zenon manual

HTML Web Engine

v.8.20

Contents

1 Welcome to COPA-DATA help

ZENON VIDEO TUTORIALS

You can find practical examples for project configuration with zenon in our YouTube channel (https://www.copadata.com/tutorial_menu). The tutorials are grouped according to topics and give an initial insight into working with different zenon modules. All tutorials are available in English.

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.

PROJECT SUPPORT

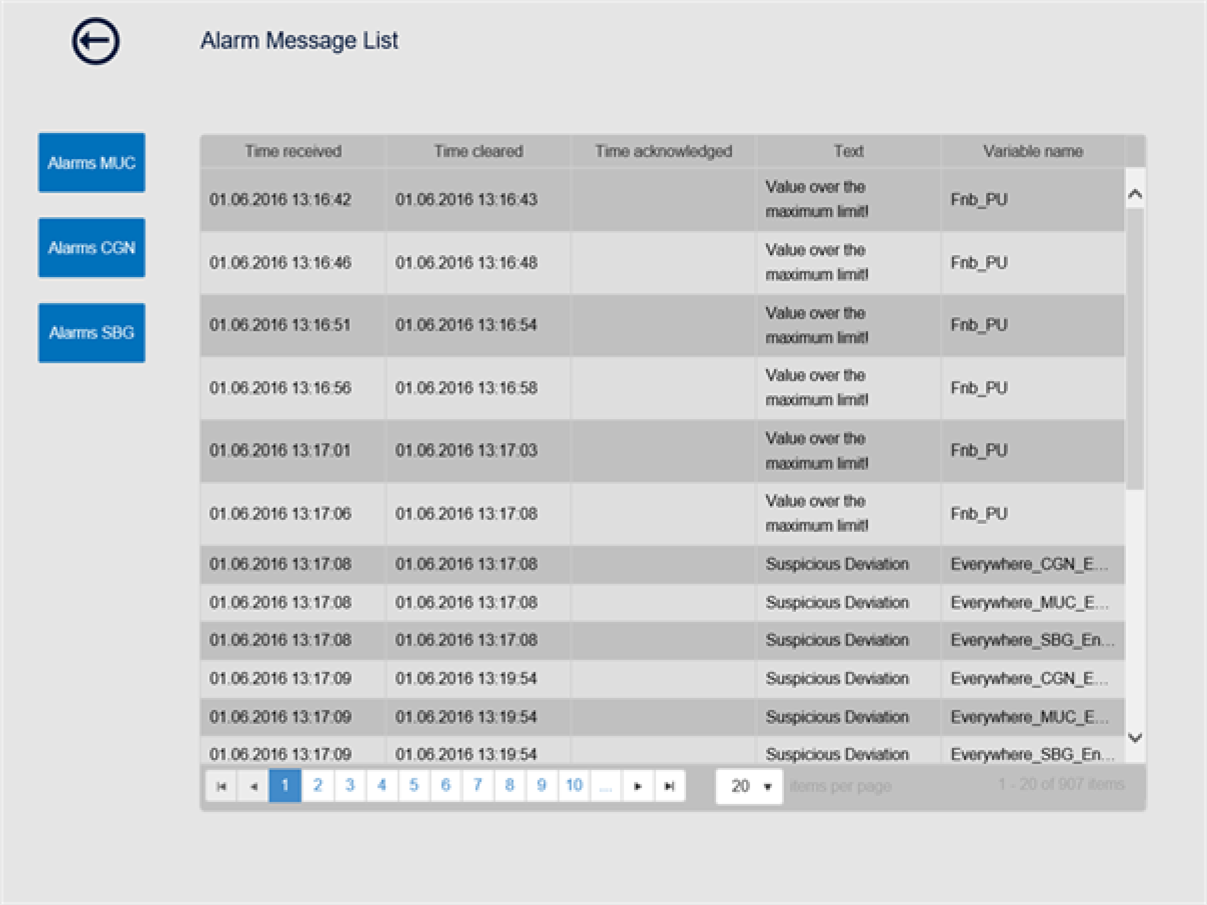
You can receive support for any real project you may have from our customer service team, which you can contact via email at support@copadata.com.

LICENSES AND MODULES

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

2 HTML Web Engine

The **HTML Web Engine** is for the provision of zenon screens as a HTML5 web page. The user interface is called up and displayed on the visualization end device using a web browser. No special software installation - or any browser plug-ins - are required on the end device. Process data for the visualization is taken from zenon Runtime.



Time received	Time cleared	Time acknowledged	Text	Variable name
01.06.2016 13:16:42	01.06.2016 13:16:43		Value over the maximum limit!	Fnb_PU
01.06.2016 13:16:46	01.06.2016 13:16:48		Value over the maximum limit!	Fnb_PU
01.06.2016 13:16:51	01.06.2016 13:16:54		Value over the maximum limit!	Fnb_PU
01.06.2016 13:16:56	01.06.2016 13:16:58		Value over the maximum limit!	Fnb_PU
01.06.2016 13:17:01	01.06.2016 13:17:03		Value over the maximum limit!	Fnb_PU
01.06.2016 13:17:06	01.06.2016 13:17:08		Value over the maximum limit!	Fnb_PU
01.06.2016 13:17:08	01.06.2016 13:17:08		Suspicious Deviation	Everywhere_CGN_E...
01.06.2016 13:17:08	01.06.2016 13:17:08		Suspicious Deviation	Everywhere_MUC_E...
01.06.2016 13:17:08	01.06.2016 13:17:08		Suspicious Deviation	Everywhere_SBG_En...
01.06.2016 13:17:09	01.06.2016 13:19:54		Suspicious Deviation	Everywhere_CGN_E...
01.06.2016 13:17:09	01.06.2016 13:19:54		Suspicious Deviation	Everywhere_MUC_E...
01.06.2016 13:17:09	01.06.2016 13:19:54		Suspicious Deviation	Everywhere_SBG_En...

FUNCTIONALITIES OF THE HTML WEB ENGINE

Overview of the functions of the HTML web engine:

- Session-based provision of HTML5 visualization content on HTML web clients.
- Display of basic visualization content that was created in the zenon Editor.
- Data view:
  - Variable values
    Displays are updated dynamically in the HTML Web Engine.
  - Chronological Event List (CEL)
    Entries in the CEL are updated dynamically in the HTML Web Engine.

  - Extended Trend (ETM)
    The display in the ETM is not updated dynamically in the HTML Web Engine. This means that the ETM only draws with existing data when called up and is then no longer updated.
  - Messages from the Alarm Message List (AML)
    Entries in the AML are updated dynamically in the HTML Web Engine.
- Forwarding of process information, such as variable values, alarm messages or event messages from a zenon Runtime to one or more HTML web clients.
- Support of active operations, such as write set value.
- Mobile, location-independent operation and observation.
- No installation and/or configuration on the end device, i.e. the client is necessary. Platform-independent display in HTML5 standard.
- Operation of the HTML web server on a different computer, such as is possible in a DMZ for example.
- Secure network communication via HTTPS, based on SSL certificates.
- Protection of sensitive visualization areas or processes by means of user authentication and support of user levels.
- Access information from the **Service Grid**.

The HTML Web Engine supports the authentication of a Web Engine client with increased security in relation to the zenon user authentication and active directory. Login is via entry of the user name and password.

REGIONAL SETTINGS

The HTML Web Engine supports most common languages, settings, number formats and date formats.

3 Required components and their definitions

Components	Description
zenon Runtime	The process data for the HTML5 visualization is provided by a zenon Runtime (server or client).
HTML Web Engine (on page 6)	The HTML Web Engine is for the provision of process screens as an HTML5 web page. The user interface is called up and displayed on the visualization end device using a web browser. Neither special software installation nor a software plug-in are required on the

Components	Description
	end device. Process data for the visualization is taken from zenon Runtime. **Note:** The HTML Web Engine processes process data for the purpose of visualization and operation by the HTML web client. The process data is only administered by zenon Runtime.
IIS (on page 15) **Publishing Service**	Services platform of Microsoft for PCs and servers. It can be used to make documents and files accessible in the network. The HTML Web Engine uses IIS as a runtime environment and for the publishing of zenon process screens. HTTPS is used as a communication protocol. **Deployment** (on page 26) is used to instance the HTML Web Engine on the IIS.
Web browser	Web browsers are special computer programs for the display of documents and data, especially web sites in the World Wide Web.
zenon Web Engine Deployment Tool (on page 26)	Provides the HTML Web Engine as a web application in IIS and allows the configuration thereof. An existing HTML Web Engine instance can also be updated or deleted.
Web engine compiler	Generates, from a zenon project, the data that the HTML Web Engine needs to provide HTML5 content for the zenon Web Client. When translating this project data, the HTML Web Engine compiler checks the project contents and provides information on non-supported functions or properties. As a result of the translation process, a file is created that is provided to the Web Engine.
SCADA Runtime Connector (on page 22)	Serves as a communication interface to zenon Runtime. You can find the **SCADA Runtime Connector** in the following path: *%programfiles(x86)%\Common Files\COPA-DATA\Connectors\zrsConnector.exe* If the web deployment tool is used, the **SCADA Runtime connector** must be executed, because its status is checked when used.

Components	Description
	The Connector Container can be started automatically using the **Startup Tool** if a user logs on to the system.
Service Grid Connector (on page 24)	The **Service Grid Connector** enables the connection of the HTML Web Engine to the Service Grid infrastructure. The HTML Web Engine can thus run outside of a SCADA network.

4 Basic system construction of the HTML Web Engine

The HTML Web Engine is a web application that provides an HTML5 web page.

⚠Attention

Recommendations:

- Always operate the whole system configuration in a trusted network area.
- Never publish the HTML5 web site in the Internet directly.

In the course of a session, a distinction between two different connection levels is made:

1. Display of visualization pages without process data
2. Display of the visualization pages and display of process data

DISPLAY OF VISUALIZATION PAGES WITHOUT PROCESS DATA

The zenon Web Client connects itself to the HTML Web Engine by calling up the URL (Uniform Ressource Locator) for the HTML5 web page. Once the session has been set up successfully, the project can be visualized without access to process data of zenon Runtime.

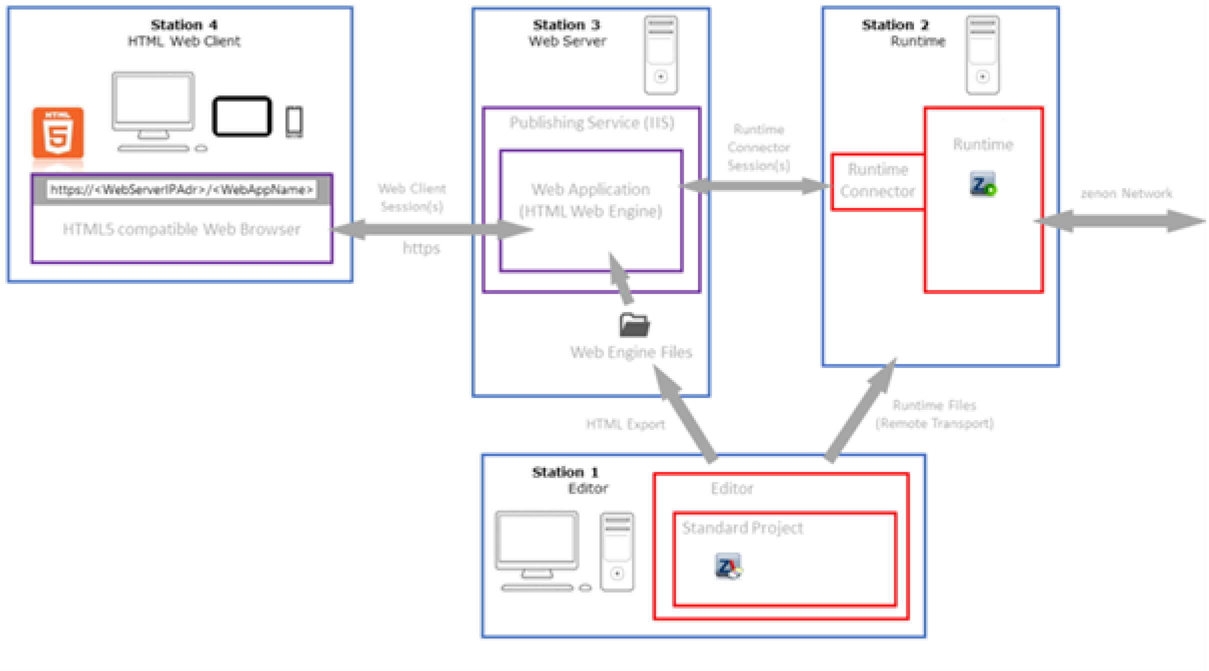
DISPLAY OF THE VISUALIZATION PAGES AND DISPLAY OF PROCESS DATA

The HTML Web Engine connects to zenon Runtime via the **SCADA Runtime Connector** (on page 22) or the **Service Grid Connector** (on page 24). This connection is only approved if user authentication on the basis of a user name and password has been carried out successfully. Authentication is carried out by means of external authentication to the user administration of zenon Runtime. The transfer of user information can be either manual by the zenon Web Client operator or automatic by the Web Engine.

No special tools are required to configure the HTML5 visualization. The screens and functions are created in the zenon Editor by default.
You can find a list of the supported elements, properties and functions in the **Supported functionalities for HTML visualization** (on page 43) chapter.

ACCESS TO HTML5 VISUALIZATION FROM AN END DEVICE

This is how you use an HTML5-compatible web browser to access the HTML5 visualization from a visualization end device:



Note: The breakdown of the components is only for simple display. The complete configuration shown here can be operated in a network or also on any one of the individual computers in any desired distribution of the components.

Procedure:

1. The operator on Station 4 connects with a standard web browser by entering the web page URL to the web server on Station 3.
   As a result, it gets the visualization pages there from the Web Engine.
2. Process data is only displayed in the HTML5 visualization after successful user authorization. After a check of the user name and password has been carried out, the connection to zenon Runtime is established (to Station 2 in the example).
3. The interface between zenon Runtime and Web Engine is formed by the **SCADA Runtime Connector** or the **Service Grid Connector**.
4. The configuration of the HTML5 visualization is derived from a zenon Editor project (on Station 1 in this case).

The project states on Station 2 (zenon Runtime) and Station 3 (Web Engine) should be identical for this.

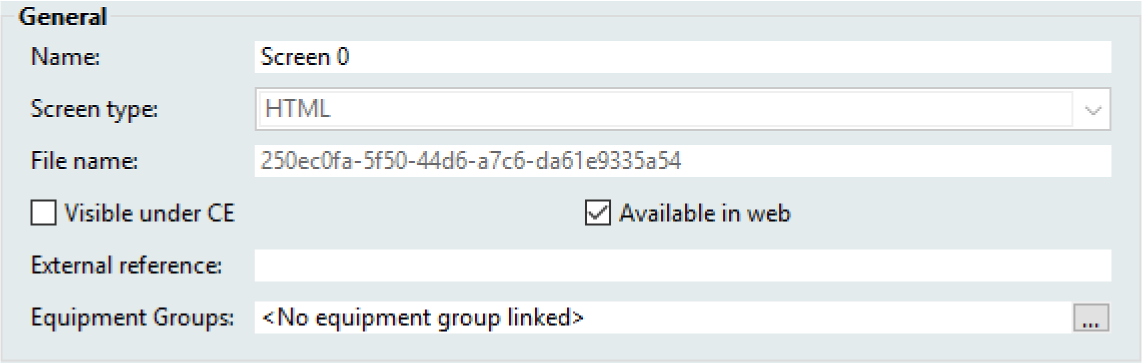
With the **Web Engine Compiler**, the project data is translated for use by the HTML Web Engine. The necessary steps for the installation of the HTML Web Engine on Station 3 and the deployment on the Internet Information Server are described in the Deployment of the web engine (on page 26) chapter.

The file from the **Web Engine Compiler** is saved in a freely-definable folder. It is read by the HTML Web Engine from here.

Notes: Provide, for the connection of HTML5 visualization, a dedicated zenon Runtime, in order to guarantee smooth interaction for process-related procedures. After changes are made in the zenon project, carry out the compiling process again.

RELEASE SCREENS

Certain screens from the active project can be called up in a web browser by means of the HTML Web Engine. You determine the screens that are unlocked for display in the web browser for each screen using the **Available in web** property.



General

Name: Screen 0

Screen type: HTML

File name: 250ec0fa-5f50-44d6-a7c6-da61e9335a54

☐ Visible under CE ☑ Available in web

External reference:

Equipment Groups: <No equipment group linked> ...

All screens for which this property has been activated are compiled with the **Web Engine Compiler** for the Web Engine and are primarily available for provision in the web browser on the client. Screens that have not been activated for this property can be used for the zenon Runtime visualization, but are not available in the web browser.

The following screen types can be used for visualization in the web:

- *AML*
- *CEL*
- *ETM*
- *Login*
- *HTML*
- *Standard*

5 System requirements

WEB SERVER

The HTML Web Engine supports the following operating systems:

Supported desktop operating systems and required service packs:

Operating system	Service packs
Windows 7 (Professional, Enterprise and Ultimate version, x86 and x64 versions).	SP 1
Windows Embedded Standard 7 (if all necessary operating system components exist).	SP 1
Windows 8 and 8.1 (Standard, Professional, Enterprise version, x86 and x64 versions)	SP 0
Windows Embedded 8 Standard (if all necessary operating system components exist).	SP 0
Windows 10 (Home, Pro, Enterprise, Education, Pro Education, Enterprise LTSB, Enterprise LTSC, IoT Enterprise, Pro for Workstations)	SP 0

Supported server operating systems and required service packs:

Server operating system	Service packs
Windows Server 2008 R2 (All editions with the exception of Core)	SP 1
Windows Server 2012 and 2012 R2 (All editions with the exception of Core)	SP 0
Windows Server 2016 (All editions with the exception of Core)	SP 0

Note: For operation of the HTML Web Engine, the .NET Framework 4.5 or higher is required.

Information

Note the limitation of the number of simultaneous client connections by the Microsoft IIS.

Note: When using Windows Server operating systems, the number of simultaneous client connections is not limited by the Microsoft IIS. When using Windows desktop operating systems, this number varies depending on the version of the installed operating system.

WEB CLIENT

No special installation is required for the HTML Web Client. It generally works with any web browser that supports the following technologies:

- HTML5
- HTML5 Canvas
- JavaScript (ECMAScript 5.1)

HTML5 content is processed in the web browser regardless of the operating system. Use of a current version of one of the following web browsers is recommended:

- Windows Internet Explorer from version 11
- Microsoft Edge
- Mozilla Firefox
- Apple Safari
  Exception: Does not support playback of audio files.
- Google Chrome

Note: JavaScript must be activated in the web browser.

CLIENT-SERVER CONNECTIONS

There must be a sufficient data rate available for the connection between web server and HTML client. With a data rate that is too low, corresponding messages are displayed on the HTML web client. A data rate that is too low can lead to a session not taking place or having to be canceled. You can find further information on behavior in the event of a connection failure in the **Reconnect** (on page 25) chapter.

Note: When using Windows Server operating systems, the number of simultaneous client connections is not limited by the Microsoft IIS. When using Windows desktop operating systems, the possible number of simultaneous connections depend on the version the installed operating system.

6 Licensing

The HTML Web Engine must be licensed for each instance. **Licensing** is carried out using the **License Manager**. If there is no license, the HTML Web Engine is started in a time-limited demo mode.

The following expansions are available for the HTML web engine:

- *Standard*: Only read access to the visualization. The **HTML Web Client** can be used as an observer.
- *Pro*: Full access to the visualization, read and write.

> **Information**
>
> Note the limitation of the number of simultaneous client connections by the Microsoft IIS.
>
> **Note:** When using Windows Server operating systems, the number of simultaneous client connections is not limited by the Microsoft IIS. When using Windows desktop operating systems, this number varies depending on the version of the installed operating system.

Differences between the HTML web engine and zenon Web Server:

- The license check is carried out on the basis of instances. Each project corresponds to an instance.
- If a license is invalid, a further license is first searched for. If none is found, a further license can be searched for using the license search button.
- If demo licenses expire, the service must be restarted in order to be able to use the demo mode. The demo mode is only available if a demo license has been saved.

7 Installation

You need the following installations to operate the HTML Web Engine:

Web Server:

- IIS publishing service:

  Set up the publishing service in accordance with the instructions in the Install IIS publishing service (on page 15) chapter.
- zenon Web Server.

  Install zenon Web Server from the installation medium.

Microsoft Web Deploy It also automatically installed during setup.

Project configuration and runtime application:

- zenon (Editor and Runtime).

Note: No special installation is required for the HTML Web Client.

7.1 IIS Publishing service installation

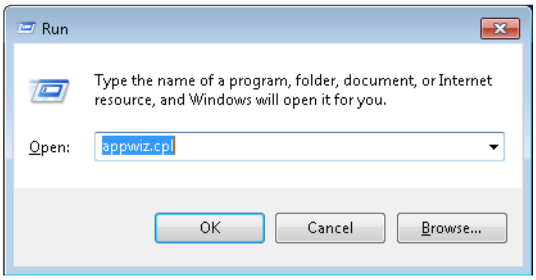
Internet information Services, abbreviated to IIS, are for the publication of documents, such as HTML pages, using the HTTP protocol. For the operating systems listed in the System requirements (on page 12) chapter, the IIS publishing service is already included in the standard installation. This need only be activated by means of the Windows features.

IIS 7, WINDOWS 7

To activate the IIS publishing service:

1. Press the **Windows key + R** keyboard shortcut.

   The dialog to enter a command for the command processing is opened.



Run

Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

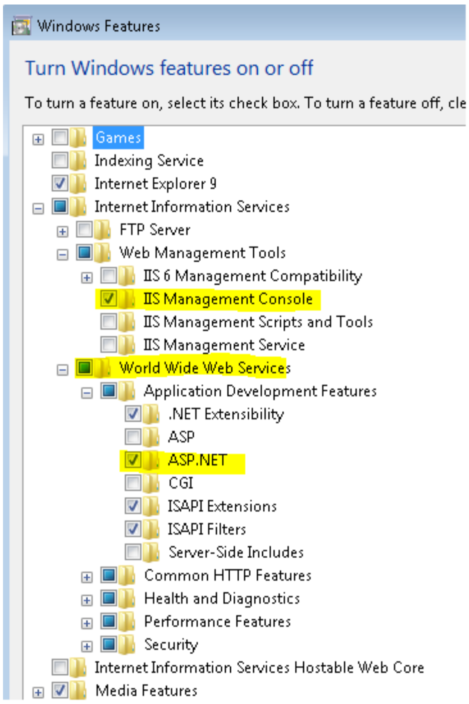
Open: appwiz.cpl

2. Enter *appwiz.cpl* in the input field.

   Click on **OK**.
   A new control panel window to configure programs and **Windows features** is opened.

3. In this window, click on **Turn Windows features on or off**.

The window to select features of the operating system is opened.



Windows Features

Turn Windows features on or off

To turn a feature on, select its check box. To turn a feature off, cle

- ☐ Games
- ☐ Indexing Service
- ☑ Internet Explorer 9
- ▣ Internet Information Services
  - ☐ FTP Server
  - ▣ Web Management Tools
    - ☐ IIS 6 Management Compatibility
    - ☑ IIS Management Console
    - ☐ IIS Management Scripts and Tools
    - ☐ IIS Management Service
  - ▣ World Wide Web Services
    - ▣ Application Development Features
      - ☑ .NET Extensibility
      - ☐ ASP
      - ☑ ASP.NET
      - ☐ CGI
      - ☑ ISAPI Extensions
      - ☑ ISAPI Filters
      - ☐ Server-Side Includes
    - ▣ Common HTTP Features
    - ▣ Health and Diagnostics
    - ▣ Performance Features
    - ▣ Security
- ☐ Internet Information Services Hostable Web Core
- ☑ Media Features

4. Expand the **Internet information services** in this node.
5. Activate *WWW services*.
   The default settings are thus set in all subfolders of the property.
6. Expand the **Application development features** node.
7. Activate the **ASP.NET** option:
8. Expand the **Web administration tools** node.
9. There, activate the **IIS administration console**.
10. Click on **OK**.

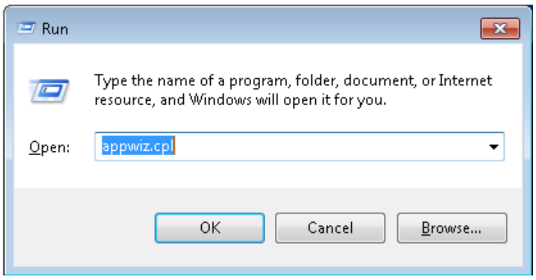
Note: In the event of subsequent installation or upgrading of the NET framework under Windows 7, this software must be registered manually in the IIS publishing service. For more details, see the chapter **.NET registration on IIS for Windows 7** (on page 21).

IIS 8, WINDOWS 8/8.1

To activate the IIS publishing service:

1. Press the **Windows key + R** keyboard shortcut.

The dialog to enter a command for the command processing is opened.



Run

Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

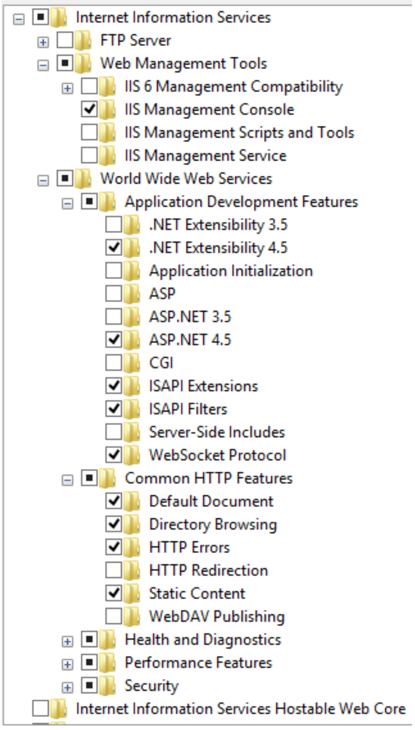
Open: appwiz.cpl

2. Enter *appwiz.cpl* in the input field.

   Click on **OK**.
   A new control panel window to configure programs and **Windows features** is opened.

3. In this window, click on **Turn Windows features on or off**.

   The window to select features of the operating system is opened.



- ■ Internet Information Services
  - □ FTP Server
  - ■ Web Management Tools
    - □ IIS 6 Management Compatibility
    - ☑ IIS Management Console
    - □ IIS Management Scripts and Tools
    - □ IIS Management Service
  - ■ World Wide Web Services
    - ■ Application Development Features
      - □ .NET Extensibility 3.5
      - ☑ .NET Extensibility 4.5
      - □ Application Initialization
      - □ ASP
      - □ ASP.NET 3.5
      - ☑ ASP.NET 4.5
      - □ CGI
      - ☑ ISAPI Extensions
      - ☑ ISAPI Filters
      - □ Server-Side Includes
      - ☑ WebSocket Protocol
    - ■ Common HTTP Features
      - ☑ Default Document
      - ☑ Directory Browsing
      - ☑ HTTP Errors
      - □ HTTP Redirection
      - ☑ Static Content
      - □ WebDAV Publishing
    - ■ Health and Diagnostics
    - ■ Performance Features
    - ■ Security
- □ Internet Information Services Hostable Web Core

4. Expand the **Internet information services** in this node.
5. Activate all *World Wide Web Services* there.
6. Expand the **Application development features** node.
7. Activate *ASP.NET 4.5*
8. Expand the **General HTTP features** node
9. There, activate the *static content*.
10. Expand the **Web administration tools** node.

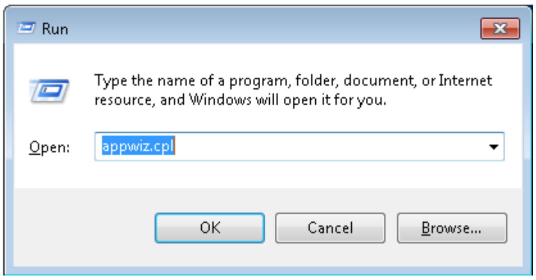
11. Optionally, activate the *IIS administration console* there.
12. Expand the **Application development features** node.
13. Activate the *WebSocket-Protokoll*.
14. Click on **OK**.

WINDOWS 10

To activate the IIS publishing service:

1. Press the **Windows key + R** keyboard shortcut.

   The dialog to enter a command for the command processing is opened.

   Run

   Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

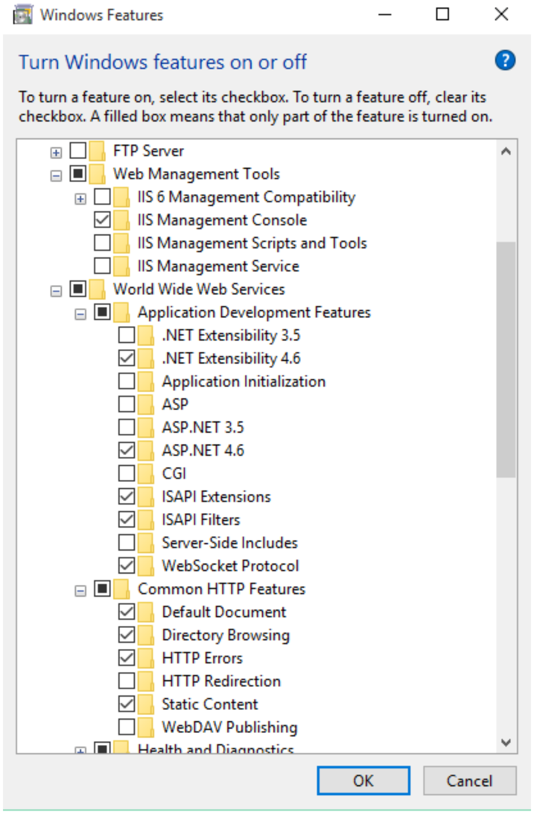
   Open: appwiz.cpl

2. Enter *appwiz.cpl* in the input field.

   Click on **OK**.
   A new control panel window to configure programs and **Windows features** is opened.

3. In this window, click on **Turn Windows features on or off**.

The window to select features of the operating system is opened.



Turn Windows features on or off

To turn a feature on, select its checkbox. To turn a feature off, clear its checkbox. A filled box means that only part of the feature is turned on.

- ⊞ □ FTP Server
- ⊟ ■ Web Management Tools
  - ⊞ □ IIS 6 Management Compatibility
  - ☑ IIS Management Console
  - □ IIS Management Scripts and Tools
  - □ IIS Management Service
- ⊟ ■ World Wide Web Services
  - ⊟ ■ Application Development Features
    - □ .NET Extensibility 3.5
    - ☑ .NET Extensibility 4.6
    - □ Application Initialization
    - □ ASP
    - □ ASP.NET 3.5
    - ☑ ASP.NET 4.6
    - □ CGI
    - ☑ ISAPI Extensions
    - ☑ ISAPI Filters
    - □ Server-Side Includes
    - ☑ WebSocket Protocol
  - ⊟ ■ Common HTTP Features
    - ☑ Default Document
    - ☑ Directory Browsing
    - ☑ HTTP Errors
    - □ HTTP Redirection
    - ☑ Static Content
    - □ WebDAV Publishing
  - ⊞ ■ Health and Diagnostics

4. Expand the **Internet information services** in this node.
5. Activate all *World Wide Web Services* there.
6. Expand the **Application development features** node.
7. Activate *ASP.NET 4.7*
8. Expand the **General HTTP features** node
9. There, activate the *static content*.
10. Expand the **Web administration tools**   node.
11. Optionally, activate the *IIS administration console* there.
12. Expand the **Application development features** node.
13. Activate the *WebSocket-Protokoll*.
14. Click on **OK**.

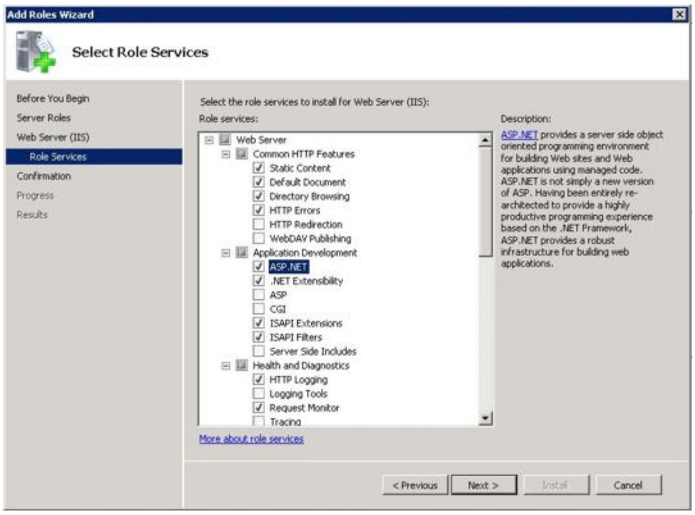
WINDOWS SERVER 2008 R2

Follow the instructions from Microsoft: https://technet.microsoft.com/en-us/library/cc771209.aspx

1. Open the **Add roles** assistant.

2. Activate the *Web Server (IIS)* role

   The **Add roles** assistant opens.



Select the role services to install for Web Server (IIS):

Role services:

- Web Server
  - Common HTTP Features
    - ☑ Static Content
    - ☑ Default Document
    - ☑ Directory Browsing
    - ☑ HTTP Errors
    - □ HTTP Redirection
    - □ WebDAV Publishing
  - Application Development
    - ☑ ASP.NET
    - ☑ .NET Extensibility
    - □ ASP
    - □ CGI
    - ☑ ISAPI Extensions
    - ☑ ISAPI Filters
    - □ Server Side Includes
  - Health and Diagnostics
    - ☑ HTTP Logging
    - □ Logging Tools
    - ☑ Request Monitor
    - □ Tracing

Description:

ASP.NET provides a server side object oriented programming environment for building Web sites and Web applications using managed code. ASP.NET is not simply a new version of ASP. Having been entirely re-architected to provide a highly productive programming experience based on the .NET Framework, ASP.NET provides a robust infrastructure for building web applications.

3. Click on **Role services**.
4. Expand the **Application development features** node.
5. Activate the following role services:

- ASP.NET
- .NET expandability
- ISAPI extensions
- ISAPI filter

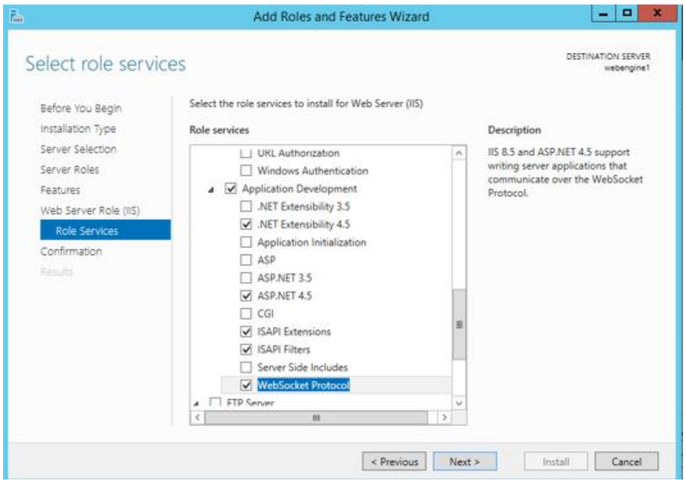
Attention: Do not deactivate any role services that have already been pre-selected by Microsoft.

Note: Use of Windows Server 2012 is recommended, because Windows Server 2008 R2 does not support WebSocket protocols.

WINDOWS SERVER 2012 (R2)

Follow the instructions from Microsoft: https://technet.microsoft.com/en-us/library/hh831475.aspx

1. Open the **Assistant to add roles and features** wizard.



Select the role services to install for Web Server (IIS)

Role services

- □ URL Authorization
- □ Windows Authentication
- ☑ Application Development
  - □ .NET Extensibility 3.5
  - ☑ .NET Extensibility 4.5
  - □ Application Initialization
  - □ ASP
  - □ ASP.NET 3.5
  - ☑ ASP.NET 4.5
  - □ CGI
  - ☑ ISAPI Extensions
  - ☑ ISAPI Filters
  - □ Server Side Includes
  - ☑ WebSocket Protocol
- □ FTP Server

Description

IIS 8.5 and ASP.NET 4.5 support writing server applications that communicate over the WebSocket Protocol.

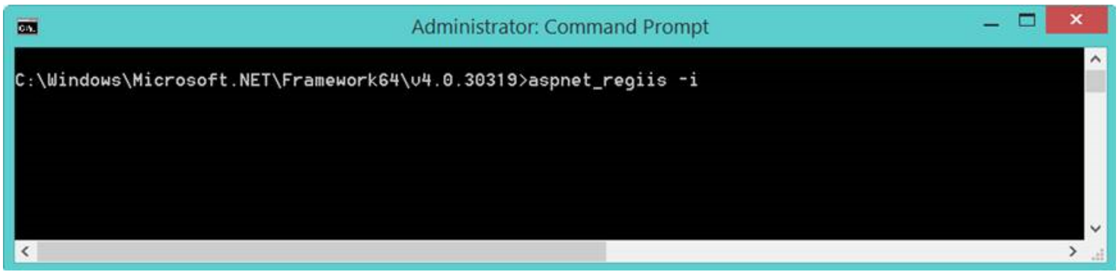
2. Expand the **Application development** node.
3. Activate the following role services:
   - NET expandability 4.5
   - ASP.NET 4.5
   - ISAPI extension
   - ISAPI filter
   - WebSocket protocol

7.1.1 .NET registration on IIS under Windows 7

In the event of a subsequent installation or upgrade of the .NET framework under Windows 7, it is necessary to register with the IIS publishing service.

To do this:

1. Open the Windows command prompt with the *as administrator* option.
2. Switch to the Microsoft.NET installation directory with the highest version number.
3. Enter command **aspnet_regiis -i**.

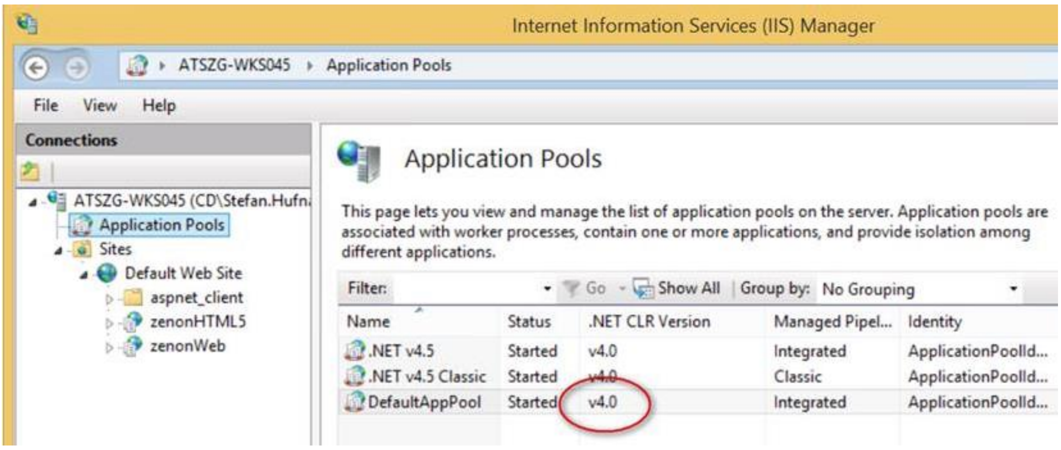


```
C:\Windows\Microsoft.NET\Framework64\v4.0.30319>aspnet_regiis -i
```

After successful registration, the current Microsoft .NET framework version is available for use with IIS.

4. Ensure that the Application Pool on the IIS in which the HTML Web Engine is operated uses the current .NET version.

   This can be checked and set with the Information Services Manager:



Application Pools

This page lets you view and manage the list of application pools on the server. Application pools are associated with worker processes, contain one or more applications, and provide isolation among different applications.

Name	Status	.NET CLR Version	Managed Pipel...	Identity
.NET v4.5	Started	v4.0	Integrated	ApplicationPoolId...
.NET v4.5 Classic	Started	v4.0	Classic	ApplicationPoolId...
DefaultAppPool	Started	v4.0	Integrated	ApplicationPoolId...

8 Connectors

In order for the HTML Web Engine to be able to query and display data from zenon Runtime, connection via a connector is needed.

The following are available:

- **SCADA Runtime Connector** (on page 22)
- **Service Grid Connector** (on page 24)

User login is possible for both connectors with the zenon user's login data.

8.1 SCADA Runtime Connector

The **SCADA Runtime Connector** is also installed when zenon Runtime is installed. The interaction starts with the user authentication for the first HTML web client.

REQUIREMENTS

The **SCADA Runtime Connector** communicates via the COM interface.
The following must be entered in zenon6.ini:

[VBA]

EIN=*1*

EVENT=*1*

START

The SCADA Runtime Connector can be started manually or automatically:

- Manually:
    - Start the application in the **Startup Tool** under **Tools**. Select, under **Available applications (current folder)**, the **SCADA Runtime Connector** entry. Confirm the selection by clicking on **Start**.
    - Start the application directly from the folder *%programfiles(x86)%\Common Files\COPA-DATA\Connectors\zrsConnector.exe*
- Automatic:
    - Activate Autostart for the Connector Container in the **Startup Tool**.

For each HTML Web Engine session, a separate **SCADA Runtime Connector** session is set up.
The following are transferred in this session:

- Variables:
    - Screen variables that are needed for the current screen display on the HTML client:
      For example, variables for displaying variable values or element dynamics. Variables can be registered and deregistered for a session. Once the user has been authenticated successfully for an HTML web client, the HTML Web Engine reports a list of variables for communication and spontaneous updating by means of the SCADA Runtime Connector. The HTML Web Engine can thus forward value changes to the web clients that are currently connected.
    - **Permanently-monitored variables**, which remain permanently advised (on page 64) and are required to execute a function in the event of a limit value violation.
- Write set value via the SCADA Runtime Connector:
  To increase security before a value change, an explicit check of the authenticity is carried out with zenon Runtime on the basis of the user data of the **HTML Web Client**. A block or removal of users by zenon Runtime becomes effective for the writing of set values immediately.

ENCRYPTED COMMUNICATION

The TCP connection between **SCADA Runtime Connector** (**zrsConnector.exe**) and **SCADA Runtime Connector Client** (**zrsConnCli.dll**) can be encrypted with AES.

To use the encrypted communication, issue an encryption password for Runtime and Client.
To do this:

1. In the zenon Startup Tool, enter the password in the **Network configuration** tab.
   This is also saved in zenon6.ini in encrypted form.
   Section: **[ZRSCONNECTOR]**
   Entry: **ENCRYPTION_PWD**

2. In the **Deployment Tool** of the **HTML Web Engine**, in the **Security options** window, set the same password in the **Security options** (on page 31) tab. This is saved in encrypted form in **web.config** in the **Encryption_Password** setting.

If, during validation, the **SCADA Runtime Connector** can be reached on the set target computer but the encryption password does not correspond, the connector test times out. You receive notification that the password set may be incorrect.

8.2 Service grid connector

The **Service Grid Connector** enables the connection of the HTML web engine to the Service Grid infrastructure. The HTML Web Engine can thus run outside of a SCADA network.

Note: The HTML Web Engine uses the **Service Grid Connector** to detect if the connection to zenon has been disconnected. In the event of disconnection, the active HTML Web Engine users are automatically logged out. In the event of configured redundancy (server/standby), a maintenance dialog is displayed and the HTML Web Engine attempts to connect to the standby server, which now becomes the primary server. An error message appears if this is not successful.

Prerequisite: The zenon Runtime whose visualization via the HTML Web Engine is wanted must be incorporated into the Service Grid infrastructure.

You can find details on the Service Grid in the Service Grid manual.

8.2.1 zenon variables

In order to use zenon variables in Service Grid, they must be configured for it.
Note: Only variables with **simple data type** are supported. Structure variables and root variables of arrays are not available.

Tipp Alarme: If not all expected alarms are displayed, check the configuration of the variables for Service Grid.

To configure variables:

1. Select the desired variable
2. Open the **External settings** group in the properties.
3. Switch to the **Service Grid settings** subgroup.
4. Configure the variable for use in the Service Grid.

Configurable properties:

- **Access permission**: Configuration of which access authorizations there are for a variable in the Service Grid.
  Select from drop-down list:

- *None*: This variable is not available in the Service Grid.
- *Read*: There is read access to this variable in the Service Grid.
- *Read and write*: This variable can be read and written in the Service Grid.

Default: *none*

You can find detailed information on variables in zenon in the **Variables** manual.

8.3 Reconnect

If the connection between Web Engine and Runtime or Service Grid is lost, an attempt is made to establish the connection again.

After successfully reconnecting, login is automatic if so configured. If the currently-active user session is also disconnected, an attempt is made to reestablish this too.

Automatic reconnection is carried out in the following situations:

- Interruption of the connection between Web Engine frontend and backend.
  For example: disconnection with mobile connection or IIS restart.
- Loss of connection between Web Engine backend and connector/data hub.
- Loss of connection by reloading zenon Runtime.

Note on Service Grid connector: With network projects for the Web Engine, the addresses of the server must be entered with their fully-qualified domain name (FQDN). This is because the IDs of the clients are derived from the computer’s FQDN. If the configured network names and the computer’s FQDNs are different, the service grid connector cannot correctly detect the status of zenon Runtime.

9 Client authentication for a connection to Runtime

Runtime data from zenon Runtime, such as variable values for display or for display dynamics, are only provided if the HTML client can authenticate itself to zenon Runtime. This can happen in two ways by means of user name and password:

- Automatic login by configuring a user as part of deployment.
  For more details, see the Deployment of the Web Engine (on page 26) chapter.
- Manual login by the zenon Web Client in a *login* screen.
  For details, see the Create *login* screen chapter.

Note: Authentication can be carried out by transferring the login data (user name and password) for a zenon user or an Active Directory user. The user data is validated by zenon Runtime.

10 Deployment of the Web Engine

The **zenon Web Engine Deployment Tool** offers important operations for the management of the web engine as a web application in the Internet Information Services (IIS).

To start the **Deployment Tool**:

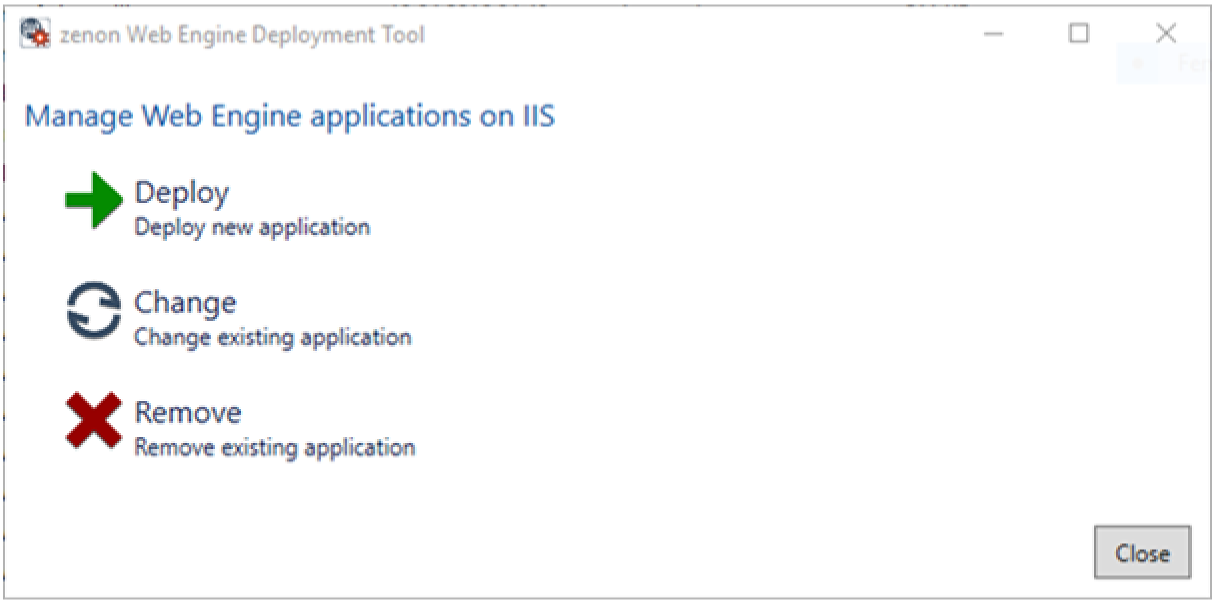
1. Open the zenon **Startup Tool**.
2. click on the **Tools** button.
3. Under **Available applications**, select the *Web Engine Deployment Tool* .
4. Click on **Start**.

   The tool is started

> **Information**
>
> **Note:** The **Deployment Tool** is automatically installed with the zenon Web Server. Administrator rights are required for the use of this tool.
>
> The **Deployment Tool** is only available in English.

10.1 zenon Web Engine Deployment Tool

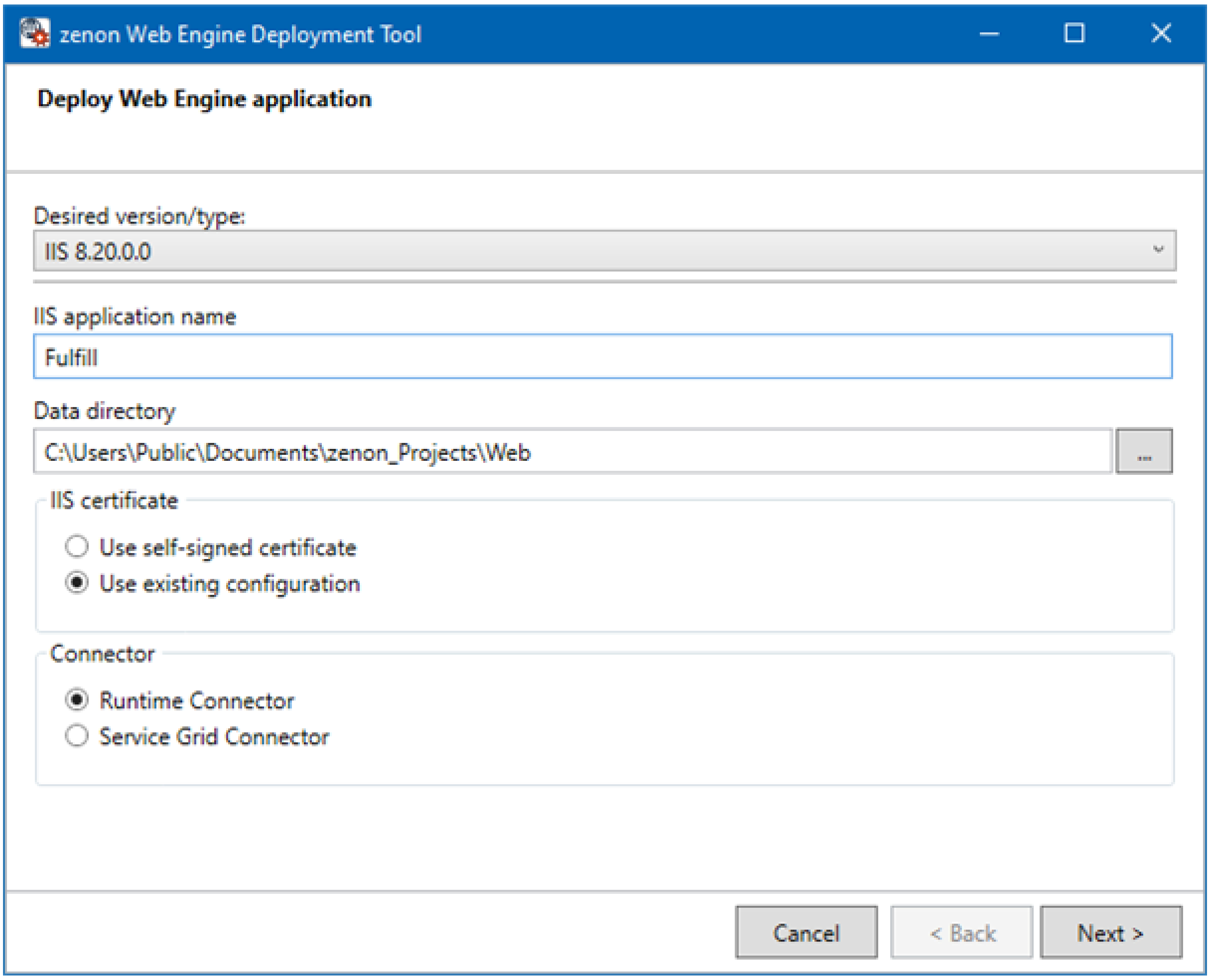
HTML web engine instances are administered on the IIS with the **zenon Web Engine Deployment Tool**. You can create new instances and amend or remove existing ones.


zenon Web Engine Deployment Tool
Manage Web Engine applications on IIS
Deploy
Deploy new application
Change
Change existing application
Remove
Remove existing application
Close

Option	Description
Deploy	Provides a new instance of the **HTML Web Engine** on the IIS. The necessary options are configured in the following tabs.
Change	Updates existing Web Engine Applications. The following can be amended for the **HTML Web Engine** in the following tab: - Version. - Configuration. - Selection of the connector and its configuration. - Path of the data folder. Data Directory in which the exported project data (***.webx**) is. - Security settings: User name, password or switch for automatic sign-in. **Note:** Not available if no valid Web Engine application is available on the IIS.
Remove	Removes a running application of the **HTML Web Engine**. The version to be removed is selected in the following tab. If the web engine is to be replaced by a more recent version, use the **Change** option. **Note:** Not available if no valid Web Engine application is available on the IIS.

10.1.1 General settings for the Web Engine

In this dialog, you configure the general settings for the operation of the HTML Web Engine.



Deploy Web Engine application

Desired version/type: IIS 8.20.0.0

IIS application name: Fulfill

Data directory: C:\Users\Public\Documents\zenon_Projects\Web

IIS certificate

○ Use self-signed certificate

● Use existing configuration

Connector

● Runtime Connector

○ Service Grid Connector

The options that are available depend on the option that was selected in the start dialog:

- **Deploy**: New configuration of a **HTML Web Engine**.
- **Change**: Amendment to an existing **HTML Web Engine**.
- **Remove**: Removal of an existing **HTML Web Engine**.

Option	Description
Web Engine application to change	Selection of the instance that is to be amended. **Note:** Not available with the **Change** selection in the start dialog.
Web Engine application to change	Selection of the instance that is to be deleted. **Note:** Not available with the **Remove** selection in the start dialog.

Option	Description
Desired version/type	Selection of the version from the drop-down list. **Note:** Only available for **Deploy** and **Change**.
IIS application name	Entry of the desired name for the Web Engine application This name will be part of the URL under which the HTML5 web page will later be reached. Example**: https://Server address/Application name** **Note:** Only available for **Deploy** and **Remove** (display only).
Data directory	Folder from which the Web Engine is to read the exported project data. **Note:** The webx file generated by the **Web Engine Compiler** must be available in this folder. When the HTML Web Engine is first accessed, the first webx file in the file list is loaded. If a webx file that was loaded by the web engine is amended or deleted, the Web Engine automatically restarts and in turn loads the first webx in the folder. **Default folder:** *C:\Users\Public\Documents\zenon_Projects\Web* **Note:** Only available for **Deploy** and **Change**.

IIS CERTIFICATE

zenon HTML Web Engine communication is always via a secure (HTTPS, port 443) connection.

In this area, you define whether the **Deployment Tool** uses a self-signed certificate for communication. If there is not yet a self-signed certificate, it is created by the **Deployment Tool**. This certificate is assigned to the IIS.

Note: A security certificate is a mandatory requirement for communication between the zenon Web Server and zenon Web Client.

Option	Description
Use self-signed certificate	Activate this option if you want to create a temporary, self-signed certificate. This option is offered as a default for the first **Deploy**.

Option	Description
	Note: This option is mandatory if there is no certificate present on the IIS.
Use existing certificate	Select this option if there is already a valid configuration. If there is already a valid configuration, this option is offered as a default. **Note:** Use of an official certificate from a certification body is recommended. **Possible certification body:** **https://www.digicert.com/ssl-certificate-installation-microsoft-iis-8.htm**

CONNECTOR

Selection of the connector (on page 22) via which the connection is established.

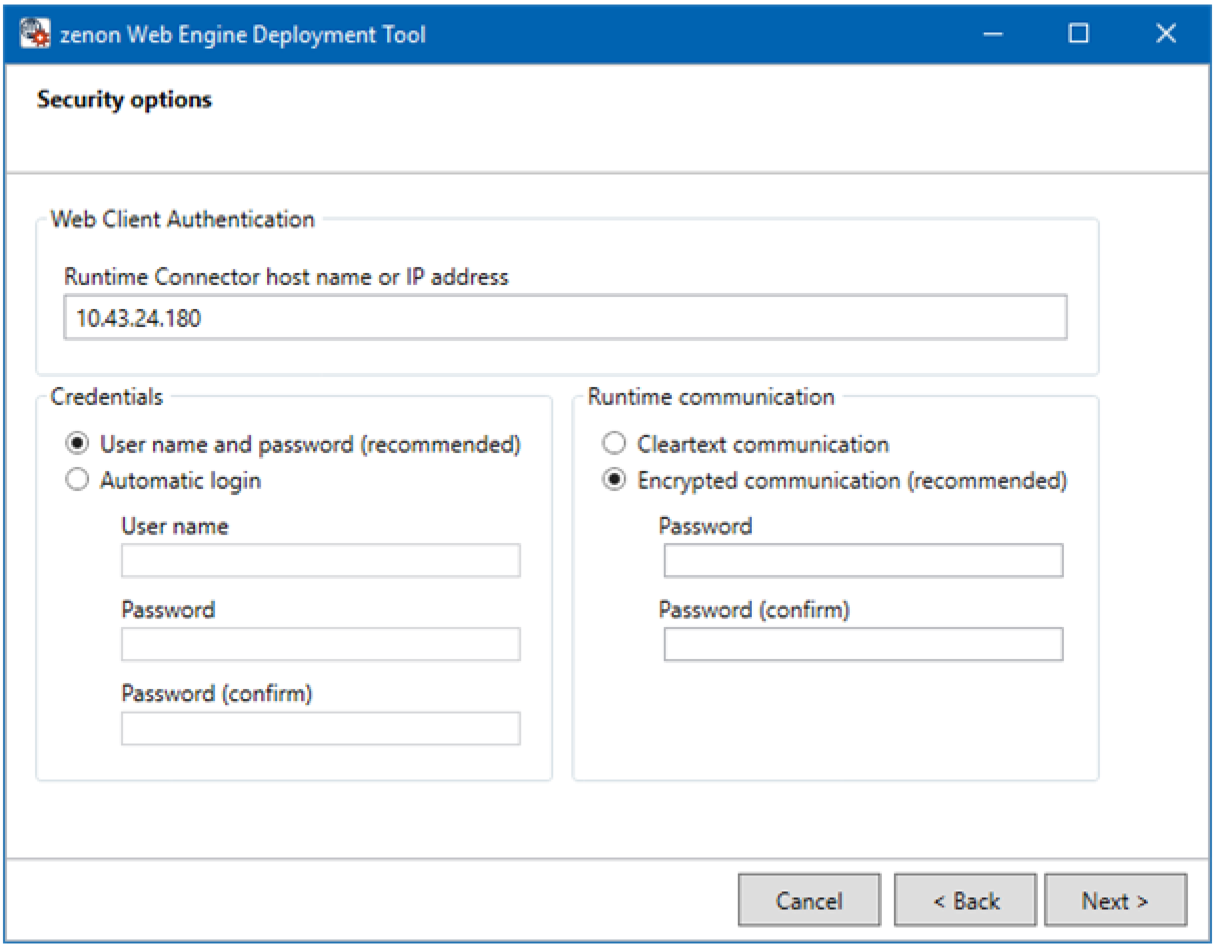
Option	Description
Runtime Connector	The connection is established via the Runtime connector.
Service Grid Connector	The connection is established via the Service Grid connector.

NAVIGATION

Cancel	Discards changes and closes the dialog.
Back	Goes back one tab in the tool.
Next	Goes forward one tab in the tool.

10.1.2 Settings for the Runtime connector

In this dialog, you configure the general settings for the operation of the HTML Web Engine via the Runtime connector.
After configuration, the settings are validated (on page 36) in a separate tab.



zenon Web Engine Deployment Tool

Security options

Web Client Authentication

Runtime Connector host name or IP address: 10.43.24.180

Credentials

● User name and password (recommended)
○ Automatic login

User name

Password

Password (confirm)

Runtime communication

○ Cleartext communication
● Encrypted communication (recommended)

Password

Password (confirm)

WEB CLIENT AUTHENTICATION

Settings for the connection.

Option	Description
Application version	zenon version of the instance (display only). **Note:** Only available for **Remove**.
Runtime Connector host name or IP address	Enter the host name or the IP address of the computer on which the zenon Runtime and the SCADA Runtime Connector are installed. **Note:** Only available for **Deploy** and **Change**.

CREDENTIALS

In this area, you define how the authentication of the zenon Web Client is to be carried out.

Option	Description
User name and password (recommended)	The zenon web client is authenticated by manual entry of the user name and password. **Note:** The user name and password must be entered in a zenon **login** screen.
Automatic login	Activate this option if you want the web engine to automatically establish a connection to zenon Runtime. The given user data is used for authentication. **Attention:** When this option is used, each zenon Web Client receives a connection to zenon Runtime.
User name	Input field for zenon user name. Enter the desired user name here.
Password	Input field for zenon user password. Enter the user password here. **Note:** Not available if *User name and password* has been selected for **Web Client Authentication**.
Password (confirm)	Enter the user password again. **Note:** Not available if *User name and password* has been selected for **Web Client Authentication**.

RUNTIME COMMUNICATION

Configuration of communication to the Web Engine.

Option	Description
Cleartext communication	Communication to the web engine is implemented by means of plain text without encryption.
Encrypted communication (recommended)	Communication to the web engine is encrypted. **Note:** The settings for Runtime are configured in the **Startup Tool** in **Network configuration** tab

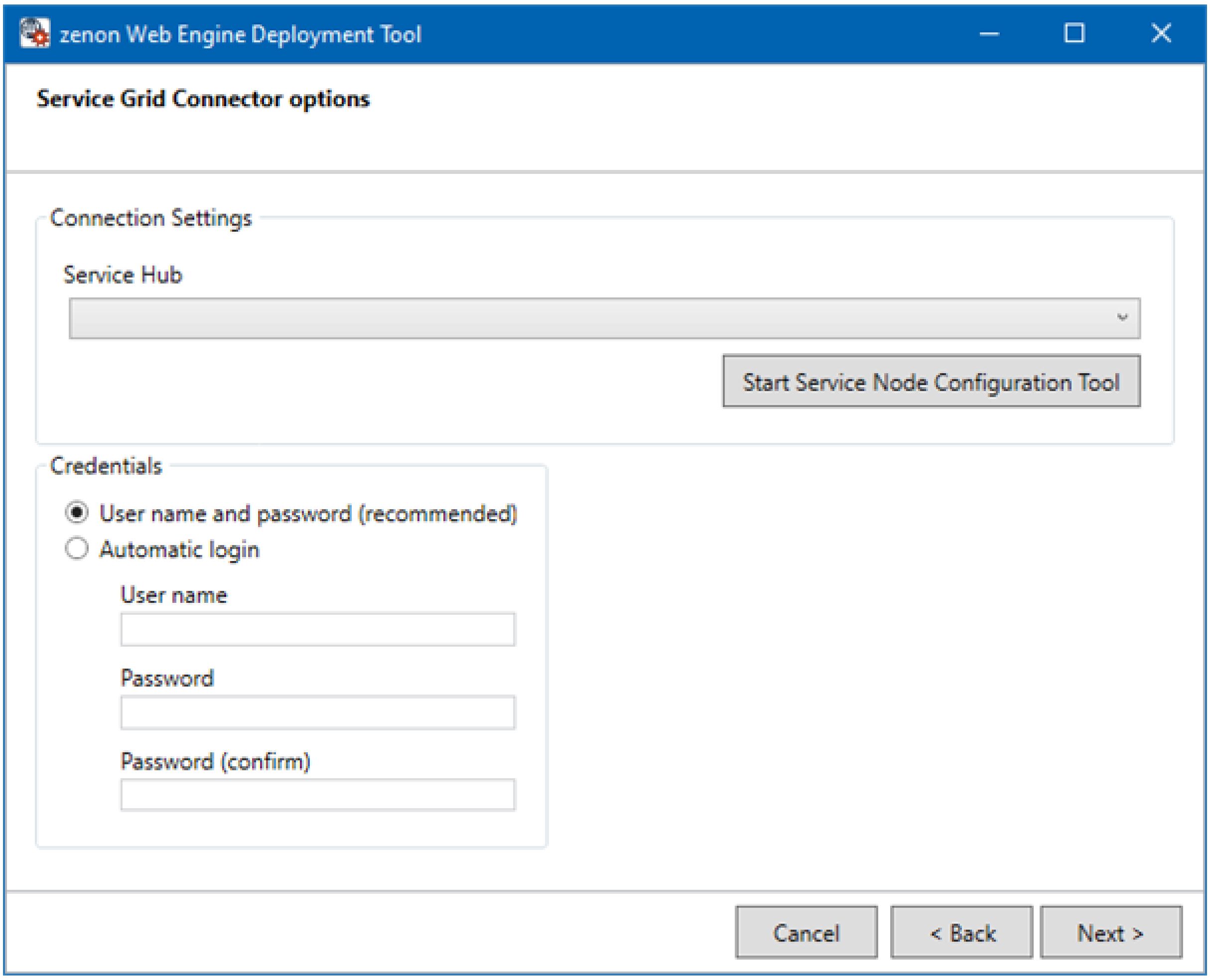
Option	Description
	with the **Encrypt Runtime Connector communication** property. The passwords for Runtime and the HTML Web Engine must correspond.
Password	Input field for password for secure communication. Enter the user password here.
Password (confirm)	Enter the user password again.

NAVIGATION

Cancel	Discards changes and closes the dialog.
Back	Goes back one tab in the tool.
Next	Goes forward one tab in the tool.

10.1.3 Settings for the service grid connector

In this dialog, you configure the settings for the operation of the HTML Web Engine via the **Service Grid Connector** (on page 24).
After configuration, the deployment is started (on page 37) by clicking on the **Next** button.



zenon Web Engine Deployment Tool

Service Grid Connector options

Connection Settings

Service Hub

Credentials

● User name and password (recommended)
○ Automatic login

User name

Password

Password (confirm)

CONNECTION SETTINGS

Settings for the connection to the Service Grid.

Option	Description
Service Hub	Selection of the **Service Hub** from a drop-down list. All configured service hubs are offered.
Start Service Node Configuration Tool	Click on the button to open the **Service Node Configuration Tool**. The connection to an additional **Service Hub** can thus be configured if

Option	Description
	the desired **Service Hub** does not appear in the drop-down list

CREDENTIALS

In this area, you define how the authentication of the zenon web client is to be carried out.

Option	Description
User name and password (recommended)	The zenon web client is authenticated by manual entry of the user name and password. **Note:** The user name and password must be entered in a zenon **login** screen.
Automatic login	Activate this option if you want the web engine to automatically establish a connection to zenon Runtime. The given user data is used for authentication. **Attention:** When this option is used, each zenon Web Client receives a connection to zenon Runtime.
User name	Input field for zenon user name. Enter the desired user name here.
Password	Input field for zenon user password. Enter the user password here. **Note:** Not available if *User name and password* has been selected for **Web Client Authentication**.
Password (confirm)	Enter the user password again. **Note:** Not available if *User name and password* has been selected for **Web Client Authentication**.

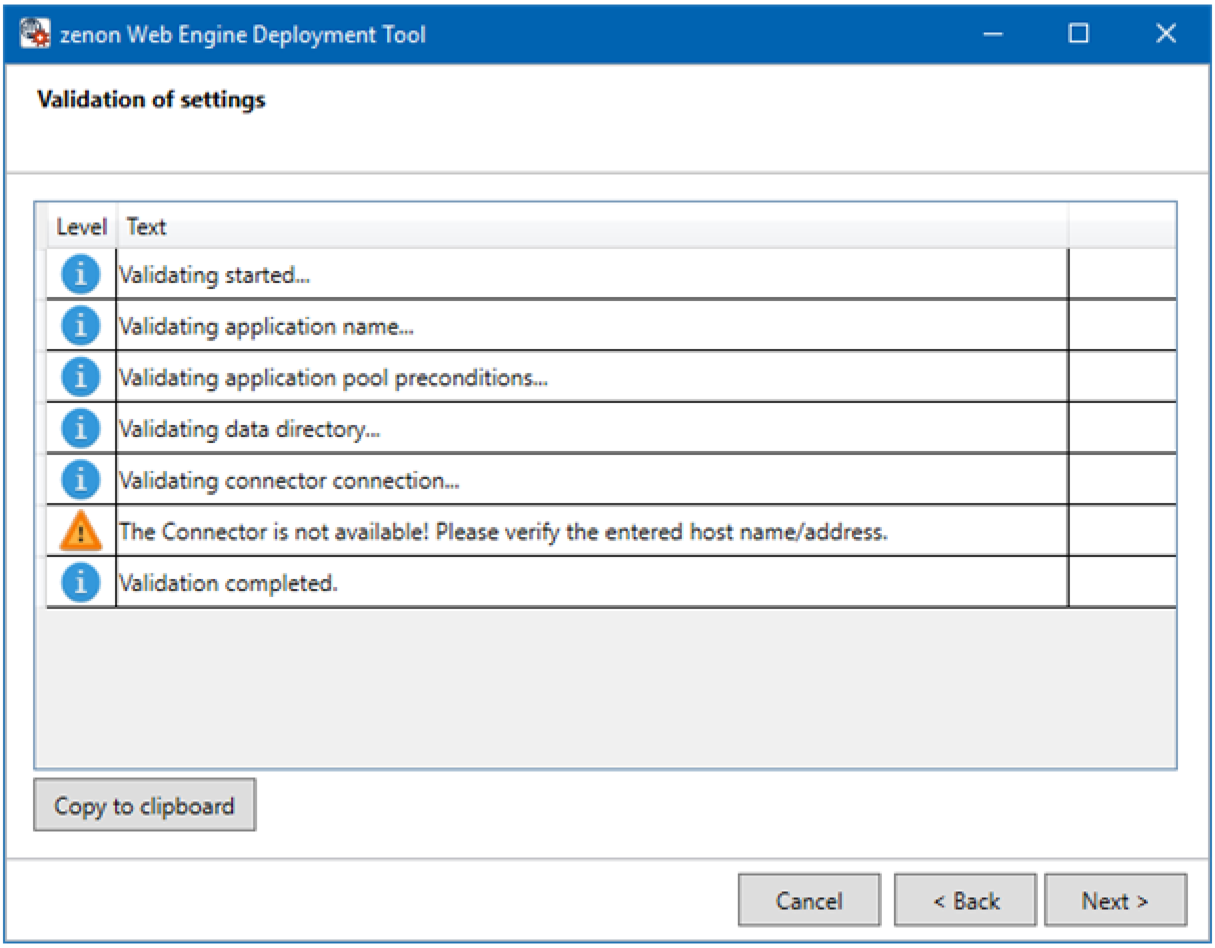
NAVIGATION

Cancel	Discards changes and closes the dialog.
Back	Goes back one tab in the tool.
Next	Goes forward one tab in the tool.

10.1.4 Validation of the settings

The settings are validated in this dialog. The progress is shown with a green bar during validation.

The result of the validation is shown in a list.

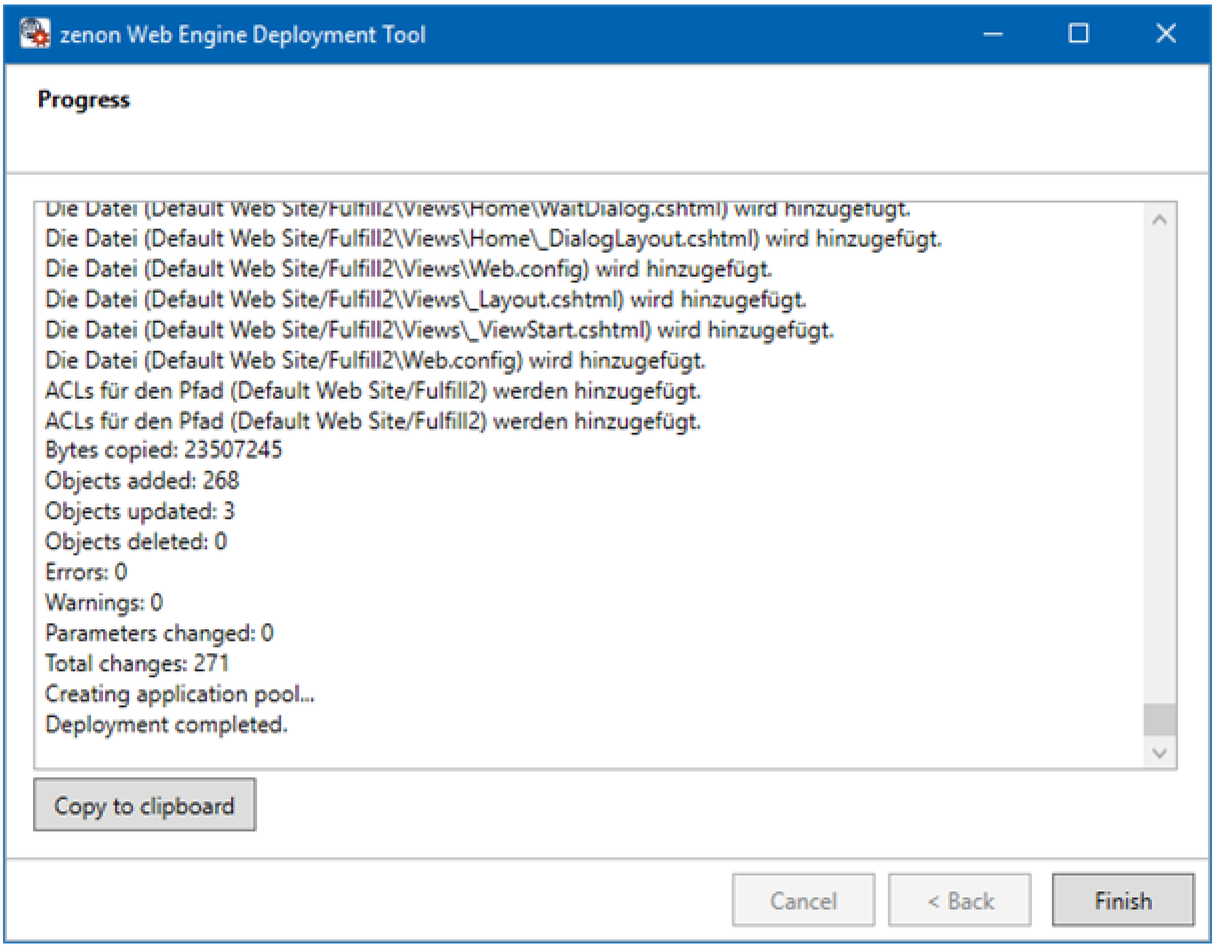


Level	Text
ⓘ	Validating started...
ⓘ	Validating application name...
ⓘ	Validating application pool preconditions...
ⓘ	Validating data directory...
ⓘ	Validating connector connection...
⚠	The Connector is not available! Please verify the entered host name/address.
ⓘ	Validation completed.

Copy the output to the clipboard by clicking on the **Copy to clipboard** button.

10.1.5 Progress and result

In this dialog, you see the progress of the **Web Engine Deployments** on the IIS.



```
Die Datei (Default Web Site/Fulfill2\Views\Home\WaitDialog.cshtml) wird hinzugefügt.
Die Datei (Default Web Site/Fulfill2\Views\Home\_DialogLayout.cshtml) wird hinzugefügt.
Die Datei (Default Web Site/Fulfill2\Views\Web.config) wird hinzugefügt.
Die Datei (Default Web Site/Fulfill2\Views\_Layout.cshtml) wird hinzugefügt.
Die Datei (Default Web Site/Fulfill2\Views\_ViewStart.cshtml) wird hinzugefügt.
Die Datei (Default Web Site/Fulfill2\Web.config) wird hinzugefügt.
ACLs für den Pfad (Default Web Site/Fulfill2) werden hinzugefügt.
ACLs für den Pfad (Default Web Site/Fulfill2) werden hinzugefügt.
Bytes copied: 23507245
Objects added: 268
Objects updated: 3
Objects deleted: 0
Errors: 0
Warnings: 0
Parameters changed: 0
Total changes: 271
Creating application pool...
Deployment completed.
```

Copy the output to the clipboard by clicking on the **Copy to clipboard** button.

Once the procedure has been completed, click on the **Finish** button to close the Deployment Tool.

11 Engineering in the Editor

The data required by the Web Engine for visualization is created from a zenon standard project. The zenon Editor must be open for this.

Note: When engineering, please take the supported properties, screen elements and functions of the HTML Web Engine into consideration.

CREATE PROJECT

For HTML Web Engine visualization with the zenon web server, create a standard zenon project. You can also use certain resources from a global project for this.

> ⚠**Attention**
>
> Only the *standard* screen type can be used as a start screen for the HTML web engine. Special screen types are not suitable.

11.1 Create, amend and call up an HTML screen

Set up an *HTML* screen and create a screen-switch function

CREATING A SCREEN OF THE TYPE HTML

Engineering

Two procedures are available to create a screen:

- The use of the screen creation dialog
- The creation of a screen using the properties

Steps to create the screen using the properties if the screen creation dialog has been deactivated in the menu bar under **Tools**, **Settings** and **Use assistant**:

1. Create a new screen.

   To do this, select the **New screen** command in the tool bar or in the context menu of the **Screens** node.

2. Change the properties of the screen:

   a) Name the screen in the **Name** property.

   b) Select *HTML* in the **Screen type** property.

   c) Select the desired frame in the **Frame** property.

3. Configure the content of the screen:

   a) Select the **Elements (screen type)** menu item from the menu bar.

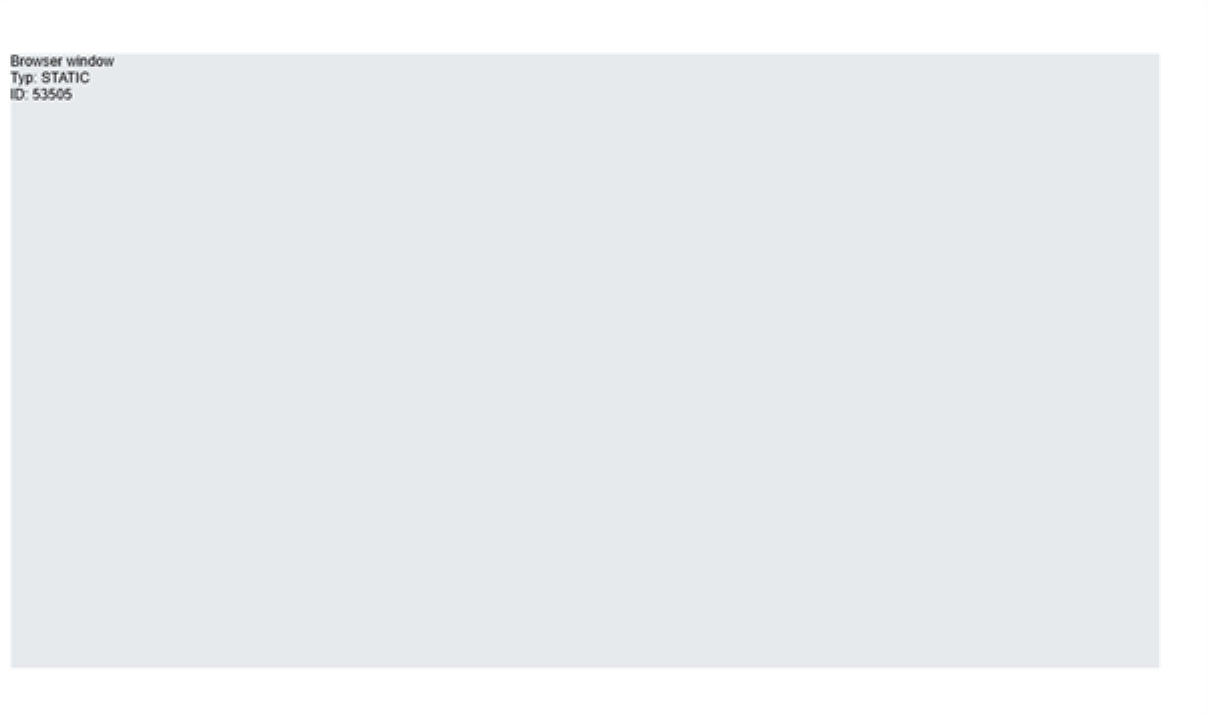
   b) Select *Insert template* in the drop-down list.
   The dialog to select pre-defined layouts is opened. Certain control elements are inserted into the screen at predefined positions.

   c) Remove elements that are not required from the screen.

d) If necessary, select additional elements in the **Elements** drop-down list. Place these at the desired position in the screen.

4. Create a screen switch function.

HTML SCREEN



Control element	Description
Insert template	Opens the dialog for selecting a template for the screen type. Templates are shipped together with zenon and can also be created by the user. Templates add pre-defined control elements to pre-defined position in the screen. Elements that are not necessary can also be removed individually once they have been created. Additional elements are selected from the drop-down list and placed in the zenon screen. Elements can be moved on the screen and arranged individually.
Browser	Control elements for the browser.
Browser Window	The browser is displayed.
Address field	Field for entry of the address (URL).
Home page	The start page is called up.

Control element	Description
Search forward	Go forward.
Search back	Go back.
Refresh search	Refresh display.
Cancel	Stop navigation.
Search	Control elements for the search. When clicking a link in the **Search** field, the corresponding page is shown in the browser. So e.g. in the field **Search** a navigation bar or the results of a search engine can be displayed without changing the contents, when a link is activated.
Search window	Display of the search.
Search field	Search for address or file.
Home	Back to home in the search area.
Forward	Page down in the search area.
Search back	Page up in the search area.
Refresh	Refresh display in search area.
Stop	Cancel search action.
Filter	Open filterbox.

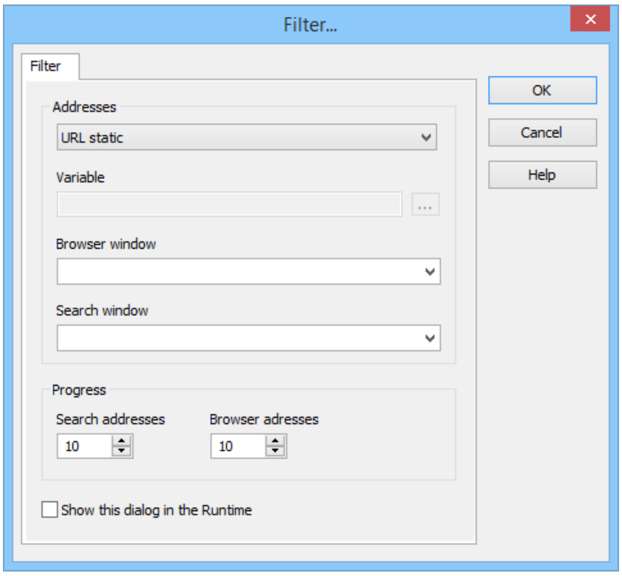
AMEND SCREEN

To amend the screen for use with the HTML Web Engine:

1. Activate the **Available in web** property in the group **General** for this screen.
2. Note the **options for opening external web pages** (on page 41).

CREATE SCREEN SWITCHING

Create a screen switch function in order to be able to call up the screen in the Runtime.



Filter...

Filter

Addresses

URL static

Variable

Browser window

Search window

Progress

Search addresses: 10

Browser adresses: 10

□ Show this dialog in the Runtime

The HTML Web Engine supports the following options:

Option	Description
Addresses	- *URL static* The URL is set as static. **Note:** This setting is mandatory.
Browser window	Enter the complete URL of the external web page here, including **https://** **Example:** **https://serveraddress**

11.1.1 Options for opening external web pages

The opening of external web pages is supported by the HTML Web Engine either by means of an embedded browser or as a new browser window.

EMBEDDED BROWSER WINDOW:_

To see an embedded browser window that you have entered in the screen switching function from the URL in Runtime:

- In the **Elements [screen type name]** menu, select **Browser -> Browser window**.

- Draw the frame for the browser window in the screen. The given website (URL) is displayed in this frame.

Note: The opening of the embedded display (**iFrame**) only works if the X-Frame options on the remote server are configured accordingly. In addition, the address of the embedded web page must also be available for HTTPS via the HTTPS connection between the HTML Web Engine and HTML web client.

NEW BROWSER WINDOW:

To open an external web page in a new browser window:

- Delete the browser window control from the inserted template in the screen.

  The external web page is opened in a new browser window if there is no browser window control element in the screen.

Note: The calling up of several URLs is supported via script. However some browsers prevent tabs being called up with their pop-up blocker. In such cases, a dialog appears with the URLs that cannot be called up. These URLs can be opened manually.

11.2 Example: Simple start screen

Check the functionality of the HTML Web Engine with a simple example.

To do this:

1. Create a *standard* screen.
2. Activate the **Available in web** property in the group **General** for this screen.
3. Enter this screen into the project properties in the **Graphical design** group as **Start screen**.

   Note: You can define any desired start page for the web with an Autostart script.
4. Add simple elements to the screen, for example a **Rectangle**, **Circle** or **Static text**.
5. Ensure that the current project is set as a start project.
6. Export the project for HTML5 visualization.

EXTENSION: AUTHENTICATION WITH LOGIN

In order to be able to exchange data with zenon Runtime, the zenon Web Client must be authenticated as a user to Runtime. You can read more details about this in the following chapter: **Client authentication for a connection to Runtime** (on page 25).
This is how you expand a project for manual authentication:

1. Add a dynamic element to display a variable value, for example **dynamic text** or **numeric value**.

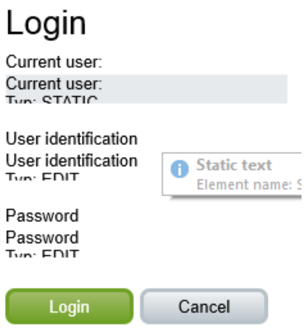
2. Assign this element a variable from the project.

   Note: Variable values can also be used for position or color dynamics of an element.

3. Create a new *login* screen which you can display in the visualization.
   Use the **Screen switch** function for display.

4. Activate the **Available in web** property in the screen in the **General** group.

5. In the *login* screen, add the elements **user name**, **password**, **login** and **Cancel**.



6. Carry out the HTML export for the current project.

The zenon Web Client now has the possibility to carry out authentication by means of entry of the user data. If authentication is successful, a connection to zenon Runtime is established. As a result of this, variable values for HTML5 visualization are available, for example.

11.3 Supported functionalities for HTML visualization

For HTML visualization, basic elements, properties and functions are available:

GENERAL

Functionality	Support for
Variables	- Display of static and dynamic values. - **Write set value** in the HTML Web Engine directly - Making the element display dynamic with variable values. For example **Visibility/flashing**, rotation, positioning, size adjustment... - Limit values for the dynamic aspects of the element display (such as **Limit value color**, **Limit value text**) are fundamentally supported. - Permanent reading (on page 64) of variables that execute a zenon function in the event of limit value violations. **Attention:** Limit value information of reaction matrices is not

Functionality	Support for
	supported! Archives and variables can also be taken from the **Service Grid**.
Font types and font lists	- Selection and display of any desired font lists that are available on the system. **Note:** Selected font types must be available on both the project configuration computer and on the zenon Web Client. The steps must be defined in the local project. - Display in *Normal* font style, *italic* and *bold*. - Selection and display of the font in accordance with the font list. - Online switching of the font list. **Note:** The first font list of the zenon project is shown when a session starts

REPLACE LINKS

Linkings are replaced in the same way as substitution in zenon Runtime.

> ⚠**Attention**
>
> Limitation when substituting variables for symbols:
>
> If variable names are only different due to upper case and lower case, several target variables can be found that meet the criterion during substitution. This is the case if the **Note upper and lower case** option is deactivated.
>
> In this case, one of the variables is selected at random in the Editor and Runtime. This selection cannot be reproduced in the HTML Web Engine.
>
> Behavior when compiling the files for the web:
>
> - If several possible target variables are found, the compiling will be unsuccessful. An error message is issued.
> - A warning is given is no possible target variables are found.

Hint: When configuring projects, ensure that there are always unique variable names that do not depend on a distinction between upper case and lower case.

GLOBAL PROJECT

The HTML export takes the use of the following resources from a global project into account:

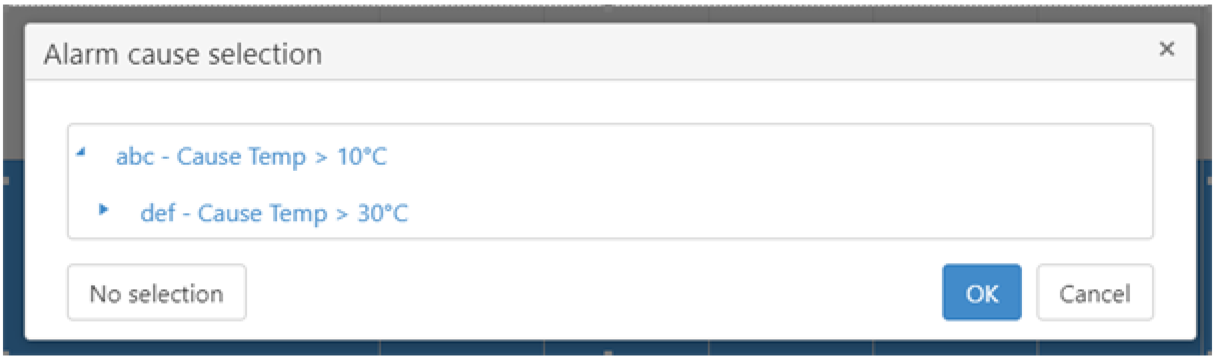
- **Frames**
- **Color Palettes**
- **Language Files**

11.3.1 AML and CEL

The *AML* and *CEL* screen types support the following functions for the HTML Web Engine:

ALARMS: ALARM CAUSE REQUIRED

If the **Alarm cause required** property is activated for limit values, the pre-configured alarm cause can be selected in the Web Engine.
The dialog for selection is called up if an alarm cause is necessary:



Option	Description
List of alarm causes	Pre-configured context list with possible alarm causes. Selection by clicking on the entry.
No selection	Clears selection.
OK	Accepts selection and closes the dialog.
Cancel	Discards selection and closes dialog.

LIST DISPLAYS

Lists are subdivided into pages. A footer for navigation is shown under the list.
List entries can be called up by clicking on the symbols. The number of the list entries shown can be defined by clicking on the drop-down list.

SORTING IN RUNTIME

Lists in AML and CEL can be shown in sorted form.
To change the sorting of a list, click on the column title.
Clicking switches between:

- Ascending
- Descending
- Standard (as supplied in zenon Runtime)

The behavior of the header, the sorting and the column widths is configured in the zenon Editor in the project properties.

SORTING OF TEXT

Text is sorted according to natural sorting:

- alphabetic sequence
- Figures with several digits are seen as a character

Example:

- Alphabetic sorting: 1, 11, 2, 3, 33, 4
- Natural sorting: 1, 2, 3, 4, 11, 33

COLUMN LABELING

The **Identification** and **Resources label** columns are shown with the labeling **Identification** and **Resource Label**. The language of the texts used in these columns can be switched.

LANGUAGE SWITCHING IN RUNTIME

Language switching is available for the following columns of the AML and CEL:

- **text**
- **Identification**
- **Resources label**
- **Measuring unit**

Note: Language switching is carried out for each cell individually. If switching takes place in Runtime, the switching to the new language can take a few seconds.

TIME FORMATS

Date and time in AML and CEL are shown in the localized display of the respective client. In doing so, the **UTC-DateTime** is transfered and reformatted on the basis of the settings of the local computer.

You can find further information and examples in the **Runtime** manual, in the **Handling of date and time** chapter.

11.3.1.1 Properties and options

The HTML Web Engine supports the following properties and options for control elements, screen switching and project properties:

AML AND CEL SCREEN SWITCH FUNCTION:

Tab	Group	Settings and notes
General	**Variable filter**	- **Variable name** - **Identification** **Note:** Capitalization is not taken into account.
	Alarm type (AML only)	Options: - **Only non-acknowledged alarms** - **Only cleared alarms** - **Only current alarms** **Note:** Other settings are ignored.
	Origin of the data	Settings are ignored. **Historical data** from zenon Runtime is always used. Maximum: 65535
	Runtime settings	Settings are ignored. The **Show list without refresh** list is always used. The list entries that were present at the time of the screen switching in zenon

Tab	Group	Settings and notes
		Runtime are shown.
Time	**Filter**	Options: - **No time filter** - **Absolute time period** - **Relative time period**
	Settings	The **Preset** option only. All other settings are ignored.
Column settings	**Columns**	- **Alarm condition** (Including display by means of circle symbol or graphics file.) - **Time received** - **Time cleared** (AML only) - **Time acknowledged** (AML only) - **text** - **Variable name** - **Value** - **Measuring unit** - **User - full name** - **Computer name** - **Comment** **Note:** The set display sequence is taken into account as follows. The column labeling can be edited and the language can be switched.
	Table settings	**Always active:** - **Use alternating background colors** - **Display grid** - **Sort descending** **Note:** Color palette switching is supported for:

Tab	Group	Settings and notes
		- **Row color 1** - **Row color 2**
Equipment Modeling		From local and global project.

CONTROL ELEMENTS FOR THE ALARM MESSAGE LIST SCREEN TYPE

Group	Subgroup	Settings and notes
Header and grid	**Header**	**Show header**: Setting always *active*. **Display style**: Setting always *standard*. **Fill color**: provides background color. Is also used in the footers in the web for: - Static/fixed color. - Color palette and switching of color palettes. **Font** is also in the footer in the web Applied for: - Static selection. - Switching of font lists. **Text color** is also in the footer in the web Applied for: - Static/fixed color. - Color palette and switching of color palettes.
Fill	**Fill**	**Text color** is used for: - Static/fixed color - Color palette and switching of color palettes
Representation	**Representation**	Is used for: - Static selection

Group	Subgroup	Settings and notes
		- Switching of font lists

AML PROJECT PROPERTIES

Property group	Supported properties
Alarm Message List	**Header AML**
Data storage AML	not available.
Alarm received	All.
Alarm cleared	All.
Alarm acknowledged	All.
Confirm alarm acknowledgement	Not available. Alarms whose acknowledgment has been confirmed are not shown in the HTML Web Engine.
Alarm status line	not available.

11.3.1.2 Alarm acknowledgment

Alarms can be selected and acknowledged individually. Only alarms with the **To acknowledge** property activated are supported.

In doing so, the following applies:

- Alarms are acknowledged using the **Acknowledge** button in the *alarm administration* screen type.
- If the **To acknowledge** property is not activated for a limit value, the **Acknowledge** button in the *alarm administration* screen is deactivated.
- The **Acknowledge all** button of the *alarm acknowledgment* screen type is not supported.
- The acknowledgment of an alarm required the necessary function authorizations for signed-in users.
  An error message is shown if this is not the case.

  Note: Ensure that the user has the *Acknowledge alarm via Alarm Message List screen* function authorization.
  You can find further information in the User administration manual in the Function authorizations chapter.

11.3.2 Screens, frames, elements and symbols

Support for screens and frames as well as elements and symbols in screens.

SCREENS AND FRAMES

Screen/Frame	Support for
Frame	▸ Calling up rectangular frames at an absolute position. **Note:** Frame names must be unique. They are not case sensitive.
Screens in general	▸ Display of screens in the size of the linked frame. ▸ Display of background color and background graphics. ▸ Execution of a start and end function for a screen.
Standard screen	▸ Display of this type of screen.
Login screen	▸ Display of this type of screen. ▸ User authentication with the screen-type-specific elements **Enter user name** and **Enter password** or **Login** command.
AML screen type (on page 45)	▸ Display of this type of screen. ▸ Display of dynamic AML lists. Dynamic means that the data displayed is updated.
CEL screen type (on page 45)	▸ Display of this type of screen. ▸ Display of dynamic CEL lists. Dynamic means that the data displayed is updated.
HTML screen type (on page 38)	▸ Display of this type of screen. ▸ Display of the elements specific to the screen type, such as web browser
ETM screen type (on page 55)	▸ Display of this type of screen. ▸ Static display of simple line diagrams. Static means the the data displayed is not updated.

SCREEN ELEMENTS

STATIC SCREEN ELEMENTS

The following are supported:

Element	Support for
Circle	- Element-specific display options for display, colors and fill options. Effects are not supported. - Display dynamics via variables for coloring and position dynamics. - **Invert color gradient** property. When selecting *color gradient* for the **Fill pattern** property, the brightness gradient of the fill pattern is shown as inverted.
Arc of a circle	- Element-specific display options for display, colors and fill options. Effects are not supported. - Display dynamics via variables for coloring and position dynamics.
Segment of a circle	- Element-specific display options for display, colors and fill options. Effects are not supported. - Display dynamics via variables for coloring and position dynamics. - **Invert color gradient** property. When selecting *color gradient* for the **Fill pattern** property, the brightness gradient of the fill pattern is shown as inverted.
Line	- Element-specific display options for display, colors and fill options. Effects are not supported. - Display dynamics via variables for coloring and position dynamics.
Polygon	- Element-specific display options for display, colors and fill options. Effects are not supported. - Display dynamics via variables for coloring and position dynamics. - **Invert color gradient** property. When selecting *color gradient* for the **Fill pattern** property, the brightness gradient of the fill pattern is shown as inverted. - **Note:** The **Rounding** property is not supported.
Polyline	- Element-specific display options for display, colors and fill options. Effects are not supported. - Display dynamics via variables for coloring and position

Element	Support for
	dynamics. - **Note:** The **Rounding** property is not supported.
Rectangle	- Element-specific display options for display, colors and fill options. Effects are not supported. - Display dynamics via variables for coloring and position dynamics. - **Invert color gradient** property. When selecting *color gradient* for the **Fill pattern** property, the brightness gradient of the fill pattern is shown as inverted.
Static text	- Element-specific display options for display, colors and fill options. Effects are not supported. - Display dynamics via variables for coloring and position dynamics.

GENERAL DYNAMIC SCREEN ELEMENTS

The following are supported:

Element	Support for
Button	- Element-specific display options for display, colors and fill options. Effects are not supported. - Display dynamics via variables for coloring and position dynamics. - Execution of functions of the local project. **Note:** The corresponding user level is checked for the execution of the action. - **Fill pattern** and **Color gradient** properties: **Color gradient** only has an effect if *color gradient* is selected for **Fill pattern**.
Combined element	- Element-specific display options for display, colors and fill options. Effects are not supported. - Display dynamics via variables for coloring and position dynamics. - Setting of numerical limit values via the standard dialog. - Switch and push button

Element	Support for
	For details, see the **Combined element** (on page 58) section.
Dynamic text	- Element-specific display options for display, colors and fill options. Effects are not supported. - Display dynamics via variables for coloring and position dynamics. - Display of variable information *variable value*, **Name**, **Identification**, **Resources label**, **Measuring unit** and **Limit value text** possible. - **Write set value** with dialog. - **Write set value** via element. For details, see the **Write set value** (on page 60) section. - **Note:** The corresponding user level is checked for the execution of **Write set value**.
Numeric value	- Element-specific display options for display, colors and fill options. Effects are not supported. - Display dynamics via variables for coloring and position dynamics. - Write set value with dialog, taking into account static setpoint limits. **Note:** The corresponding user level is checked for the execution of the action.

ELEMENT GROUPS AND SYMBOLS

The following are supported:

Functionality	Support for
Element group	- Display of element groups. - The elements contained are displayed in accordance with their configuration and supported properties.
Linked symbol	- Display of linked symbols. - Support of **Replace linking** when used in a screen. The resulting entries must refer to resources in the local project. - The elements contained are displayed in accordance with their configuration and supported properties.

Functionality	Support for
Symbol properties	- Released symbol properties for all properties supported by the HTML Web Engine. This includes all zenon properties of elements that are supported by the HTML Web Engine. Unsupported properties result in an error message when compiling (on page 67).
Grouping of approved properties	- Groups for approved symbol properties are applied and the properties contained therein are applied.

GLOBAL PROJECT AND LOCAL PROJECT SYMBOLS

Symbols from the following can be exported and used:

- Global project
- Local project

Symbols from the global project are given the prefix **(g)** when compiling.

Note:

If there is a symbol in the local project that has the same name during export as a symbol in the global project, the symbol from the local project will be used. For example:

- Global project: **rotor**
- Local project: **(g)rotor**

During export, **rotor** becomes > **(g)rotor** in the global project. This name corresponds to **(g)rotor** from the local project. Only the symbol from the local project is used.

11.3.2.1 Screens of type Extended Trend

The *Extended Trend* screen type supports the following functionalities for the HTML Web Engine:

OPTIONS WITH THE SCREEN SWITCH FUNCTION

Tab	Group	Settings and notes
Data	**Origin of the data**	Option: **Archive data**
	Options	The **Options** settings are not supported for the HTML Web Engine.
	Curves	Display of simple curves without further options.

Tab	Group	Settings and notes
Display	**Diagram name**	Display of the diagram name, as configured
	Refresh	Options in the **Refresh** properties group are not supported.
	Design	Options in the **Design** properties group are not supported. A pre-defined display is shown.
	Runtime	The display of this dialog is not supported in the HTML Web Engine. Filtering is not possible.
X-Axis	**X-Axis**	Option: - **Only YT display** **Note:** XY display is not supported.
Time	**Filter**	The following are supported: - **Absolute time period** - **Relative time period** All other time filters are not supported. **Attention:** Screen switching with invalid time configurations are not available in the HTML Web Engine. Buttons with corresponding calls are deactivated.
Lots		The settings of the **Lots** tab are not supported for the HTML Web Engine.
Column settings		The settings of the **Column settings** tab are not supported for the HTML Web Engine.
Printer properties		The settings of the **Printer settings** tab are not supported for the HTML Web Engine.

DIAGRAM CONTROL ELEMENT

Group	Subgroup	Settings and notes
Darstellung		The configuration of the display is not supported for the HTML Web Engine. The

Group	Subgroup	Settings and notes
		action is defined and fixed.
Linien		The configuration of the lines is not supported for the HTML Web Engine. The action is defined and fixed.
Füllung		The configuration of fills is not supported for the HTML Web Engine. The action is defined and fixed.

11.3.2.1.1 Data aggregation

Data is aggregated under certain conditions for display in the extended trend. The speed of the display via the HTML Web Engine is thus sped up so that it corresponds to that of zenon Runtime.

Procedure:

- The maximum number of variables to be displayed is determined by the width of the template (in pixels) on which the screen is based.
- There is no aggregation if, during an archive query, fewer variables are returned than pixels are present.
- If the number of the archive variables exceeds the column width, average values are generated.
  To do this, the time axis is divided into time ranges.
  The archive values that are transferred by Runtime are arranged into the respective time ranges. The average value within a time range is calculated using the archive values. If there is no archived value within a time range, this tie range is deleted.
  Note: The generation of the average values is not just carried out for numeric values, but also binary variables.
- Unnecessary variables continue to not be taken into account, because these are not required for the current display. This reduces background calculations, saves computer processing power and speeds up the switching time.
  This happens regardless of the number of archive values and the average value calculation.
  To do this, 3 archived values of a variable that are consecutive in terms of time are analyzed.
  If all 3 variables have the same value, the average variable is removed.
  If, for example, a recorded archive variable has the same value over the queried time period, only 2 variables (start and end) are displayed.

Note: When generating average values, the trend line is generally shown differently to with raw values.

11.3.2.2 Combined element

The combined element supports the following in the HTML web engine:

Group and/or property	Remark
Reference point	
Bitmap settings	
Flashing	Not supported: - **Recognize flashing of all variables**
Representation	Not supported: - **Style group** - **3D** - **Display status of main variable** - **Display status with square** - **Type of display** - **Configuration and test** - **Apply project properties if "locked"**
Fill	Not supported: - **Fill style**
Position	
Switch	The element switches the value of a bit variable.
Visibility	Not supported: - **Interlocking**
Write set value	Numeric values via write set value dialog and function. The linked function is executed as in the zenon Runtime only after closing the dialog to write a set value. Closing is carried out through successful writing by clicking on **OK** or by canceling.

Group and/or property	Remark
WRITE SET VALUE/BINARY VALUE	The **Reset on exit** property is always active. Not supported: - **Keep pushed state** - **Fast reaction** - **Holding time [ms]**
Write set value/Numeric value	Only the *Standard dialog* setting is supported for the **Write set value via** property. Not supported: - **without dialog** - **External program** - **Use screen Keyboard** - **Screen Keyboard**
Write set value/Set value limits static	
Pushbutton and **Pushbutton On**	The **Reset on exit** property is always activated and also cannot be deactivated.
Text	Not supported: - **Text style**
Text chained	
Variable/function	The following are not supported: - **Parameter for substitution** - **Apply from calling screen** - **Function execution on pressing**
Status selection via variable status and text dynamization:	Selection of status based on appropriate variable status, in addition to variable value: Display of status text: - **Text color dynamic** - Text when pressed - **Transparent** - Placeholder is case sensitive: *%n* (Variable name) *%I* (Variable ID)

Group and/or property	Remark
	%v (Variable value) *%u* (Unit

FORMULAS FOR STATUSES

The combined element supports the use of formulas for statuses. All value and status bits for several source variables are supported, as well as the operators available in the zenon Runtime.

Exception: The *unacknowledged* status bit is not supported. It is always treated as *0* in the HTML Web Engine.

11.3.2.3 Write set value

For the **Dynamic Text** and **Numeric value** elements, the *Standard dialog* or *Element* values are available for the **Write set value via** property.

WRITE SET VALUE VIA ELEMENT.

In Runtime, the input function is activated when an element is clicked. If a text or a default text is available, this will be preselected. On mobile devices, the keyboard will be displayed after activation. **Attention:** When using iOS, the element must be selected again in order to display the keyboard.

DIFFERENCES BETWEEN ZENON RUNTIME AND HTML WEB ENGINE

zenon Runtime and HTML Web Engine are displayed differently:

Default in Editor for Write set value via element	zenon Runtime	HTML Web Engine
Defaults: - **Dynamic text** and *String* type variable - **propose current value**: *activated*	No value is shown.	The current value is shown and selected.
Defaults: - **Dynamic text** and *String* type variable - **propose current value**: *deactivated* - **Set value text**: Text available	No value is shown.	The set value text is shown and selected.

Default in Editor for Write set value via element	zenon Runtime	HTML Web Engine
Defaults: - **Dynamic text** and *String* type variable - **propose current value**: *deactivated* - **Set value/change by**: contains value	The current variable value is shown.	The entered set value is shown.
Defaults: - **Hidden input**: *activated*	When entering the value, the letter defined for generic input in the Editor is shown in a text field.	When entering the value, the standard character for generic input is shown in a text field: Dot or star.

11.3.3 Functions

Functions and scripts

Function	Support for
Screen switch	- Calling screens of the local project. - Support for *replace linking*, whereby resulting entries must refer to resources in the local project.
Close frame	- Closing of frames with the given frame name.
Write/modify set value	- Direct writing of pre-defined variable values.
Language switch	- Online switching of **language file** and **font list**.
Switch palette	- Online switching of the **color palette** for graphic display.
Script: execute	- Execution of functions of the local project. Non-supported functions are excluded from execution.
Logout	- Logging a user out of a zenon Web Client session and disconnecting from zenon Runtime - The zenon Web Client session is continued in offline mode
Play audio file	- Plays an audio file (on page 63).

Function	Support for
Start continuous tone	▸ Starts a continuous tone (on page 63).
Stop continuous tone	▸ Stops a continuous tone (on page 63).

11.3.3.1 Automatic script call when starting a zenon Web Client session

A script can be called up automatically when starting an HTML web client session. A special start page can thus be prescribed for the web application, for example.

The following script name is reserved for the script: *AUTOSTART_HTML_WEBCLIENT*

Note: This script is executed whenever a session of an HTML web client starts. The name of the script must not be changed.

11.3.3.2 Individual script call by means of URL expansion when starting a zenon Web Client session

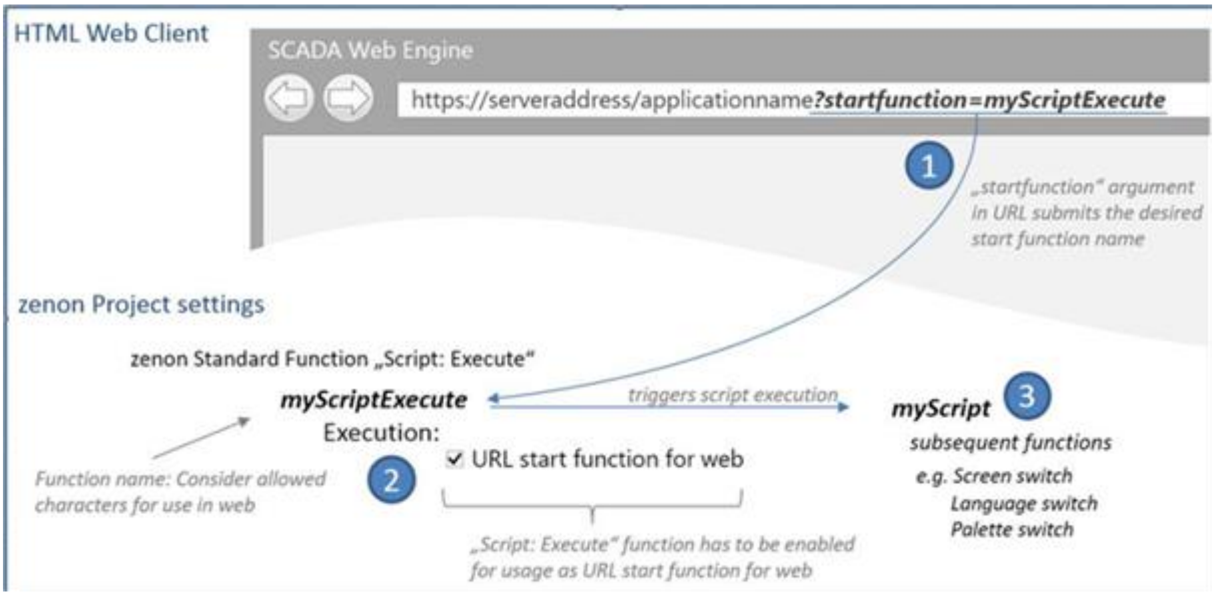
The HTML Web Engine allows the individual execution of zenon functions as part of a session start for an HTML web client. The function is executed by a script that is started using the **Script: execute** function. The name of the desired start function is transferred in the URL for the call to the web page with the **startfunction** argument. The function for the execution of the script must be explicitly approved for call-up as a URL start function.

Note: All settings that are required for the use of the URL start function must be set in the project at the time of compilation for the web.

Process for the application of the functionality:

1. The web page is called up with the additional **startfunction** argument for the call of the desired **Script: execute** function:
   The call is made using the function name.

2. If the function for the use as a URL start function has been approved, the assigned script is executed. The execution of the script is carried out individually for this HTML web client session. If the addressed function does not exist or has not been unlocked, a standard web page call is made. The session is then set up according to the call without **startfunction** argument.


HTML Web Client
SCADA Web Engine
https://serveraddress/applicationname?startfunction=myScriptExecute
1
„startfunction" argument in URL submits the desired start function name
zenon Project settings
zenon Standard Function „Script: Execute"
myScriptExecute
Execution:
triggers script execution
myScript
3
subsequent functions
e.g. Screen switch
Language switch
Palette switch
Function name: Consider allowed characters for use in web
2
☑ URL start function for web
„Script: Execute" function has to be enabled for usage as URL start function for web

Note: This screenshot is only available in English.

The following limitations are applicable for naming the function name in the web:

Permitted	Forbidden
Alphanumeric characters (0-9, a-z, A-Z)	Umlauts
`$ - _ . + ! * ' ( ) ,`	ASCII control characters (**0x00-0x1F;0x7F**)
	Reserved characters (**& / : ; ? @**)
	Unclear characters (such as spaces and **" ' < > # % { } I \ ^ ~ [ ]**)

11.3.4 Play audio signal or continuous tone

Audio signals and continuous tones can be played in the HTML Web Engine.

- Audio signal: An audio file is played once.
  Use the **Play audio file** function to start it.

- Continuous tone: An audio file is played continuously.
  The **Start continuous tone** function is used to start it and the **Stop continuous tone** function is used to stop it.

> ⚠**Attention**
>
> Audio files must be available as ***.wav** files.
>
> These are not supported for:
>
> - Safari browser
> - iPad

EXECUTE FUNCTIONS

The functions for playing audio signals or for starting and stopping a continuous tone can be executed by

- the button linked to the function
- a call in a script that is linked to a button
- a limit value that calls the respective function in the event of a violation

RULES

The following rules apply for playing audio signals and continuous tones in the HTML Web Engine:

- Only one audio file can be played at a time.
  Starting an audio signal or a continuous tone may interrupt a different audio file that is already playing.
- Playback length:
  - Audio signal: The audio file determines the length of the audio signal.
  - Continuous tone: The call of the **Start continuous tone** and **Stop continuous tone** functions determines the length of the tone.
- Several functions can be created and called for different audio signals and continuous tones. To do this, each one must be linked to a separate button.
- If several audio files are gathered in a script, only the last audio file will be played.
- Switching between screens does not interrupt the playing of an audio file.

11.3.5 Variables

The following variables are assigned in the HTML Web Engine:

- All variables of the current screen.
  When switching screens, the variables of the previous screen are unadvised and the variables of the new screen are advised.
- **Permanently-monitored variables**:
  Variables that call a function when their value is changed. They remain permanently assigned.
  For these, the following applies:
  - The variable must have at least one limit value.
  - At least one limit value has a linked function that is supported by the HTML Web Engine.

> ### ⚠Attention
>
> **Permanently-monitored variables** are not identical with the variables for which the **Permanently read variable** property was activated in the zenon.
>
> In zenon, this property causes the variable to be assigned to the driver and all changes on the PLC are always reported to the zenon.
>
> In the HTML Web Engine, a decision is made automatically whether a variable must be continuously read by Runtime. For example, if an audio signal is to be played automatically in the event of a limit value violation.

BEHAVIOR OF PERMANENTLY MONITORED VARIABLES

To keep the number of connections for reading **Permanently-monitored variables** to a minimum, variables behave as follows:

- All variables of the screen are linked by each individual HTML Web Engine session.
- All **Permanently-monitored variables** are only linked once, and all HTML Web Engine sessions use this same link. This ensures that changes to **Permanently-monitored variables** values need not be transferred individually for each session.

Note:

- **Permanently-monitored variables** are linked independently of the variables of the current screen.
  In this way, variables can be double-linked. This has no effect at the current time. This is because changes to **Permanently-monitored variables** values only cause the function to be called in the event of a limit value violation. This does not occur for changes made to the values of screen variables.
- Switching between different screens causes the variables of the previous screen to be unadvised and those of the new screen to be linked. **Permanently-monitored variables** are not affected by this.

INITIAL LINK

The initial link of **Permanently-monitored variables** is established by the user who initially sets up an HTML Web Engine session. This link is maintained until all HTML Web Engine sessions that use this link are closed. That means: HTML Web Engine sessions added subsequently can continue to use **Permanently-monitored variables** even if the user for whose HTML Web Engine session the link to **Permanently-monitored variables** was set up has already logged out.

DISCONNECTION AND RECONNECTION:

In the event of disconnection, all variables are unadvised in all HTML Web Engine sessions.

If the connection is reestablished, **Permanently-monitored variables** is not available until the user has logged on again.

FUNCTION CALL DUE TO LIMIT VALUE VIOLATION

The HTML Web Engine checks for **Permanently-monitored variables** whether the limit value for this variable has been violated. If a limit value violation is found, the function saved in the limit value is executed.

In doing so, the following applies:

- The function is executed regardless of whether the variable is available in the screen.
- The function is executed for all logged in users.
- The function must be supported by the HTML Web Engine.
- If two limit values have the same value, then there is no rule about which one will be executed. The sequence must be taken into account when using the Web Engine Compiler.

Definition of limit value violation for HTML Web Engine:

- The **Limit value active** property must be active for the variable.
- The following applies if the **Minimum/Maximum** property has been configured as *Maximum*:
  - The limit values are sorted in descending order.
  - The first limit value that is less or equal to the current variable value is used.
  - The value for **Limit Value** must be greater than the existing variable value.
- The following applies if the **Minimum/Maximum** property has been configured as *Minimum*:
  - The limit values are sorted in ascending order.
  - The first limit value that is greater or equal to the current variable value is used.
  - The **Limit Value** value must be less than the existing variable value.

12 Compile project for web

With the **Web Engine Compiler**, the data that the **HTML Web Engine** needs to provide HTML5 content for the zenon Web Client is provided from a zenon project. When translating this project data, the **Web Engine Compiler** compiler checks the project contents and provides information on non-supported functions or properties. As a result of the translation process, a file is created that is provided to the **Web Engine**.

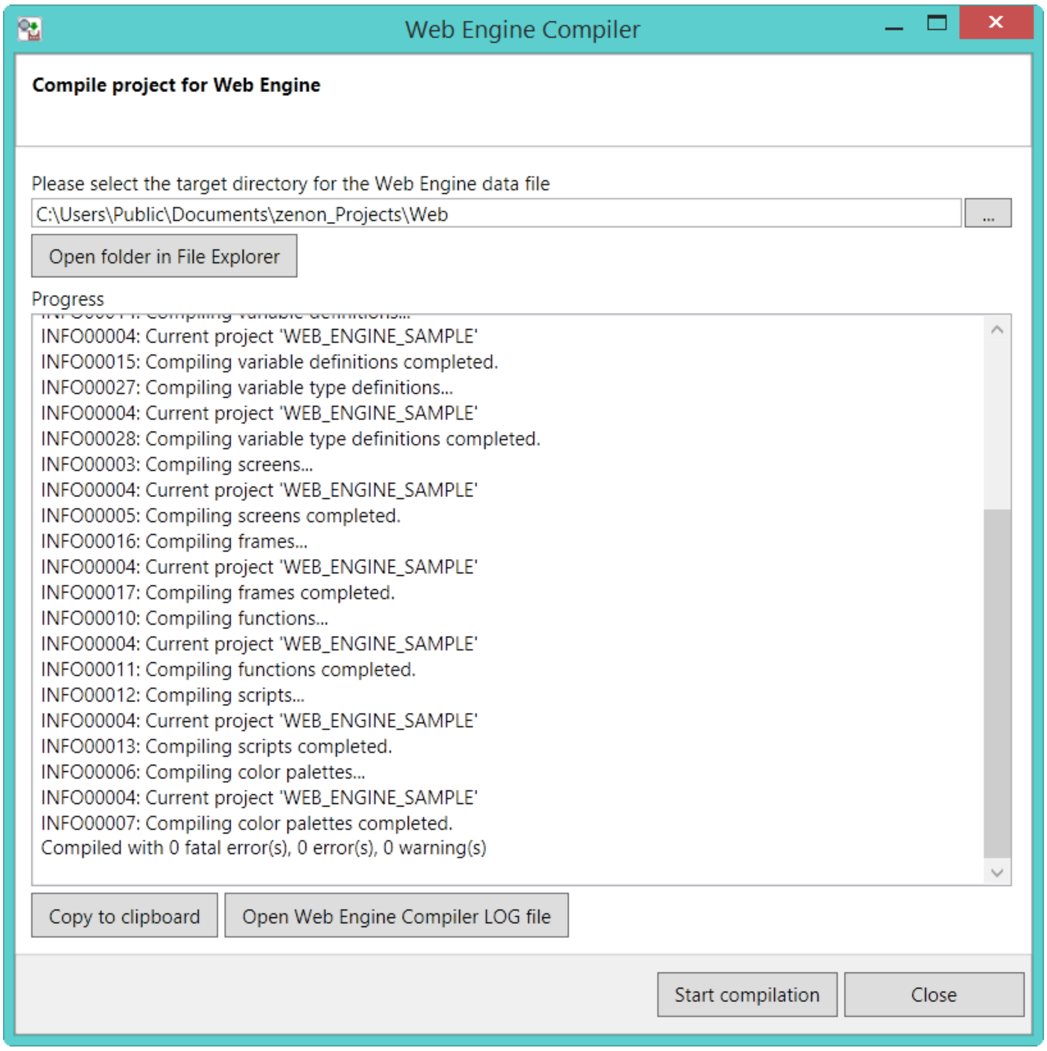
To open the **Web Engine Compiler**:

1. Click on **Extras** in the menu bar of the Editor.
2. Click on **Compile project for web...**

   The dialog to compile the project is opened.
   Note: This is only available in English.

> ### ⚠Attention
>
> If the **Web Engine Compiler** is executed several times for different zenon projects, several Web Engine project files are also saved in the output folder. In this case, the HTML Web Engine always uses only the project file that comes first in the alphabet.

HTML WEB ENGINE COMPILER DIALOG.



Web Engine Compiler
Compile project for Web Engine
...
Open folder in File Explorer
Progress
Copy to clipboard
Open Web Engine Compiler LOG file
Start compilation
Close

Please select the target directory for the Web Engine data file

C:\Users\Public\Documents\zenon_Projects\Web

```
INFO00004: Current project 'WEB_ENGINE_SAMPLE'
INFO00015: Compiling variable definitions completed.
INFO00027: Compiling variable type definitions...
INFO00004: Current project 'WEB_ENGINE_SAMPLE'
INFO00028: Compiling variable type definitions completed.
INFO00003: Compiling screens...
INFO00004: Current project 'WEB_ENGINE_SAMPLE'
INFO00005: Compiling screens completed.
INFO00016: Compiling frames...
INFO00004: Current project 'WEB_ENGINE_SAMPLE'
INFO00017: Compiling frames completed.
INFO00010: Compiling functions...
INFO00004: Current project 'WEB_ENGINE_SAMPLE'
INFO00011: Compiling functions completed.
INFO00012: Compiling scripts...
INFO00004: Current project 'WEB_ENGINE_SAMPLE'
INFO00013: Compiling scripts completed.
INFO00006: Compiling color palettes...
INFO00004: Current project 'WEB_ENGINE_SAMPLE'
INFO00007: Compiling color palettes completed.
Compiled with 0 fatal error(s), 0 error(s), 0 warning(s)
```

Option	Description
Please select the target directory for the Web Engine data file	Select a folder to save the **Web Engine** file here. Click on the **...** button to open the dialog to select a folder.
Open folder in File Explorer	Opens the selected folder in the Windows Explorer.
Progress	Shows warnings, error messages and information during compilation.
Copy to clipboard	Copies the content of the **Progress** output window to the clipboard.
Open Web Engine Compiler LOG file	Opens the LOG file for the **Web Engine Compiler**. This contains the messages from the **Progress** output window.

Option	Description
Start compilation	Starts the compilation process.
Close	Closes the **Web Engine Compiler.**

13 Process of an HTML web engine session

The HTML visualization is available after a successful compilation of the project data.
To call up the web page:

1. Open an HTML 5-compatible web browser.
   You can find the list of recommended web browsers in the System requirements (on page 12) chapter.
2. Enter the web site URL into the address bar of the web browser for the HTML5 visualization: **https://Serveradresse/Applikationsname**.

   The HTML 5 content is provided automatically. In doing so, a separate session is created and administered for each zenon Web Client. The runtime data of zenon Runtime is available as soon as you have been successfully authenticated as a user. You can read more details about this in the **Client authentication for a connection to Runtime** (on page 25).
3. As soon as you leave the web page, the HTML Web Engine session and the connection to zenon Runtime is disconnected automatically.

Note: The web page is left when the web browser is closed, its tab is closed, view is updated or the URL is entered again (among other things).

RESTART WEB ENGINE SESSION

A dialog will display when it is necessary to restart the Web Engine session. The restart can be automatic or manual. By default, the dialog restarts the session independently after 30 seconds. In doing so, the remaining time until the restart is shown.
This time period can be configured individually.

To configure the time period until restarting:

1. Open the **appsettings.json** with a text editor.
   You can find this in the folder in which the Web Engine has been published.
   Default: **C:/inetpub/wwwroot/<Name Web Engine>**
2. Navigate to the **AutomaticSessionRestartDelay** entry.
3. Enter the desired delay in seconds.

- Minimum: *0*
  With negative values, *0* or a missing entry, there is no automatic restart.
  The restart must then be carried out manually by clicking on the **Restart** button.
- Maximum: *3600* (= *1 h*)

Default: *30*

14 System diagnosis and troubleshooting

If there are problems during a system start or during operation, error messages that provide information on the possible cause of the problem are given in the HTML web client. The logging stage of the error messages depends on how the HTML5 web page is called up:

- Local web browser: You receive detailed messages.
  To do this, the web browser must be on the same computer (with the same IP address) on which the Web Engine is being operated.
- Remote browser: General messages.
  This is applicable when being called up from a different device and/or a different IP address.

CHECKLISTS

FOR ERROR-FREE SYSTEM OPERATION

The following checks are recommended for general checking of the system configuration:

- HTML Web Engine has been installed on the IIS.
  - The web engine deployment was carried out without any errors.
  - The web server is in operation.
- Visualization data is generated with the **Web Engine Compiler**.
  - There are no errors during the compilation process.
  - The resultant data of the **Web Engine Compiler** is ready for access by the Web Engine.
    Note: The occurrence of warnings does not influence the ability of the HTML Web Engine to run in principle. However, there can be limitations to the configured functionality depending on the type of warning
- The versions of **Web Engine Compiler** and HTML Web Engine are identical.
  Note: The data created by the **Web Engine Compiler** can only be interpreted correctly by the HTML Web Engine (web application) with the same version number.

FOR THE TRANSFER OF PROCESS DATA

Please note when transferring process data:

- zenon Runtime and the configured connector have been started. The connector can be reached via the network.
  Note: The processes for zenon Runtime and **SCADA Runtime Connector** must run in the same user context.
- The Users who need to be authenticated must be available in zenon Runtime.