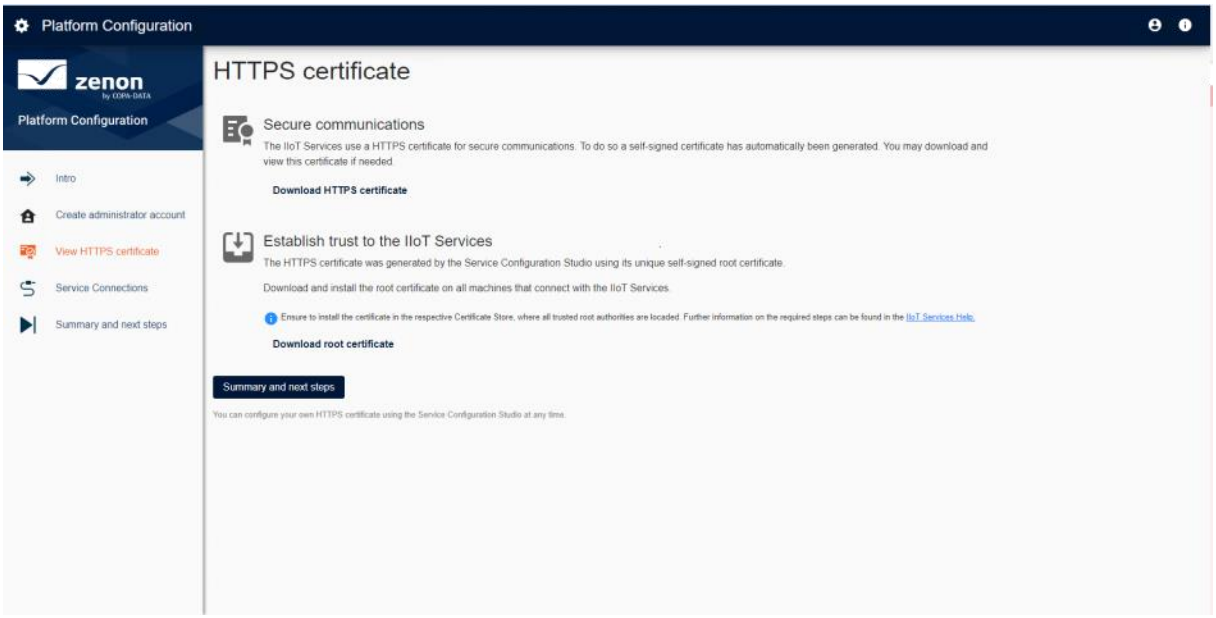
1. Click, in the **HTTPS certificate** window, on the **Download root certificate** button.

2. Open the downloaded certificate and install it in the **Trusted Root Certification Authorities Store** of Windows. To do this, open the Microsoft app *Certlm.msc*. Navigate to the node **Trusted Root Certification Authorities Store -> Certificates**. In the context menu of the **Certificates** node, you fill find the menu item: **All tasks > Import**. A wizard opens with which you can import the certificate.
   You can find further information on this in the **Trust** section and in the **Configure trust** chapter.

3. After successful installation of the root certificate, click on the **Summary and next steps** button.

> ### Attention
>
> When installing IIoT Services for the first time, your browser will issue a security warning. At this stage, you cannot verify the certificate yet. To complete the installation, you must ignore the security warning this one time.

Note: Also install the root certificate on all clients that you want to connect to IIoT Services.



HTTPS certificate

Secure communications

The IIoT Services use a HTTPS certificate for secure communications. To do so a self-signed certificate has automatically been generated. You may download and view this certificate if needed.

Establish trust to the IIoT Services

The HTTPS certificate was generated by the Service Configuration Studio using its unique self-signed root certificate.

Download and install the root certificate on all machines that connect with the IIoT Services.

Ensure to install the certificate in the respective Certificate Store, where all trusted root authorities are located. Further information on the required steps can be found in the IIoT Services Help.

You can configure your own HTTPS certificate using the Service Configuration Studio at any time.

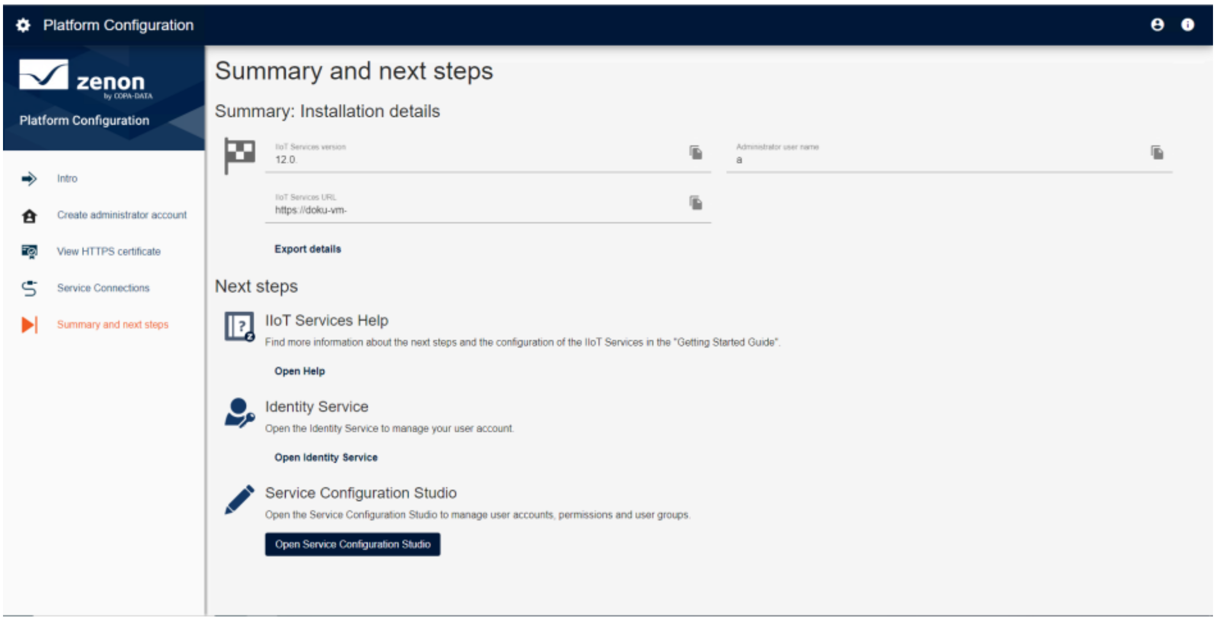
8.2.4.6.3 Summary and next steps

Here you can find a summary of the installation details as well as information about the next steps.

You have the following possibilities:

1. Start the online help.
2. Edit your user account with the **Identity Service**.

3. You can administer users and authorizations In **Service Configuration Studio**, you can administer users, groups and permissions, among other things.



Platform Configuration

zenon by COPA-DATA

Platform Configuration

- Intro
- Create administrator account
- View HTTPS certificate
- Service Connections
- Summary and next steps

Summary and next steps

Summary: Installation details

IIoT Services version: 12.0

Administrator user name: a

IIoT Services URL: https://doku-vm-

Export details

Next steps

IIoT Services Help

Find more information about the next steps and the configuration of the IIoT Services in the "Getting Started Guide".

Open Help

Identity Service

Open the Identity Service to manage your user account.

Open Identity Service

Service Configuration Studio

Open the Service Configuration Studio to manage user accounts, permissions and user groups.

Open Service Configuration Studio

Tip

Create the link for **Service Configuration Studio** as a bookmark in your browser.

Name	Sample values	Description
Service Configuration Studio	*https://mycomputer.mydomain.com:9443* System-specific value*	In the **Service Configuration Studio**, only users with administrator rights can fully administer the IIoT Services.

* Replace mycomputer.mydomain.com in the URLs with your computer's FQDN (on page 126).

8.2.4.7 Configuration

In this section, you will find information for setting up the following components:

1. Engineering Studio
2. Service Engine
3. IIoT API

8.2.4.7.1Engineering Studio

The connections must be configured beforehand in order for Engineering Studio and Service Engine to be able to communicate with IIoT Services.

8.2.4.7.2 Connection to IIoT Services

CREATE CONNECTION FOR A PROJECT

To do this, carry out the following steps:

1. Highlight the project in Engineering Studio.
2. Go to the **Network** node in the project properties
3. Go to property group **IIoT Services settings**.
4. Activate the **Activate IIoT Services** checkbox.
   This activates the configuration of the **Connection settings** property as well as the **... button**.
5. Click on the **... button** . The **IIoT Services Connection Wizard** is started.
6. Enter the URL of your IIoT Services installation and follow the instructions in the wizard.

   Skip the step for Report Engine.
7. Once the **IIoT Services Connection Wizard** has been successfully configured, you will find the used **IIoT Service URL** and the **Client-ID** in the input field of the connection settings.

Note: You can find further information on the **IIoT Services Connection Wizard** in the **IIoT Services Connection Wizard** node in the **Welcome** section.

8.2.4.7.3 Configure variables

In order to use variables in IIoT Services, they must be configured for it in Engineering Studio.

Only variables with **simple data type** are supported.

To configure variables:

1. Select the desired variable.
2. Open the **Authorization/eSignature** group in the properties.
3. Switch to the **IIoT Services settings** subgroup.
4. Configure the variable for use in IIoT Services.

Configurable properties:

Access permission

Access right of a variable in IIoT Services. Select from drop-down list:

- *None*: Variable is not available in IIoT Services.
- *Read*: IIoT Services has read access to this variable.
- *Read and write*: IIoT Services have read and write access to this variable.

Note: For reasons of security, access permission should only be set to the extent that they are actually required.

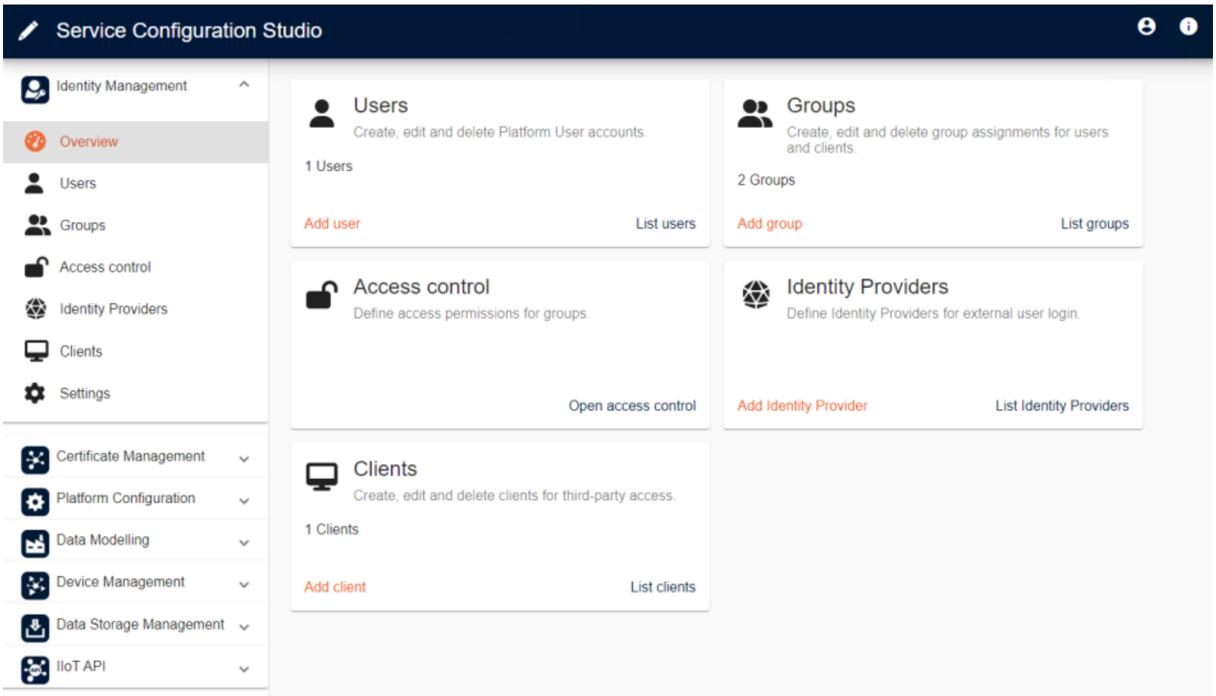
8.2.4.7.4 Starting Service Engine

Start **Service Engine** after configuration:

1. Save the project with all the changes.
2. Click on the **Geänderte Service Engine Dateien erzeugen** button.
3. Click on the **Service Engine starten** button.

8.2.4.7.5 Identity Management

In **Identity Management**, you administer users, groups, resources and privileges.



Users

Create, edit and delete Platform User accounts.

1 Users

Groups

Create, edit and delete group assignments for users and clients.

2 Groups

Access control

Define access permissions for groups.

Identity Providers

Define Identity Providers for external user login.

Clients

Create, edit and delete clients for third-party access.

1 Clients

Assign the *Administrator* user the right to access Service Engine via IIoT Services.

To do this, follow the instructions in the following chapters.

8.2.4.7.6 Creating a group and adding users

To create a user group:

1. Navigate in **Service Configuration Studio** to the **Identity Management** menu item.
2. Click on the **Groups** submenu.
3. Click on the **Add Group** button.
4. Assign the group name *Users*.
5. Click on **Add**.

The *Users* user group has been created.

To add a user to the **Users** user group:

1. Select the **Users** group.
2. Click on the **Add user** button.
3. Select the *Administrator* user.

   Note: The *Administrator* user is displayed in the list as *admin admin* (first name; last name).
4. Click on **Add**.

You have thus added the *Administrator* user of the *Users* user group.

8.2.4.7.7 Add resource and add role

To add the Service Engine resource to the *Users* user group:

1. Ensure that Service Engine has been started.
2. Navigate in **Service Configuration Studio** to the **Identity Management** menu item.
3. Click on the **Access Control** submenu.
4. Select the *Users* user group under **Groups**.
5. Click on the **Add Resources** button.
6. Select your project in the overview.
7. Click on the **Add** button.

You have thus added the resource to the *Users* user group.

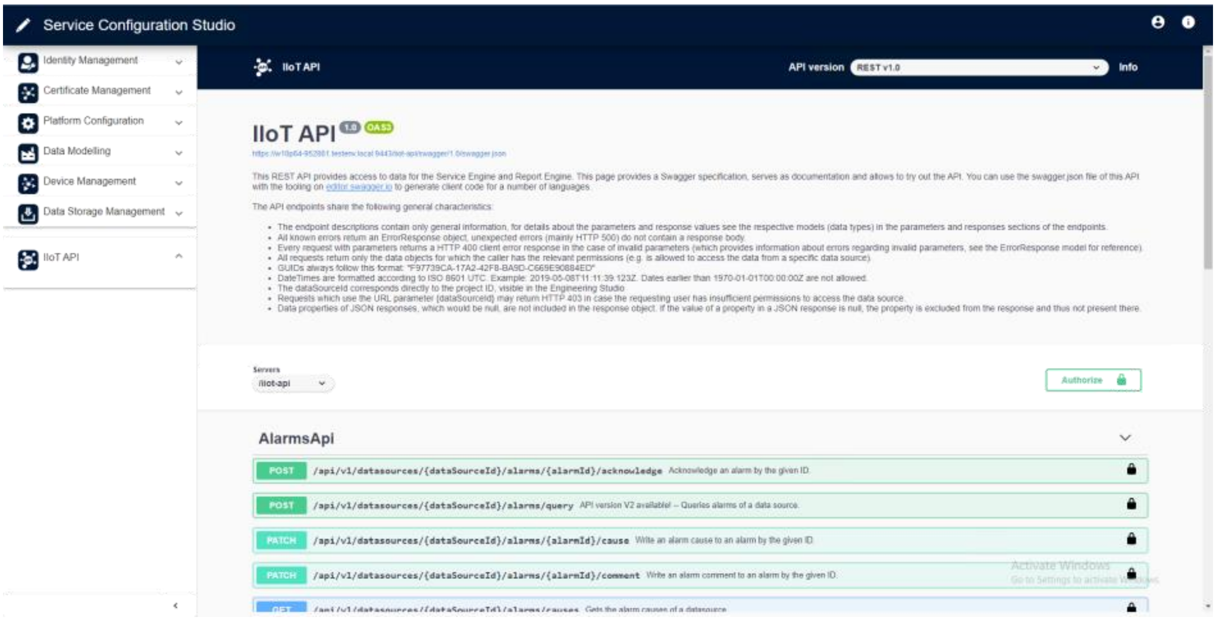
To assign the necessary role to the resource:

1. Click on the **... button** under **Assigned Resources .**
2. Select **Manage roles**.
3. Select the following permission: *Data Read*.
4. Click on the **Submit** button.

You have thus assigned the necessary role to the resource. Tthe *Administrator* user can access the released variables in Service Engine via IIoT API.

8.2.4.7.8 IIoT API

In **Service Configuration Studio**, you access the IIoT API manually as a user. With the IIoT API, you can retrieve data from the IIoT Services.



https://w10p64-952881.testenv.local:9443/iiot-api/swagger/1.0/swagger.json

This REST API provides access to data for the Service Engine and Report Engine. This page provides a Swagger specification, serves as documentation and allows to try out the API. You can use the swagger.json file of this API with the tooling on editor.swagger.io to generate client code for a number of languages.

The API endpoints share the following general characteristics:

- The endpoint descriptions contain only general information, for details about the parameters and response values see the respective models (data types) in the parameters and responses sections of the endpoints.
- All known errors return an ErrorResponse object, unexpected errors (mainly HTTP 500) do not contain a response body.
- Every request with parameters returns a HTTP 400 client error response in the case of invalid parameters (which provides information about errors regarding invalid parameters, see the ErrorResponse model for reference).
- All requests return only the data objects for which the caller has the relevant permissions (e.g. is allowed to access the data from a specific data source).
- GUIDs always follow this format: "F97739CA-17A2-42F8-BA9D-C669E90884ED"
- DateTimes are formatted according to ISO 8601 UTC. Example: 2019-05-08T11:11:39.123Z. Dates earlier than 1970-01-01T00:00:00Z are not allowed.
- The dataSourceId corresponds directly to the project ID, visible in the Engineering Studio.
- Requests which use the URL parameter {dataSourceId} may return HTTP 403 in case the requesting user has insufficient permissions to access the data source.
- Data properties of JSON responses, which would be null, are not included in the response object. If the value of a property in a JSON response is null, the property is excluded from the response and thus not present there.

```
POST   /api/v1/datasources/{dataSourceId}/alarms/{alarmId}/acknowledge  Acknowledge an alarm by the given ID.
POST   /api/v1/datasources/{dataSourceId}/alarms/query  API version V2 available! – Queries alarms of a data source.
PATCH  /api/v1/datasources/{dataSourceId}/alarms/{alarmId}/cause  Write an alarm cause to an alarm by the given ID.
PATCH  /api/v1/datasources/{dataSourceId}/alarms/{alarmId}/comment  Write an alarm comment to an alarm by the given ID.
GET    /api/v1/datasources/{dataSourceId}/alarms/causes  Gets the alarm causes of a datasource.
```

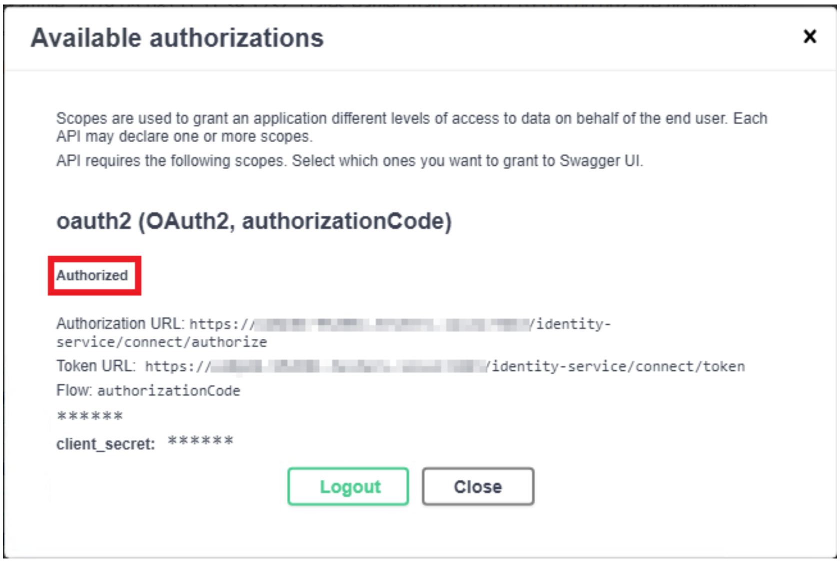
There are two possibilities:

- For test purposes, you access the IIoT API manually in **Service Configuration Studio**.
- In a productive environment, a client application automatically accesses the IIoT API. To do this, you need an accordingly programmed 3rd party application.

8.2.4.7.9 User authorization

For a manual query using the IIoT API, you must authorize yourself. To authorize a user in the IIoT API:

1. Ensure that Service Engine has been started.
2. Open the **Service Configuration Studio**.
3. Go to the **IIoT API** button.
4. Click on the green **Authorize** button. A window opens.
   Note: You are not authorized by default. The icon shows an opened lock.
5. Ensure that the value for the **client_id** field is set to *swagger_demo_api*.
6. Activate the following checkboxes:
   - **iiotServicesAPI**
   - **dataStorageAPI**
     Note: You thus determine the scope of the application.
7. Click on the **Authorize** button.
8. After successful authorization, the system shows the message *Authorized*.
9. Click on the **Close** button. Authorization remains active.
   Note: If you are authorized, you will see the locked icon.



Available authorizations ×

Scopes are used to grant an application different levels of access to data on behalf of the end user. Each API may declare one or more scopes.

API requires the following scopes. Select which ones you want to grant to Swagger UI.

oauth2 (OAuth2, authorizationCode)

Authorized

Authorization URL: `https://`[redacted]`/identity-service/connect/authorize`

Token URL: `https://`[redacted]`/identity-service/connect/token`

Flow: `authorizationCode`

******

client_secret: ******

Info

You can find the complete list of IIoT API error codes in the Troubleshooting node in the IIoT API error codes node.

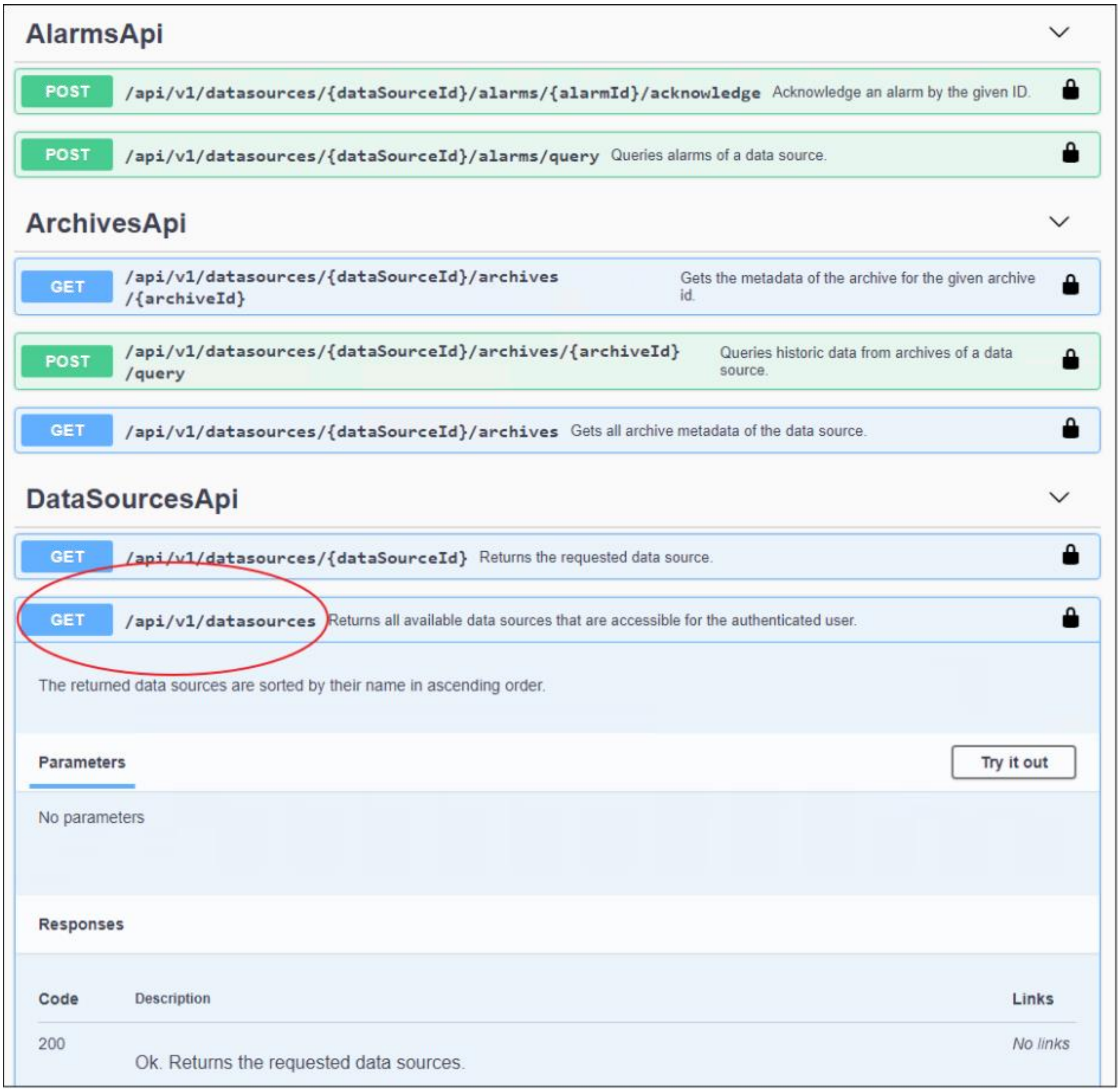
8.2.4.7.10 Test 1: Query available project

With this test, you check to see which projects you can use in Service Engine.

SELECT ENDPOINT

1. Ensure that Service Engine has been started.
2. Start **Service Configuration Studio**.
3. Go to the **IIoT API** button.
4. Ensure that the user authorization for the IIoT API (on page 119) has been carried out.
5. Check whether the value *REST v1.0* is set as **API version** in the header.
6. Go to the **DataSourcesApi** category.
7. Go within the category to the line with the */api/v1/datasources* endpoint.

You must configure this endpoint for the following query.



AlarmsApi

POST `/api/v1/datasources/{dataSourceId}/alarms/{alarmId}/acknowledge` Acknowledge an alarm by the given ID.

POST `/api/v1/datasources/{dataSourceId}/alarms/query` Queries alarms of a data source.

ArchivesApi

GET `/api/v1/datasources/{dataSourceId}/archives/{archiveId}` Gets the metadata of the archive for the given archive id.

POST `/api/v1/datasources/{dataSourceId}/archives/{archiveId}/query` Queries historic data from archives of a data source.

GET `/api/v1/datasources/{dataSourceId}/archives` Gets all archive metadata of the data source.

DataSourcesApi

GET `/api/v1/datasources/{dataSourceId}` Returns the requested data source.

GET `/api/v1/datasources` Returns all available data sources that are accessible for the authenticated user.

The returned data sources are sorted by their name in ascending order.

Parameters

No parameters

Responses

Code	Description	Links
200	Ok. Returns the requested data sources.	*No links*

QUERY PROJECT

1. Click on the blue **GET** button in the line. This expands the endpoint.
2. Click on the **Try it out** button.
3. Click on the **Execute** button.
4. Copy the *dataSourceId* into a text file. You need this value for the following test.
   Note: It is identical to the project ID of your project.

RESULT

The query shows the available project.

Note: Ensure that the project is in the **Online** state.

Note for programmers

Code sample: Response body

```
{
"dataSources": [
{
"name": "ZENON14_DEMO",
"dataSourceId": "d3058681-c6a8-4b2e-908d-610676fce605",
"state": "Online"
}
]
}
```

8.2.4.7.11 Test 2: Query available variables and variable values

With this test, you will access the variables and variable values enabled in the zenon project via IIoT Services.

OPEN ENDPOINT

1. Ensure that Service Engine is running.
2. Ensure that the user authorization for the IIoT API (on page 119) has been carried out.
3. Start **Service Configuration Studio**.
4. Go to the **IIoT API** menu item.
5. Check whether the value *REST v1.0* is set as **API version** in the header.
6. Go to the **Variables API** category.
7. Go to the the line with the */api/v1/datasources/{dataSourceId}/variables/query}* endpoint.

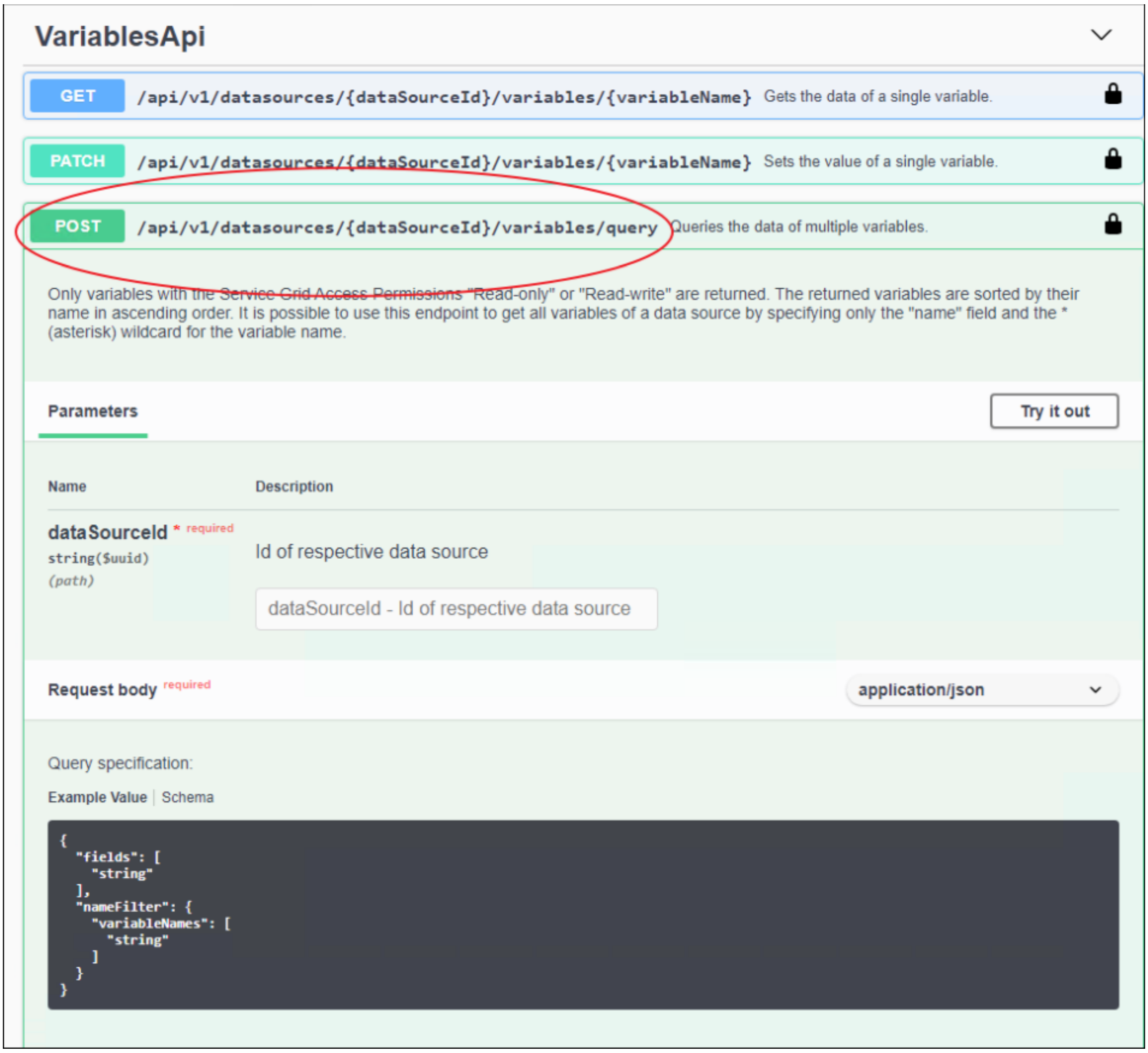
You must configure the query in this endpoint.

CONFIGURE QUERY

1. Click on the green **Post** button.

2. Click on the **Try it out** button. You have thus activated the input field for the **dataSourceId**.
3. Enter the **dataSourceId** (identical to the zenon project ID).
   Note: You have thus defined the target project for the query. (Example: Initial query (on page 125))
4. Change the following points in the **Query specification**:
   a) *fields*: Replace the predefined "string" with "name", "value".
   You thus define the data fields for the query.
   b) *nameFilter*: Replace the predefined "string" with "*".
   You use this placeholder to query all values unfiltered. (Example: custom query (on page 125))
5. Then click on **Execute** to perform the query.
6. The query is acknowledged as follows: "Code 200" "Ok. Returns the queried variables.”
7. The "Response body" section shows the query result. (Example: query result (on page 126)).

The query result shows the released variables and their variable values from the specified zenon project.



VariablesApi

/api/v1/datasources/{dataSourceId}/variables/{variableName} Gets the data of a single variable.

/api/v1/datasources/{dataSourceId}/variables/{variableName} Sets the value of a single variable.

/api/v1/datasources/{dataSourceId}/variables/query Queries the data of multiple variables.

Only variables with the Service Grid Access Permissions "Read-only" or "Read-write" are returned. The returned variables are sorted by their name in ascending order. It is possible to use this endpoint to get all variables of a data source by specifying only the "name" field and the * (asterisk) wildcard for the variable name.

Name	Description
dataSourceId * required string($uuid) *(path)*	Id of respective data source dataSourceId - Id of respective data source

Request body required — application/json

Query specification:

```
{
  "fields": [
    "string"
  ],
  "nameFilter": {
    "variableNames": [
      "string"
    ]
  }
}
```

8.2.4.7.12 Query specifications

You can find the query specifications in this section.

8.2.4.7.13 Initial query

Code Sample:

```
{
"fields": [
"string"
],
"nameFilter": {
"variableNames": [
"string"
]
}}
```

Initial query

8.2.4.7.14 Custom query

Code Sample:

```
{
"fields": [
"name", "value"
],
"nameFilter": {
"variableNames": [
"*"
]
}}
```

Query of variables and variable values

8.2.4.7.15 Query result

Code Sample:

```
{
"variables": [
{
"name": "MY_VARIABLE",
"value": "1"
}
]
}
```

The shared variable and the variable value are in the "Response body" section.

8.2.4.8 Appendix

Here you can find information on the following topics:

1. Test environment vs. productive environment (on page 150)
2. Determine FQDN (on page 126)
3. Check CPU hardware virtualization (on page 151)
4. Elevated PowerShell (on page 152)

8.2.4.8.1 Test environment vs. productive environment

The test environment described in this guide is easier to set up than a typical productive environment.

The fundamental differences are:

	Test environment	Productive environment
Installation option	IIoT Services (Docker on Windows)	▸ IIoT Services (Windows native) ▸ IIoT Services (Docker on Windows)
Number of computers	A computer for IIoT Services and all clients	▸ A computer for IIoT Services ▸ Dedicated computers for clients
Network topology	All applications run on the same computer.	The applications run on different computers. The computers are usually distributed across different remote sites.
Multi-user system	Not suitable as a multi-user	Suitable as a multi-user system.

	Test environment	Productive environment
	system.	
Passwords	It is possible to use predefined passwords in a protected test environment.	For all logins, it is essential that you assign your own secure passwords.

8.2.4.8.2 Installation options for IIoT Services

Here you can find an overview of the different types of installation of IIoT Services, as well as the instructions that you can use for the installation.

Type of installation	Instruction
Windows on-premises installation	Getting started guide - Windows (on page 107)
Installation of Docker on Windows	Follow the instructions in this guide (on page 131)
Installation of Docker on Linux	Follow the instructions in this guide (on page 132)
Docker in the cloud	Follow the basic steps in this guide and adapt the configuration files for your preferred cloud platform.
Kubernetes	Follow the basic steps in this guide and adapt the configuration files for Kubernetes.

8.2.4.8.3 Determine FQDN (Fully Qualified Domain Name)

To determine the **FQDN** of the Windows computer:

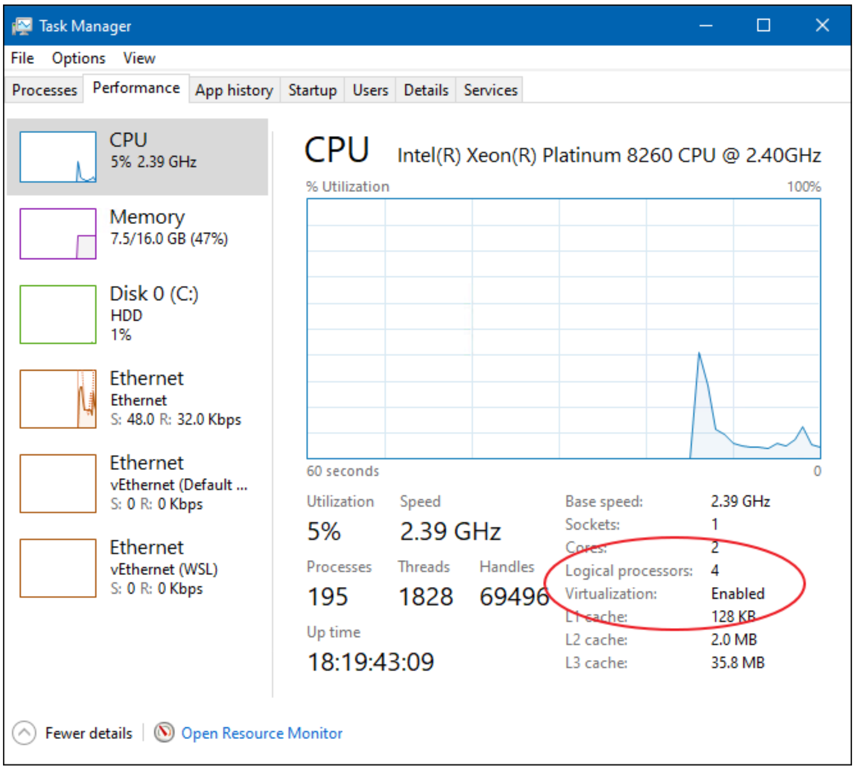
1. Open the command line using the **Windows + R** keyboard shortcut.
2. Enter **cmd.exe**.
3. Enter the **ping localhost** command.
4. The **Command Line Interface** shows your computer's FQDN, among other things.
5. Convert the FQDN to lower-case letters.

You have now determined the FQDN that you need for use in the IIoT Services.

8.2.4.8.4 Check CPU hardware virtualization

To check whether the CPU hardware virtualization has been activated:

1. Open the **Task-Manager**.
2. Click on the **Performance** tab.
3. Go to the **CPU** category there.
4. If your system is correctly configured, you will find the **Virtualization: Enabled** entry under the CPU graph.



CPU
5% 2.39 GHz

Memory
7.5/16.0 GB (47%)

Disk 0 (C:)
HDD
1%

Ethernet
Ethernet
S: 48.0 R: 32.0 Kbps

Ethernet
vEthernet (Default ...
S: 0 R: 0 Kbps

Ethernet
vEthernet (WSL)
S: 0 R: 0 Kbps

CPU Intel(R) Xeon(R) Platinum 8260 CPU @ 2.40GHz

Utilization 5%
Speed 2.39 GHz
Processes 195
Threads 1828
Handles 69496
Up time 18:19:43:09

Base speed:	2.39 GHz
Sockets:	1
Cores:	2
Logical processors:	4
Virtualization:	Enabled
L1 cache:	128 KB
L2 cache:	2.0 MB
L3 cache:	35.8 MB

8.2.4.8.5 Elevated PowerShell

An *elevated PowerShell* is a PowerShell with administrator rights. You can use it to subsequently initialize and administer IIoT Services.

To create an *elevated PowerShell*:

1. Make sure that you have Windows adminstrator privileges on the test computer.
2. Create this link to your desktop:*%SystemRoot%\system32\WindowsPowerShell\v1.0\powershell.exe*
3. Right-click on the link to open the context menu.
4. Select the **Run as Administrator** option.

5. The *elevated PowerShell* is started.

9 Smart Server & Smart Client

The necessary components for the installation are included in the **platform setup** (on page 23).

1. Start the setup.
2. Follow the procedure until you reach the **Components** (on page 26) dialog.
3. In this dialog, select the desired subcomponents for the **Smart Client** component.
4. Click on **Customize** to define the subcomponents in detail.

   The dialog for selecting the components to be installed is displayed.
5. Select the appropriate subcomponents for the installation.
6. Click on **Install** to start the installation process.

Detailed information on the configurations required to operate the **Smart Server** and the **Smart Client** can be found in the **Commissioning Smart Server and Smart Client** section.

10 Web Visualization Service

The required components for operating the Web Visualization Service are included in the platform setup.

- You can find general information on WVS in the **Web Visualization Service** node.
- You can find detailed information for the commissioning and configuration necessary for this in the **Infrastructure** node in the **Web Visualization Service (WVS)** section.

11 HTML Web Engine

You can find information on putting the **HTML Web Engine** into operation in the **Infrastructure** node in the **HTML Web Engine** section.

12 Service Engine for Linux - installation and putting into operation

Generally:

- The installation is run as a **Superuser** on the Linux computer.
- User and user groups: *zenon* (will be created automatically during installation).
- Pay attention to whether you are installing Service Engine on the Linux computer for the first time. If you already have a running installation. Create a backup of the *zenon6.ini* file of the Linux computer if you have already installed Service Engine.
- Folder for zenon project data:
  /etc/copa-data/
  This folder must contain the following subfolders and file.
  - **RT** subfolder with the Service Engine files
  - **ServiceGrid** subfolder with the connection details (*Certificate Bundles*) and relevant certificates.
  - zenon6.ini
    Note: You can find further information in the **Configuration of the zenon6.ini file** (on page 162) section.

LOGIC SERVICE

The files required for Logic Service are installed with the zenon Software Platform.
You can find these files on the Windows computer in the following folder:
%ProgramFiles(x86)%\COPA-DATA\zenon Software Platform 1400\Linux\

12.1 Installation on the Linux computer

The following steps are necessary for commissioning IIoT Services on a Linux computer. You can find a detailed description in the linked sections

1. Set up APT repository (on page 155)
2. Install Service Engine Linux packages on a Linux computer (on page 156)
3. License installation (on page 156)
4. Register and start service

OPTIONAL INSTALLATION

Optionally, install the following **IIoT Services** on your Linux computer:

1. **Device Management**
   Transports zenon projects automatically to Linux computers.
2. IIoT API & IIoT Services Gateway (on page 101)
   Offers additional functionalities of the supported services via API, e.g. alarm acknowledgment.

> **Information**
>
> You can find further Linux sources for components of the zenon software platform on **github** (**https://github.com/COPA-DATA/service-engine-linux-how-to**).

12.1.1 Setting up the APT repository

Carry out the following steps to set up the APT repository on your Linux computer:

1. Log in to the Linux computer.
   - Log in directly to the Linux computer with a physical keyboard, mouse and screen.
   - Connect to your Linux computer via ssh with a remote session.
2. Download the repository GPG key.
   To do this, execute the following command:

   ```
   wget -O- https://repository.copadata.com/zenon/1400/release/copadata-archive-keyring.gpg.key | gpg --dearmor | sudo tee /usr/share/keyrings/copadata-archive-keyring.gpg > /dev/null
   ```

3. Add the APT repository to the list of remote repositories. Make sure that you are using the correct hardware for your Linux computer:
   - For computers with an amd64 processor, execute the following command:

     ```
     echo "deb [arch=amd64 signed-by=/usr/share/keyrings/copadata-archive-keyring.gpg] https://repository.copadata.com/zenon/1400/release/ jammy main" | sudo tee /etc/apt/sources.list.d/copa-data.list
     ```

   - For computers with an arm64 processor, execute the following command:

     ```
     echo "deb [arch=arm64 signed-by=/usr/share/keyrings/copadata-archive-keyring.gpg] https://repository.copadata.com/zenon/1400/release/ bullseye main" | sudo tee /etc/apt/sources.list.d/copa-data.list
     ```

   Note: If the **arch=** is not configured, a warning is displayed during the update of the available packages.
4. Update the local index file for the repository.
   To do this, execute the following command:

   ```
   sudo apt update
   ```

12.1.2 Installation of the Service Engine package

The installation is carried out via the command line interface and applies to both amd64 or arm64 architectures.

Perform the following step to install Service Engine on your Linux computer:

sudo apt install service-engine

This command installs Service Engine and all associated software packages on your Linux computer.

Hint

Make sure that the system also installs the *locales-all* and *iiot-cli-14-0* packages, as these packages are required for the proper functioning of Service Engine. These packages are installed by default together with the *service-engine* package.

12.1.3 Licensing

A valid license is required for the operation of Service Engine on your Linux computer.

⚠Attention

If you start Service Engine on your Linux computer without a valid license, the Service Engine start will fail.

LICENSING VIA COMMAND LINE - AUTOMATION

As an alternative to the configuration of the .INI file, you can also carry out the licensing with the command line tool **LicenseManagerAutomation**. Note that .NET must be installed on the Linux computer (on page 158) for the execution of this tool.

This tool is called up with **LicenseManagerAutomation**

Information

You can find more detailed information in relation to this in the **Licensing via command line - automation** section.

LICENSING VIA MANUAL .INI ENTRY

Perform the following steps to license Service Engine on your Linux computer:

1. Make sure that your Linux computer has a functioning network connection to your license server.
2. Open the */etc/copa-data/License.ini* file, for example, with the **nano** program or another text editor.
   Execute the following command to open the *License.ini* file with **nano**:
   sudo nano /etc/copa-data/License.ini
3. Add the necessary licensing configurations to the License.ini file.
   Please note: Inputs for Linux are case sensitive
   - Service Engine-related entries:
     [Runtime]
     SERIAL0 = Serial number of the license according to the license certificate
     SERIAL0_LOCATION = Computer name of the licensing server under which the licensing server can be reached in the network.
   - Logic Service-related entries:
     [LogicRuntime]
     SERIAL0 = Serial number of the license according to the license certificate
     SERIAL0_LOCATION = Computer name of the licensing server under which the licensing server can be reached in the network.
     SERIAL0_LOCATION = <machine name of the license server>
   - Process Gateway-related entries:
     [ProcessGateway]
     SERIAL0 = Serial number of the license according to the license certificate
     SERIAL0_LOCATION = Computer name of the licensing server under which the licensing server can be reached in the network.

> **Information**
>
> You can find further information in relation to this in the **License.ini** section.

12.1.4 Checking the state of the installation and licensing

The state of Service Engine and the Device Agent as well as the status of the licensing can be checked with the following command:

sudo systemctl status serviceEngine.service

12.2 Installation of Linux components of the zenon Software Platform - Summary

Perform the following steps to install all Linux-based components of the zenon Software Platform on your Linux computer:

1. Copy the appropriate package to your client.
   You can request the necessary packages for Service Engine from your sales partner.
2. Install the **IIoT Services Gateway**
   service-grid-gateway-.deb*
   To do this, execute the following command:
   sudo apt install [package name]

   Note: This installation is mandatory if you want to connect the Service Engine with the IIoT Services.
3. Install
   service-engine-.deb*
   Start the service.
4. Optional: Install the diagnosis Server
   zenon-logging-server
   Check your installation using the command *sudo systemctl status zenLogSrv.service*
5. Optional:
   Delete the installation packages. To do this, enter the following command:
   sudo apt autoremove
6. Optional:
   Register Service Engine as a service. To do this, enter the following commands:
   - *sudo systemctl enable serviceEngine*
   - *sudo systemctl start serviceEngine*

   Note: If there are no zenon projects installed on the Linux computer, the following error message will be displayed when starting:
   ServiceEngine initialization terminated with error - -1 / load of project failed
7. Optional:
   Check your installation. To do this, enter the command:
   sudo systemctl status serviceEngine

12.3 .NET installation on Linux

Install the .NET runtime environment using the platform package administration if possible. On a platform on which the .NET runtime environment cannot be installed by means of package administration, it may be possible to install .NET with an installation script.

1. Set up APT repository (on page 155)
2. Install .NET Linux packages on the Linux computer (on page 156)

> **Information**
>
> You can find detailed information on .NET for Linux on the **Microsoft homepage** (**https://learn.microsoft.com/en-us/dotnet/core/install/linux-ubuntu#register-the-microsoft-package-repository**).

12.3.1 Set up APT repository for .NET

Before installing .NET, carry out the following commands in order to add the Microsoft package signing key to the list of trusted keys and the package repository.

```
# Get Ubuntu version

declare repo_version=$(if command -v lsb_release &> /dev/null; then lsb_release -r -s; else grep -oP '(?<=^VERSION_ID=).+' /etc/os-release | tr -d '"'; fi)


# Download Microsoft signing key and repository

wget https://packages.microsoft.com/config/ubuntu/$repo_version/packages-microsoft-prod.deb -O packages-microsoft-prod.deb


# Install Microsoft signing key and repository

sudo dpkg -i packages-microsoft-prod.deb


# Clean up

rm packages-microsoft-prod.deb


# Update packages

sudo apt update
```

⚠Attention

We recommend that you only use one repository to administer all your .NET installations. If you have previously installed .NET with the Ubuntu repository, you must clear the system of .NET packages and configure the APT so that the Ubuntu feed is ignored.

12.3.2 Installation of .NET 8

Execute the following command to install Service Engine on your Linux computer:

sudo apt install [package name]

Replace *[package name]* with the name of the .NET package that you would like to install. To install .NET SDK 8.0, for example, use the command *sudo apt install aspnetcore-runtime-8.0*.

.NET version	Product	Type	Package name
8	**ASP.NET Core**	**Runtime**	*aspnetcore-runtime-8.0*
8	.NET	**Runtime**	*dotnet-runtime-8.0*
8	.NET	**SDK**	*dotnet-sdk-8.0*

12.3.3 Check for previously-installed .NET versions

You can use a Terminal to check which versions of .NET runtime are currently installed. To do this, open a Terminal and execute the following command:

dotnet *--list-runtimes*

The result is an output similar to the following example.

```
Microsoft.AspNetCore.App 5.0.0 [/usr/lib/dotnet/shared/Microsoft.AspNetCore.App]

Microsoft.AspNetCore.App 6.0.0 [/usr/lib/dotnet/shared/Microsoft.AspNetCore.App]

Microsoft.NETCore.App 5.0.0 [/usr/lib/dotnet/shared/Microsoft.NETCore.App]

Microsoft.NETCore.App 6.0.0 [/usr/lib/dotnet/shared/Microsoft.NETCore.App]

Microsoft.NETCore.App 7.0.0 [/usr/lib/dotnet/shared/Microsoft.NETCore.App]

Microsoft.NETCore.App 8.0.0 [/usr/lib/dotnet/shared/Microsoft.NETCore.App]
```

12.3.4.NET environment variable

When starting in Service Engine using Linux, a check is carried out to see whether the **DOTNET_ROOT** environment variable has been set. If this is not configured, the following two paths are used:

- */usr/lib/dotnet/shared/Microsoft.NETCore.App*
- */usr/lib/dotnet/shared/Microsoft.AspNetCore.App*

If .NET has been installed using an install script, the environment variable must be set accordingly. In the environment variable, the save location of the .NET installation must be declared.

Additional information:

- **Install .NET on Linux by using an install script or by extracting binaries** (**https://learn.microsoft.com/en-gb/dotnet/core/install/linux-scripted-manual**)
- **Deploy .NET apps on ARM single-board computers** (**https://learn.microsoft.com/en-us/dotnet/iot/deployment**)

12.3.5 Uninstallation

If you have installed .NET via a package manager, uninstall it in the same way with the following command:

sudo apt-get remove *[package name]*

Example: sudo apt-get remove aspnetcore-runtime-8.0.

12.4 Commissioning of a zenon project on Linux computers

The following options are available for commissioning a zenon project for Service Engine on Linux:

- Manual installation
- Installation via Device Management of IIoT Services

12.4.1 Manual installation

Carry out the following steps to install a zenon project manually on a Linux computer:

1. Copy the Service Engine data from a Windows computer (on page 162).
2. Adapt the zenon6.ini file on the Linux computer (on page 162).

12.4.1.1 Copy zenon project data to a Linux computer

Copy the zenon project data to your Linux computer.

1. To do this, generate all Service Engine files in Engineering Studio on the engineering computer.
2. On the engineering computer, go to the saving location of the files you generated. This corresponds to configuring the **Service Engine folder** property in Engineering Studio.
3. Copy all the data of the folder to a folder on the Linux computer.
   Default: */etc/copa-data/RT*
4. Check the permissions for the folder. To do this, enter the following command:
   sudo chown zenon:zenon -R /etc/copa-data/

> ⚠**Attention**
>
> If the project contains content that is not supported on the Linux operating system, the project will not start correctly. You can find an overview in the **Scope of functions** node.

12.4.1.2 Configuration of the zenon6.ini file

Customize the *zenon6.ini* to your Linux system.

1. Go to the saving location of the *zenon6.ini* file. To do this, enter the following command:
   cd /etc/copa-data
2. Correct the permissions for the *zenon6.ini* file. To do this, enter the following command:
   sudo chown zenon zenon6.ini
3. Show the current *zenon6.ini*. To do this, enter the following command:
   cat /etc/copa-data/zenon6.ini
4. Configure the following entries with an editor of your choice:
   - **[Default]**
     Defanwendung30= zenon project name on the Linux computer.
   - **[PATH]**
     VBF30= Path of the current project database. This path corresponds to the folder with the copied Service Engine files.
   - Optional: Customize the parameters for the Diagnosis Service. *localhost* is given by default.
     [LOGGING_SYSTEM]
     Config= Parameter for connection with the respective diagnosis server.

> **Attention**
>
> When a package is installed, you will be asked if the *zenon6.ini* file should be overwritten by a default file. If this file is overwritten, any customizations made so far to this file must be configured again.

12.4.2 Installation via Device Management

Carry out the following steps to install a project via Device Management on your Linux computer:

1. Install IIoT Services certificate
2. Installing Device Agent
3. Prepare the project in Engineering Studio for Device Management.

12.4.2.1 Installing CA certificate of <NAME_SERIVCE_GRID>

The following steps are only important if you are using the standard HTTPS certificate or another self-signed HTTPS certificate for communication with IIoT Services, e.g.**Device Management**. As the default IIoT services HTTPS certificate is not trusted, you must add the corresponding CA certificate to the list of trusted certificates.

If you already have a running installation of IIoT Services on the Linux computer and this IIoT Services already uses a trusted certificate, it is not necessary to install the CA certificate again.

> **Hint**
>
> As an alternative to manual installation of a certificate, certificates can also be generated directly on the Linux client via the **IIoT Services CLI**.

Carry out the following steps to install the CA certificate of IIoT Services on your Linux computer:

1. Open **Service Configuration Studio** and download the CA certificate on your local computer.
   Note: If your Linux computer has a graphical user interface and a browser, you can download Service Configuration Studio directly on the Linux computer.
2. Transfer the CA certificate to the Linux computer using a suitable mechanism such as scp, ftp or network file shares.
   Note: You can find further information on scp on the ubuntu manual website (https://manpages.ubuntu.com/manpages/trusty/man1/scp.1.html).
3. Copy the CA certificate file on the Linux computer to the */usr/local/share/ca-certificates* directory. The name of the certificate file is of no importance.

4. Update the system's trusted CA certificates with the following command:
   sudo update-ca-certificates
   This updates the CA certificates of the Linux computer and adds the certificate for IIoT Services to the list of trusted certificates.
5. Check the installed certificate. To do this, execute the following command:
   curl 'https://[url-of-iiot-services]:[port-of-iiot-services]
   Example:

> **Information**
>
> You can find detailed information on the certificate handling of IIoT Services in the **Certificate Management** and **Certificates** sections.

12.4.2.2 Installing Device Agent

The **Device Agent** is installed together with the IIoT command line interface.

> **Attention**
>
> **Device Management** is not available for Service Engine in Docker environments.

You can find detailed instructions for installation on Linux systems in the **Service Engine on Linux** section.

Carry out the following steps to install the service for **Device Management** (= **Device Agent**) on a Linux device:

1. Update the list of the zenon software packages available. Execute the following command:
   sudo apt update
2. Install the command line interface on the Linux device. Execute the following command:
   sudo apt install iiot-cli-14-0
3. Carry out the installation of the Device Agent service. To do this, enter the following command:
   a) If you have web access for authentication on the Linux device:
   iiot-cli setup-agent -u [URL to the IIoT Services]:Port -n [Name of the device]
   The name of the device is optional. If the command is entered without a device name, the pre-configured CLI client is used for the connection (= *DeviceManagementAgentCliClientId*).
   Example: *iiot-cli setup-agent -u https://iiot-docu-v8.testenv.local:9443 -n LinuxDevice*

   b) If you do not have web access for authentication on the Linux device:
   iiot-cli setup-agentsetup-agent -u https://iiot-docu-v8.testenv.local:9443 -n LinuxDevice --use-device-code

The **--use-device-code** command tag is used for authorization via the Identity Service in Service Configuration Studio.
The URL and code for this authorization is visualized in the command line.
After successful authorization, installation on the Linux device will be complete.

4. If you have installed more than one version of IIoT-CLI , you can switch between the versions. To do this, execute the following command:

   sudo update-alternatives --config iiot-cli

   Then follow the instructions of CLI.

5. Check the status of the device agent:

   - Check the connection status of the device in the user interface of **Device Management** device administration in Service Configuration Studio. The Linux device must display the status Online.

   - On the Linux computer, the status of the device agent is checked with the following command:
     sudo systemctl status device-agent.service

Information

Ensure that the following language settings (**Locales**) are installed on your system:

- *en_US.UTF-8*

- *UTF-8/en_US.UTF-8*

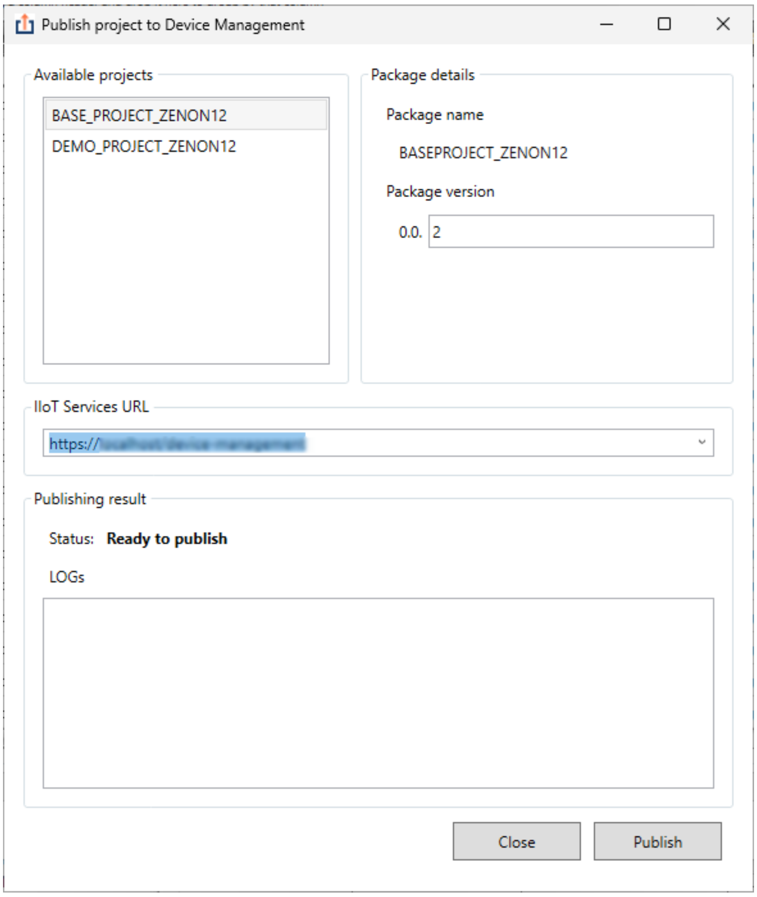
- *UTF-8*

If these **Locales** are missing, it can happen that the Device Agent is closed with the following error message:

terminate called without an active exception Aborted

12.4.2.3 Preparing zenon projects in Engineering Studio for Device Management

The deployment of packages is implemented in Engineering Studio with a wizard.



Publish project to Device Management

Available projects

BASE_PROJECT_ZENON12

DEMO_PROJECT_ZENON12

Package details

Package name: BASEPROJECT_ZENON12

Package version: 0.0. 2

IIoT Services URL: https://[redacted]

Publishing result

Status: **Ready to publish**

LOGs

You can find detailed information for the transfer of a project from Engineering Studio in the **Device Management** node in the **IIoT Services - configuration in Engineering Studio** node.

12.5 Starting Service Engine

To register manually, enter the following command: *sudo systemctl enable serviceEngine*

Start Service Engine on your Linux client with the following command: *sudo systemctl start serviceEngine*

RESTARTING

Enter the following command to restart a running Service Engine on your Linux client: *sudo systemctl restart <service-name>*

Example: *sudo systemctl start serviceEngine*

12.6 Uninstallation

Perform the following steps to uninstall the zenon installation on your Linux computer.

1. Go to the appropriate folder. To do this, enter the following command:
   sudo apt remove [package name]
   Example: *sudo apt remove service-engine*
   This deletes all the content installed with the package.
2. Repeat this step for all installed packages.
3. The *zenon6.ini* file will not be deleted. Enter the following command to also delete the configuration file:
   sudo apt purge [package name]

13 FAQ

Errors during the installation mostly occur when the replacement or creation of files is prevented by a virus scanner or by existing installations. Here you can find the most frequent reasons for installation errors and their solution.

ZENON

Problem	Solution
Installation is terminated. Typical error message: **Error 1304. Error writing to file...**	Deactivate the virus scanner. Close unnecessary programs.
Demo projects were installed but are not displayed. New projects cannot be created.	Check the computer name. The computer: ▸ must not consist of more than 15 characters ▸ must be in accordance with the convention of the NetBIOS computer name
Error message that a service cannot be started.	▸ first reboot the computer ▸ then start the zenon setup again

SQL SERVER

GENERAL

Problem	Solution
The installation is unsuccessful because the password does not meet the requirements.	The installation of SQL Server is not possible if the security requirements do not permit a password length of 20 characters (A-Z, a-z, 0-9 and special characters). Each character can only be used once.

14 Technical support

BASIC SUPPORT

If you need support for the installation, our employees in Technical Consulting would be happy to help you.

User with basic support can reach the hotline at the following e-mail address: support@copadata.com.

ADVANCED AND PREMIUM SUPPORT

If you have an Advanced or Premium service agreement, please use the telephone number or email address provided in that. Our sales employees (sales@copadata.com) will gladly assist you, if you want to upgrade your free basic service agreement to an Advanced or Premium service agreement.