zenon manual

Release notes zenon 8.00

v.8.00
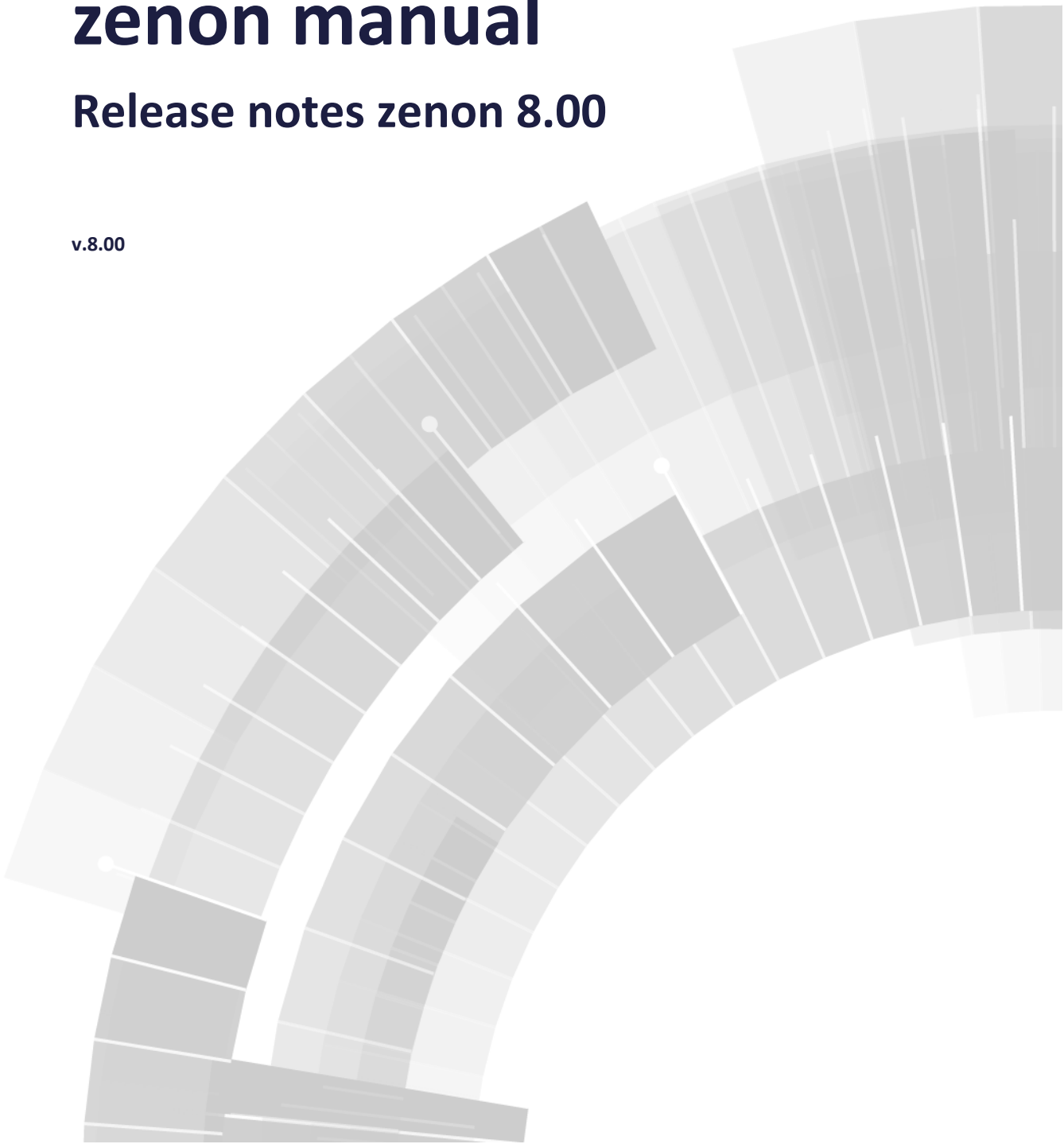
Contents

1. Welcome to COPA-DATA help

ZENON VIDEO-TUTORIALS

You can find practical examples for project configuration with zenon in our YouTube channel. 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 Support Team, who 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. General

2.1 64-bit time stamp for internal time stamp [μs] (S 51288)

The internal time stamp of variables supports the full resolution of microseconds (μs).

2.2   Autostart Connector Container (F 66286)

The connector container can be started automatically using the **Startup Tool** if a user logs on at the computer. Autostart is activated via **Application -> Options** in the **Startup** tab.

2.3   Converting projects

Before you convert a project, please read back all Runtime changeable files (User Administration, Standard Recipes, Recipegroup Manager, Scheduler/PFS) into the Editor. This ensures a complete data conversion and makes sure that none of the changes made in the Runtime are lost. After converting to the new version, create all Runtime files once including Runtime changeable data.

Note: You can find important information for the conversion of certain versions in the zenon help in the Project conversion manual.

CONVERTING MULTI-USER PROJECTS

Multi-user projects can only be converted if no elements are checked out. This means that all people configuring projects have to **accept** their changes first.

If this is not possible for some reason, you have to create a project backup of the project on the project database server and then immediately restore it. This resets all the **Under construction** information. Attention: All changes in the local project versions are lost!

The conversion can only be done on the PC, on which the central project database resides. If there is no Editor on the PC (standalone database server – no longer supported), you must install the Editor first. Only after that can the conversion be done on this PC.

CONVERSION FROM VERSION 6.01 AND 6.20

zenon projects in version 6.01 and 6.20 can no longer be directly read back in zenon 7.10 or higher.

Background: Versions that are based on the MSDE (SQL Server 2000) are not compatible with the SQL Server 2012 used in zenon.

Solution: First convert in zenon 7.0 and then in 7.10 or higher.

CONVERSION OF PROJECTS FOR 7.20

For compatibility with version 7.20, there is an additional possible selection - "`Most recent version`" - available for the Create Runtime files property. This can be selected with the `7.20 SP0 + [most recent build no.]` in the drop-down list.

With this selection, the Runtime files for the current build of version 7.20 are provided. Functionality that has since been incorporated into version 7.20 after the official release of 7.20 is thus supported. This is applicable most of all for enhancements to drivers that are now supported with this option. Please note that, to use the most recent build, you must have zenon 7.20 installed on your computer.

Note: The 7.20 SP0 selection compiles the Runtime files - as before - to the default settings of 7.20 SP0.

2.4   Supported operating systems

2.4.1 Desktop operating systems

Supported desktop operating systems and required service packs:

Operating system	zenon Editor	zenon Runtime	zenon Web Server	zenon Web Client	zenon HTML Web Engine	zenon Logic Runtime	zenon Analyzer Server
Windows 7 (Professional, Enterprise and Ultimate version, x86 and x64 versions).	SP 1	SP 1	SP 1	SP 1	SP 1	SP 1	Cannot run
Windows Embedded Standard 7 (if all necessary operating system components exist).	Cannot run	SP 1	SP 1	SP 1	SP 1	SP 1	Cannot run
Windows 8 and 8.1 (Standard, Professional, Enterprise version, x86 and x64 versions)	SP 0	SP 0	SP 0	SP 0	SP 0	SP 0	Only x64 with SP 0
Windows Embedded 8 Standard (if all necessary operating system components exist).	Cannot run	SP 0	SP 0	SP 0	SP 0	SP 0	Cannot run
Windows 10 (Home, Pro, Enterprise, Education, Pro Education, Enterprise LTSB, IoT Enterprise, Pro for Workstations)	SP 0	SP 0	SP 0	SP 0	SP 0	SP 0	Only x64 version of Home, Pro and Enterprise with SP 0.

2.4.2   Server operating system

Supported server operating systems and required service packs:

Server operating system	zenon Editor	zenon Runtime	zenon Web Server	zenon Web Client	zenon HTML Web Engine	zenon Logic Runtime	zenon Analyzer Server
Windows Server 2008 R2 (All editions with the exception of Core)	SP 1	SP 1	SP 1	SP 1	SP 1	SP 1	Cannot run
Windows Server 2012 and 2012 R2 (All editions with the exception of Core)	SP 0	SP 0	SP 0	SP 0	SP 0	SP 0	Only x64 with SP 0
Windows Server 2016 (All editions with the exception of Core)	SP 0	SP 0	SP 0	SP 0	SP 0	SP 0	Only x64 with SP 0

2.4.3 Microsoft SQL Server 2012 SP4

The **Microsoft SQL Server 2012** is now installed with **SP4** by default.

2.5 Windows updates impair zenon functionality

Due to security vulnerabilities with Intel and AMD processors (known as Spectre and Meltdown), Microsoft and other manufacturers of hardware and software released security updates at the start of 2018.

You can find the complete list of the affected products and operating systems here: https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/ADV180002 (https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/ADV180002)

⚠ **Attention**

All Information is correct as at the time of editorial conclusion of the release notes on February 19, 2018.

This primarily concerns the Microsoft KBs stated.

ZENON VERSIONS

Problems were detected in the following zenon product components:

▶   zenon 8.00 (Beta)

▶   zenon 7.60

▶   zenon 7.50

▶   zenon 7.20

▶   zenon 7.11

Note: Older zenon that have already been discontinued may be affected.

DETECTION

The most frequent symptoms of problems when applying the Microsoft security updates:

▶   Error message when starting zenon

▶   Drivers do not run correctly

- ▶ Missing values in a process screen
- ▶ Gaps in the Historian data logging
- ▶ Missing alarms and events
- ▶ Data not available and connection problems with zenon clients in the network
- ▶ Effects on zenon Remote Transport

RECTIFICATION OF PROBLEMS

Microsoft has replaced the KBs that cause problems and replaced them with working KBs.

To solve problems brought about by the updates:

- ▶ Avoid applying defective updates.
- ▶ If defective updates have been applied, remove them.
- ▶ Apply the working updates.

LIST OF KBS:

List of KBs that cause problems or rectify them:

KB causes problem	KB rectifies problem	Version
KB 4056898	KB 4057401	▶ Windows 8.1 ▶ Windows 8.1 (x64) ▶ Windows Server 2012 R2 Standard
KB 4056896	KB 4057402	▶ Windows Server 2012 Standard
KB 4056895	KB 4057401	▶ Windows 8.1 (x64) ▶ Windows Server 2012 R2 Standard
KB 4056893	KB 4075199	▶ Windows 10 Version Enterprise
KB 4056892	KB 4073291	▶ Windows 10 Version 1709
KB 4056891	KB 4057144	▶ Windows 10 Version 1703 (x64)
KB 4056890	KB 4057142	▶ Windows Server 2016 ▶ Windows 10 1607
KB 4056888	KB 4075200	▶ Windows 10 Version 1511

Last updated on: 20. 02. 2018.

2.6　Performance

Further performance improvements have been implemented for version 8.00.

2.6.1　Buffer for seamless redundancy (S 41498)

The standby buffer necessary for seamless redundancy has been revised. The saving of data has been optimized in this buffer.

Optimizations for the standby buffer in redundant networks:

▸ The filling and processing of the buffer in the event of redundancy switching has been revised.

- Buffer values are processed on the standby server directly.
  In doing so, these buffer values are received directly by the standby server's driver.

- No data sync is carried out in the event of a primary server failure. This is no longer necessary with the new system. The values are taken from the standby server's buffer, which it also maintains, in the event that it becomes the primary server.

▸ The network load for the data sync in the event of redundancy switching has been heavily reduced.

2.6.2　Runtime (F 49826)

The start of zenon Runtime has been sped up. In addition, the start of Runtime was adapted and optimized for parallel driver registration.

2.6.3　Optimization for cyclical archives - use of all available processors (F 49807)

The recording and of cyclically-recorded archives has been revised. Cyclical archives now use all available processors of a computer. Lot archives of a cyclical archive have also been optimized.

In addition, the die logging of the archiving has been enhanced with further LOG messages.

2.6.4   BACnetNG Driver (S 73759)

REQUEST FOR SEVERAL OBJECTS POSSIBLE (S 73759)

`present-value` and `status-flags` are read for several objects in a **Read-Property-Multiple Request**. In doing so, the driver attempts to use the configured **Max. APDU** size.

These **Object-Properties** were previously used individually for each **Object**.

OBJECT_LIST AND OBJECT_NAMES

The attendant objects are initially determined with a `read-Request` to **Object_List-Property** and **Property list** . A list of IDs that can be resolved into names is received from the **BACnet Device** as a response.

UNSUBSCRIBE COV SUBSCRIPTION IN THE EVENT OF UNADVISE

- ▶ `COV Subscriptions` Are advised if variables are inactive and when the driver is closed.

- ▶ If a variable is configured as **COV**, but is not contained in the notification, the INVALID bit is set and a corresponding LOG message is created.

- ▶ If no `Notification` is received after a `Subscription` from the **BACnet Device** (APDU Timeout x (Retries +1)), the INVALID bit is set for the variable and a corresponding LOG message is created.

3. Licensing (F53496, 53827, 71759, 71763, 71688)

In order to be able to use COPA-DATA products, they must be licensed using a serial number. A product runs in demo mode until it has been licensed.

The licensing of COPA-DATA products has been reimplemented for version 8.0. Licenses for previous versions are no longer valid as of version 8.00. Only CodeMeter dongles are used. WibuKey dongles can no longer be used.

LICENSING NEW

Licenses can be used with:

- ▶ Hardware dongles

- ▶ Software dongles (computer-bound)

- ▶ VM dongles

Note: Licenses for physical computers cannot be used for VMs.

For new licensing the following is especially true:

- ▶ Licenses are activated and managed (on page 22) in the new **COPA-DATA License Administration**.

- ▶ The version check is always carried out for the products and versions mentioned in the license. Licenses are only valid for these versions. If there is a maintenance contract, licenses are valid for all versions covered by the maintenance contract.

- ▶ If several licenses are available, you can define the order of their use. If a license becomes invalid, an automatic change to the next fitting license is carried out.

- ▶ License can be bought for different products with different expansions each.
  This applies for: zenon Editor, zenon Runtime, zenon Analyzer, zenon Logic Runtime, zenon web server, zenon Web Engine and Process Gateway.
  For all products limited demo licenses are also available.

- ▶ Several licenses can be provided in a file and can be used via collective activation. Collective activation is possible via:

  - • **COPA-DATA License Administration**: Single activation from a list.

  - • Command Line Tool: Activation of several licenses from a list at the same time.

- ▶ Licenses can be given back and then transferred to other computers.

- ▶ Drivers can now be licensed according to different models:

  - • Individual drivers

  - • Price groups

3.1 License.ini (F 53496)

Licenses and certain settings for the **COPA-DATA License Administration** are administered in the new file called License.ini. Addition, activation and editing of licenses is carried out by means of **license management**.
Path to **Lizense.ini**: `%programdata%\COPA-DATA\`

License.ini contains a separate line with the serial number for each licensed product. If several licenses have been acquired for a product, each serial number is shown in a separate line. The serial numbers are checked row-by-row when the licensed products are queried.

Note: Do not edit **License.ini** manually!
Changes to the file can lead to licensed products no longer being able to be used. Carry out all licensing actions using the license manager.

3.2 License Manager (F 53827, 71759)

The new license management makes it possible to obtain, administer, update and activate licenses.

Licenses can be obtained as a software dongle or hardware dongle, each either locally or in the network. The licensing of products for use on VMs is possible. The licensed modules and the structure of these is shown in the license information for the products.

4. Editor

4.1 Default zoom settings (F 50077)

An initial zoom factor for screens and symbols can be set in the editor under **Window**, **Settings**, **Screen editing** in the **Default zoom setting** drop-down list.

4.2 Start page (F 68022)

The start page is shown in the screen area by default and makes it possible to read and create project content. These always relate to the project that is currently set as active in the Editor. The start page can be turned on or off manually.

4.3 Improved zoom possibilities (F 50077, F 12433)

The possible zoom factor has been increased to 15 - 1600%.

A zoom factor is automatically selected by clicking on **Fit**, in order to be able to show the complete screen in the editor, in proportion.

Keyboard shortcuts have been created for simple use of the zoom function.

Changes to the size can also be carried out with the mouse.

5. Automatic Line Coloring - Standard

The **Automatic Line Coloring** - **ALC** module has been comprehensively enhanced and supplemented with the new **Load Flow Calculation** and **State Estimator** modules from the **Energy Edition**. You can find additional information in relation to this in these release notes in the Automatic Line Coloring - Topology (on page 25) chapter.

The following enhancements have been implemented for the standard applications with ALC:

5.1 ALC documentation revised

The documentation for the **Automatic Line Coloring** module has been revised.

5.2 ALC aliases enhanced (F 62752)

The ALC aliases have been enhanced with the following possibilities:

- ▶ With the **Screen switch** function in the **Replace Indices** tab, **ALC aliases** are also supported in addition to variables and functions.

- ▶ With screen elements that use an ALC alias, the **Use alias** property can be configured with direct input.

  Before zenon 8.00, input was possible by selecting an existing screen and its list of ALC elements.

5.3 ALC link can be configured as STRING (S 65653)

With the zenon `combined element` screen element, links can now be configured as a STRING for **Automatic Line Coloring**. Before zenon 8.00, identification was only possible with numbers.

The property has been renamed accordingly: **Link name**. Existing project configurations are retained during conversion.

6. Energy Edition

6.1 Command Processing (F 73121)

New **topological interlockings** have been provided, which are based on the enhancements for the **Load Flow Calculation** module. The **ALC configuration** project property and the **command processing** module have been enhanced with the following topological interlockings:

- ▶ **Interconnect grids**

  This interlocking prevents unintended connection of two networks with different generator sources. In doing so, the IDs of the sources that are switched on are compared to one another.

- ▶ **Line overload**

  This interlocking prevents the execution of switching actions that cause a power overload on lines. The calculation of this overload is carried out in the **Load Flow Calculation** module.

- ▶ **Interconnect various voltage levels**

  This interlocking prevents unintended connection of two networks with different nominal voltages. To do this, the values of the **Voltage [kV]** option configured for an ALC source are compared to one another.

STATUS BIT M1 AND TIMEOUT ARE NOW ALSO SET AUTOMATICALLY FOR COMMAND VARIABLES

The status `M1` (locked with the `lock` command) and runtime exceeded `[TIMEOUT]` are now also written to the command variable. As a result, this status can be detected by an `auto/remote command` action type. The new **Write status bits to command variables** property has been provided for this when configuring a command group.

With the property activated:

- ▶ In the event of a lock, the `M1` is set in the response variable and the command variable.
  With the property deactivated, the status bit is only set for the response variable.

- ▶ If the `timeout` is exceeded during runtime monitoring, the `[TIMEOUT]` status bit is set in the response variable and the command variable.
  With the property deactivated, the status is only set for the response variable.

WATCHDOG TIMER FOR SET VALUE INPUT ACTION TYPE

The **watchdog timer** now also supports the **set value input** action type. The status bits `COTx` of the command variable are evaluated for this. The value of the response variable is not monitored.

6.2 Automatic Line Coloring - Topology (F 49541)

The **Automatic Line Coloring** - **ALC** module has been comprehensively enhanced, primarily for the **Load Flow Calculation** and **State Estimator** modules. Enhancements for `Alias` and `Link` have already been made for standard applications with ALC.

The ALC aliases have been enhanced with the following possibilities:

▶ With the **Screen switch** function in the **Replace Indices** tab, **ALC aliases** are also supported in addition to variables and functions.

▶ With screen elements that use an ALC alias, the **Use alias** property can be configured with direct input.

Before zenon 8.00, input was possible by selecting an existing screen and its list of ALC elements.

The ALC link: With the zenon `combined element` screen element, links can now be configured as a STRING for **Automatic Line Coloring**. Before zenon 8.00, identification was only possible with numbers.

▶ The property has been renamed accordingly: **Link name**. Existing project configurations are retained during conversion.

The documentation for the **ALC** module has also been revised.

The following enhancements have been developed for topological networks:

6.2.1 Curb for line elements (S 49544, S 50209)

The ALC line elements `Line`, `Polyline`, `Leitung` can be configured with a curb for the visualization of a ground fault or short circuit.

▶ The configuration is implemented in the editor with the new **Use curb** property and **Curb width [px]**.

▶ The display in Runtime is visualized with the configured source color for `ground fault` or `short circuit`.

6.2.2 Topological interlocking: Voltage to ground beyond transformer (F 49546)

The **Voltage to ground** topological interlocking is now taken into account beyond a `transformer`.

Example: The situation of one side of a transformer being grounded and the second side being supplied leads to the interlocking being active.

6.2.3   ALC for Load flow calculation and State Estimator

API - New Events for the ALCEngine Interface (S 62882)

3 events have been added to the zenon API **ALCEngine** Interface.

These are always executed if the following is applicable in zenon:

- ► `Updated`
  An **ALCEngine** calculation has been completed.

- ► `LoadflowCalculated`
  The calculation of the statistical load flow analysis - in the **Load Flow Calculation** module - has been completed.
  Note: A load flow analysis of the topological interlocking check or the N-1 analysis of the **State Estimator** do not trigger this event.

- ► `NMinus1Calculated`
  The N-1 calculation - in the **State Estimator** module - has been completed.
  Note: An N-1 analysis of the topological interlocking check does not trigger this event.

Configurable nominal voltage for ALC source (S 59122, S 61163)

It is possible to set a nominal voltage for each configured source (not for system sources) in the ALC **ALC configuration** project property.

To this end, the **ALC configuration** dialog was supplemented with the **Voltage [kV]** column. This property of the sources is taken into account for the **load flow calculation** and for the **topological interlockings** module.

The property is taken into account when the Runtime files are created in the zenon Editor. A corresponding error message is shown in the output window if ALC elements (`source`, `generator`, `transformer`) that are in the same network segment have been linked to sources with different voltages.

New topological interlockings (S 49547, S 54543, S 61156, S 59124)

The **ALC configuration** project property and the **command processing** module have been enhanced with the following topological interlockings:

- ► **Interconnect grids**

  This interlocking prevents unintended connection of two networks with different generator sources. In doing so, the IDs of the sources that are switched on are compared to one another.

- ► **Line overload**

This interlocking prevents the execution of switching actions that cause a power overload on lines. The calculation of this overload is carried out in the **Load Flow Calculation** module.

▶ **Interconnect various voltage levels**

This interlocking prevents unintended connection of two networks with different nominal voltages. To do this, the values of the **Voltage [kV]** option configured for an ALC source are compared to one another.

Screen elements - new properties for ALC (F 49540, F 49545, F61164)

With the zenon `combined element` screen element, several new properties have been added for **Automatic Line Coloring**. The new properties have been grouped into the following areas:

▶ **Load flow calculation - input** (S 51765)

The input parameters for the **load flow calculation:** static input or via the linked variables with current measured values: **Active power [MW], reactive power [MVar]** and **Voltage [kV]**.

▶ **Load flow calculation output** (S 51788, S 51790, S 73991)

The output from **load flow calculation**: Reactive power, phase and voltage.

▶ **Condition** (S 49545)

The **Load Flow Calculation** can write the information about number of the current sources to linked values in ALC sources - at the input or output of the element.

▶ **Load flow calculation transformer input** (S 54537, S 56917, S 59129, S 73991)

The input parameters for the **load flow calculation** via the process-technical properties of a transformer. Two-winding and three-winding transformers are supported. Reverse feed is also taken into account.

▶ **Load flow calculation transformer output** (S 59129, S 69008, S 69011)

Output from **load flow calculation** via the transformer.

▶ **Capacitor** (S 56917, S 59134)

The input parameters for the **load flow calculation** via the process-technical properties of a condenser.

New properties were also added for **ALC** for screen elements for the lines (`line`, `polyline` etc.). These are summarized in the following property groups:

▶ **Load flow line parameter** (S 51765, S 54537)

▶ **Load flow line result** (S 56916, S 62873)

6.3   Load flow calculation (F 49540, F 54542, F 54544)

The **Load Flow Calculation** module implements the following functionality:

▶ Derivation of the load flow model from screens with ALC elements (active elements, closed switches etc.)

▶ Calculation of the load flow for the current status (from the values of the ALC elements).

▶ Topological interlockings, based on advance calculation of the ALC model.

▶ (N-1) calculation.

The configuration is carried out in the zenon Editor by setting the parameters of ALC properties for the corresponding screen elements (`combined elements`, `line`, ...). For these configurations of the load flow calculations, the properties of the corresponding ALC screen elements have been considerably enhanced in the zenon Editor.

In zenon Runtime, the calculation is carried out on the basis of the Newton-Raphson method for iterative and approximative solution of non-linear equation systems. The problem is set with complex values: applicable for N bars, of which G with generators, is 2N - G - 1 real unknown (voltage on the load bars, phase of the bars). The nominal voltage without phase moving is assumed as a starting value.

The results are transferred in the properties of the screen elements with activated ALC linked variables, as on a new "`load flow (n-1) calculation`" screen.

SCREEN TYPE: LOAD FLOW (N-1) CALCULATION

The new `load flow (n-1) calculation` screen type visualizes the calculated "N-1" scenario in Runtime, for example a possible network overload in the event of a failure of a line.

6.4   State Estimator (F 61164)

The **State Estimator** module is an additional module to the **Load Flow Calculation** module.

The **State Estimator** module determines approximated values for voltages and phases if the known measured values for a load flow calculation are not sufficient. To do this, it uses the values of measuring points on lines.

▶ The values of the measured values are configured in the properties of the ALC elements.

▶ The result of the **State Estimator** is given immediately, like the result of a load flow calculation. This result is also used for a topological interlocking check, as well as the (n-1) calculation.

Starting from a given Jacobian matrix of the **Load Flow Calculation** module, the voltages and phase differences of the individual busbars are calculated.

These calculated values are compared to the measured values. The calculation is repeated recursively until the precision required for the **State Estimator** has been achieved.

7. Runtime

7.1 Automatic creation of curves in the ETM when using the lasso function (F 50127)

With the lasso function, you can now also, in Runtime, insert variables of dynamic elements into the diagram or the curve list of Extended Trend.

The curves are automatically created in the Extended Trend for the added variables.

Note: The functionality cannot be deactivated. The VBA macros are also executed, which must be noted during project configuration.

7.2 Display of multi-line texts in lists (F 66280)

Longer texts over several lines can now also be displayed in Runtime if the necessary configuration steps have been carried out beforehand in the Editor.

To do this, activate the checkbox of the **Automatic word wrap** property in the respective list in the **Representation**.

Note: The line height for multi-line output must be amended manually.

7.3 Multi-Touch functionality for lists (F5538)

Multi-Touch functionality is also available in Runtime for some lists.

7.4 System driver (S 53890)

The following system driver variables have been removed from the list of configurable system driver variables:

- ▶ **[Project information] driver queue overflow**

▶ **[Project information] driver queue overflow name**

These variables are retained for existing project configurations from previous versions. However they are no longer supplied with values from version 8.00.

> 💡 **Information**
>
> Due to the optimizations to the connection to the zenon, there can no longer be a queue overflow.

7.5 Optimized Runtime start and reload behavior (F 49803, 49800)

The starting and loading behavior for projects in zenon Runtime has been further optimized:

▶ The reload time for changes to configurations has been optimized.

▶ For the distribution of values and signing-in of variables of a project, all available processor `Cores` of the computer executing the process are used.

▶ The advising of the variables on the driver (`advise`) is now only carried out if the variable is needed for display in Runtime.
A general advising of all variables when starting Runtime or when reloading a project is no longer the case.

▶ Variables that are no longer needed are unadvised with increased time criticality (`unadvise`). Free memory space can thus be gained for ongoing operation.

▶ All optimizations are also applicable for cross-project variables.

> 💡 **Information**
>
> These optimizations also cause an improvement in performance in the zenon network. In addition to reduced network load, the server and clients are also available more quickly.

7.6 Start of the SCADA Runtime connectors with Startup Tool (F66276)

The **SCADA Runtime Connector** can now be started manually using the **Startup Tool**. You can find the **SCADA Runtime Connector** under **Tools** and **Available applications (current folder)**.

7.7 Performance optimization (F 49826)

The start of zenon Runtime has been sped up. In addition, the start of Runtime was adapted and optimized for parallel driver registration.

8. Screens

8.1 Undo actions (F 12433)

In the Editor, under **Extras**, **Settings**, **Screen editing** under **Undo steps**, up to 250 actions can be undone.

Default: 100 actions

8.2 (N-1) calculation - new zenon screen type

The new `load flow (n-1) calculation` screen type visualizes the calculated (N-1) scenario in Runtime, for example a possible overload of the network in the event of a failure of a line.

8.3 Create screen with screen creation dialog

In the new **New screen** dialog, you create the necessary selection possibilities for the creation of a screen in a dialog.

The following options can be selected manually:

- ▶ **Name**
- ▶ **Frame**
- ▶ **Screen type**
- ▶ **Template**

In addition, you can create an equipment group and activate the **"Screen active" - create and link variable** checkbox.

8.4 Screen elements

8.4.1 Screen elements - new change of the properties for ALC (S 51765)

The grouping of the properties for the screen element for the **Automatic Line Coloring** module have been restructured:

▸ The ALC project properties group **Fault location from protection/load distribution** has been renamed to **Topological properties**.

▸ From this properties group, the following properties have been summarized into the new **Compatibility** properties group:

- **Measured value of the actual current flow**

- **Measuring value of the calculated current flow**

- **Measured value of the current apparent power**

- **Load**

- **Load fixed [A]**

- **Load dynamic**

▸ In the grouped view in the zenon Editor, the following groups have been summarized into the **Topology** group:

- **Ground fault recognition**

- **Short-circuit detection**

- **Topological properties**

- **Compatibility**

8.4.2 Screen elements - new properties for ALC (F 49540, F 49545, F61164)

With the zenon `combined element` screen element, several new properties have been added for **Automatic Line Coloring**. The new properties have been grouped into the following areas:

▸ **Load flow calculation - input** (S 51765)

The input parameters for the **load flow calculation:** static input or via the linked variables with current measured values: **Active power [MW]**, **reactive power [MVar]** and **Voltage [kV]**.

▸ **Load flow calculation output** (S 51788, S 51790, S 73991)

The output from **load flow calculation**: Reactive power, phase and voltage.

▸ **Condition** (S 49545)

The **Load Flow Calculation** can write the information about number of the current sources to linked values in ALC sources - at the input or output of the element.

▸ **Load flow calculation transformer input** (S 54537, S 56917, S 59129, S 73991)

The input parameters for the **load flow calculation** via the process-technical properties of a transformer. Two-winding and three-winding transformers are supported. Reverse feed is also taken into account.

▸ **Load flow calculation transformer output** (S 59129, S 69008, S 69011)

Output from **load flow calculation** via the transformer.

▸ **Capacitor** (S 56917, S 59134)

The input parameters for the **load flow calculation** via the process-technical properties of a condenser.

New properties were also added for **ALC** for screen elements for the lines (`line`, `polyline` etc.). These are summarized in the following property groups:

▸ **Load flow line parameter** (S 51765, S 54537)

▸ **Load flow line result** (S 56916, S 62873)

8.4.3 Improved possibility for selecting an element when several elements are above one another (F 12433)

The **Select element** context menu makes it possible, when creating a screen or symbol, to quickly and unambiguously select an element when there are several elements above one another.

8.5 New colors for color selection (F 50127)

New default colors have been defined in the zenon color selection dialog.

There is a new default color palette available for the curves in the Extended Trend (ETM). The first 14 curves are shown in different colors before the color palette restarts again at the start of the color palette.

8.6 New keyboard shortcut for creation / removal of element groups (F 12433)

Element groups can now also be created with keyboard shortcuts (`Ctrl + G`) and also ungrouped again (`Ctrl + Shift key + G`).

8.7 New property when screen switching to worldview overview (F 70250)

The new **Show invisible elements in station list** property makes it possible to hide the invisible elements in the station list.

8.8 Substitution of screens (F 12433)

By activating the new **Take over from calling screen** property, the parameter for substitution of the calling screen is applied for the substitution of the screen to be called up.

The property is now available for the following screen elements:

- ▶ **Button**
- ▶ **Combo-/Listbox**
- ▶ **Combined element**

8.9 WPF transparency - incompatibility (B 108908)

The WPF element can be shown as `transparent` in the Editor with the **Fill/Transparent** property. In zenon 7.60, the default setting was `transparent`.

The element is not configured as transparent in zenon 8.00. It thus remains compatible with versions before zenon version 7.60.

For zenon 7.60, the setting must be corrected manually after project conversion. This is also applicable for other uses for which the property was set to `transparent` by default.

9. Functions and scripts

9.1 Enhancement for time period in the filter for screen switch function (F 53723)

The following possibilities for selection are available if the **Time Period** option has been selected in the filter dialog in the **Time** tab under **Filter**:

- ▶ **Display selection dialog**
- ▶ **Use current date and time**

This concerns the screen switch functions for the following types of screen:

- ▶ AML
- ▶ CEL
- ▶ ETM
- ▶ Archive revision
- ▶ ReportViewer

9.2 Enhancement of the time format selection during AML/CEL export (F 66279)

For the **export AML** and **export CEL** functions, there is now the possibility to select the time format of the export in the **Export Format** tab.

The following time formats are new:

- ▶ **Export timestamp in local time (ISO 8601)**
- ▶ **Export timestamp in UTC (ISO 8601)**

10. Menus

10.1 New nodes and new entry for action types in the properties of the context menu (F 50127)

The action type `Show Extended Trend` is now also available in the properties of the context menu, under **Representation/type** and **Action type**.

As a result, an ETM screen can be called up using the context menu of a dynamic element.

The variables linked to the dynamic element are automatically shown as a curve.

11. Modules

11.1 Alarms administration

11.1.1 64-bit time stamp (S 53259)

The time-related information can be shown in the full resolution of microseconds in the `Alarm Message Screen`.

The display in microseconds can be activated with the new **Microseconds** option in the filter dialog of the screen switching in the **Column settings** tab.

In addition, the following new **column types** have been added:

- ▶ `Outgoing internal time stamp`
- ▶ `Outgoing external time stamp`
- ▶ `Incoming internal time stamp`
- ▶ `Incoming external time stamp`
- ▶ `Reactivated internal time stamp`
- ▶ `Reactivated external time stamp`

11.1.2 "Delete alarm" renamed to "two-stage acknowledgment" (D 36368)

The **delete alarm** process has been renamed to **two-stage acknowledgment** and **Confirm acknowledgment**. All corresponding functions and buttons have been renamed.

11.1.3 Display of the configured equipment group (53262)

The information on equipment modeling can be visualized in the `alarm message list` type screen for display in Runtime.

To do this, in the filter dialog, in the **Column settings** tab, the new `Equipment group` **column type** has been added.
For display, configured equipment models can be selected using a drop-down list.

11.1.4 New selection dialogs for alarm/event group and class (F 56123)

Two new dialogs are now available for the selection of alarm/event group and alarm/event class.

You can do the following with alarm/event groups in these dialogs:

▶ create

▶ filter

▶ edit

▶ mode of operation

11.1.5 Shift filter for AML (F 57964)

The AML screen can now also be filtered for shifts. Only entries whose time stamp is within a shift from the shift filter and its equipment linking matches are shown. If no equipment filter is defined for a shift, only the time is taken into account for this shift.

Note:

▶ The shift filter requires a configured time filter. If the time filter is set to the **No time filter** option, the shift filter is deactivated. A notice of the cause of the deactivation is shown.

▶ If the lot filter is activated, the shift filter is automatically deactivated. Both filters mutually exclude one another. A notice of the cause of the deactivation is shown.

▶ If the **Show shift selection** option has been selected for the shift, a shift must be selected in Runtime in order for the screen to be able to be called up. If, in the **General** tab, the **Show this dialog in Runtime** option is activated, the complete dialog is called up.

11.1.6 Further configuration possibility for the display of alarms (F 56056)

There is now also a possibility to configure graphical display available for reactivated alarms.

11.1.7 Time period during configuration for absolute time period in the time filter increased (F 67326)

The absolute time period in the filter dialog and the time filter screen was extended to the year 2050.

Note: If the Runtime files are compiled in a project from a version of zenon before version 7.60, and the time filter has a start/end date later than 2038 in a function, the date is automatically set to 19/01/2038 03:14:07 (GMT).

11.1.8 New color selection possibilities for alarm texts and backgrounds (F 72201)

With the **Alarm Message List**, there is now the possibility to select the coloring for individual alarm texts and background colors. The flashing function has also been revised.

The following selection possibilities are available in the **Flashing** property:

- ▶ `Inactive`
- ▶ `Unacknowledged alarms`
- ▶ `Acknowledged alarms`
- ▶ `Unacknowledged alarms - time columns flash`
- ▶ `Acknowledged alarms - time columns flashing`

11.2 Equipment Modeling

11.2.1 Display of the number of active alarms (F 53261)

The number of active alarms can be visualized in the **Equipment modeling** module.

- ▶ The new **Aggregated alarms** property group was introduced in the equipment modeling properties for project configuration.
- ▶ The **Class linking** property group is also available in the equipment modeling properties.
- ▶ The dialog for switching screens to a zenon equipment modeling screen has been supplemented with the **alarm display** options.
- ▶ The number of active alarms can be shown on a configurable status variable.
- ▶ The display in the equipment tree can be configured as a number or with linked graphics.

11.2.2 Delete filter in Runtime (F 54380)

In the screen for equipment modeling, filters set in Runtime can now be reset again. There is a dedicated button for this: All equipment filters that have been set are thus removed from the linked screens and these are shown as unfiltered again.

11.2.3   Multiple selection in Runtime (F 54380)

Equipment groups can now be selected with multiple selection in the `equipment model` screen in Runtime. If more than one equipment group is selected, the **Execute function** button is deactivated.

11.2.4   Show complete structure (F 54380)

When screen switching to an `equipment model` screen, the new **show complete structure** option can be used to stipulate how objects are displayed in the equipment model tree in Runtime:

▸ Complete path of the object is shown.

▸ Only the name of the selected object is shown.

The complete path is shown by default. The option is set to display the name during project conversion however. Compatibility is thus ensured.

11.3   Historian

11.3.1   Shift filter for Archive revision (F 57964)

The archive revision screen can now also be filtered for shifts. Configuration is carried out with the function for screen switching.

Note:

▸ The shift filter requires a configured time filter. If the time filter is set to the **No time filter** option, the shift filter is deactivated. A notice of the cause of the deactivation is shown.

▸ If the lot filter is activated, the shift filter is automatically deactivated. Both filters mutually exclude one another. A notice of the cause of the deactivation is shown.

▸ If the **Show shift selection** option has been selected for the shift, a shift must be selected in Runtime in order for the screen to be able to be called up. If, in the **General** tab, the **Show this dialog in Runtime** option is activated, the complete dialog is called up.

11.3.2   Optimization for cyclical archives - use of all available processors (F 49807)

The recording and of cyclically-recorded archives has been revised. Cyclical archives now use all available processors of a computer. Lot archives of a cyclical archive have also been optimized.

In addition, the die logging of the archiving has been enhanced with further LOG messages.

11.4   Batch Control

11.4.1   Release master recipe function (F 54407)

With the help of the new zenon **Release master recipe** master recipe function, it is possible to release a master recipe in Runtime. The function releases the master recipes that have been found whose properties suit and whose possibly-embedded partial recipes have already been released. It can be instigated by a computer in the zenon network.

11.4.2   Limit values for parameters using variables (F 67709)

The minimum value and maximum value for a parameter can now also be dynamically stipulated in Runtime using variables. The variables are defined in the Editor using the **Variable for min. value** and **Variable for max. value** properties.

The variable values are read once when loading or reloading a recipe and applied to the master recipe.

11.4.3   Preventing parallel branching (F 67704)

You stipulate whether parallel branching is permitted using the new property **Disallow parallel branching**. If the property is activated, the validation in Runtime does not accept any parallel branches. Recipes that have already been validated are validated again before starting.

11.4.4   Change parameter values using the keyboard (F 67707)

Values for **numerical parameters** and **string parameters** can only be amended in the parameter list directly using a keyboard screen.

11.4.5   Changing parameter values in Runtime (F 67709)

The procedure to change parameter values in Runtime has been enhanced. Previously, the **changeable in master recipe** property allowed the changing of parameter values in Runtime. This property has been renamed and supplemented with a new property.

- ▶ **Parameter value can be changed in master recipe**: Allows the changing of values in Runtime.

- ▶ **IsEditableLimitsInRecipeLocal**: Defines whether only values or also the limit values (minimum/maximum) can be changed.

11.4.6   Touch operation for control strategies (F 67708)

Control strategies can now also be easily assigned a phase in Runtime by means of touch operation instead of the context menu. To do this, the control strategy is selected from a list and assigned a phase by means of a button.

11.4.7   Touch operation optimized (F 70662, F67708)

The following has been carried out to simplify touch operation in Runtime:

- ▶ Symbols have been added for the creation of matrix recipes with control elements

- ▶ The active touch elements have been made displayable

SYMBOLS AS CONTROL ELEMENTS

New features:

- ▶ New **matrix recipe creation** group with the following entries:
  - • **Delete row/column**
  - • **Move columns left / steps upwards**
  - • **Move columns right / steps down**
  - • **Activate selected elements**
  - • **Deactivate selected elements**

Enhancement to the **General** group in the **Recipe Control** group with these new entries:

- • **Save master recipe**
- • **Switch recipe to test mode**
- • **Release recipe**
- • **Switch recipe to edit mode**

- **Close recipe**
- **Close all other recipes**

SHOW ACTIVE ELEMENT

The new **Touch execution positions** property can be used to show semi-transparent circles that show the active touch elements.

11.5   User Administration

11.5.1   Login without existing domain connection (F 54377)

AD users can now also be logged on in Runtime if Runtime has been started by a local user (not an AD user). To do this, the **Access to Active Directory** must be activated in the **User Administration**. The name of the desired domain can then be entered in the **Acive Directory domain** property.

11.5.2   Read local user name from domains for AML and CEL (F 26334)

Complete Windows user data for local users can be read in Runtime via the domains and used for entries in the AML and CEL. To do this, the user names must be stored in the domain controller.

There must already be a connection to the domain controller when Runtime is started. Otherwise no complete user data are displayed. They are also not displayed if a connection to the domain controller is subsequently established. In this case, Runtime must be restarted with the connection established.

11.5.3   New power user user type (F 54402)

A new user type has been introduced for user administration: Power user.

This power user is positioned between user and administrator. They can create and administer other users in Runtime. If configured, they can also create and administer other power users.

11.5.4 Strengthened password encryption (S 70360)

The password encryption used for zenon user administration has a stronger algorithm when compiling Runtime files from version 8.00.

11.6 Chronological Event List

11.6.1 64-bit time stamp (S 53259)

The time-related information can be shown in the full resolution of microseconds in the `chronological event list` screen.

The display in microseconds can be activated with the new **Microseconds** option in the filter dialog of the screen switching in the **Column settings** tab.

In addition, the following new **column types** have been added:

- ▶ `Incoming internal time stamp`
- ▶ `Incoming external time stamp`

11.6.2 Display of the configured equipment group (53262)

The information on equipment modeling can be visualized in the `chronological event list` type screen for the display in Runtime.

To do this, in the filter dialog, in the **Column settings** tab, the new `Equipment group` **column type** has been added.
For display, configured equipment models can be selected using a drop-down list.

11.6.3 New features for the display of set value changes (F56056)

If, in the editor, in the project properties under **Chronological Event List**, the **Only log set value in the event of a change** property has been activated, an entry in the CEL is only made in the event of a change. There is no logging in Runtime if the same value is written once again.

The texts in the event of a set value change can be created by amending the following properties:

- ▶ **Text in the event of a change to a set value**
- ▶ **Text in the event of a change to a set value (old/new value)**

The texts created this way are output in Runtime in the CEL.

11.6.4 Shift filter for CEL (F 50857, 51862, 69893)

The CEL screen can now also be filtered for shifts. Only entries whose time stamp is within a shift from the shift filter and its equipment linking matches are shown. If no equipment filter is defined for a shift, only the time is taken into account for this shift. Screen switching to a linked filter screen can also be carried out. Shift filters can be used in the process.

The CEL filter settings are automatically amended by the shift filter.

In the screen switching to a CEL screen, the **Always show system messages in list** is taken into account by shift management.

Note:

▶ The shift filter requires a configured time filter. If the time filter is set to the **No time filter** option, the shift filter is deactivated. A notice of the cause of the deactivation is shown.

▶ If the lot filter is activated, the shift filter is automatically deactivated. Both filters mutually exclude one another. A notice of the cause of the deactivation is shown.

▶ If the **Show shift selection** option has been selected for the shift, a shift must be selected in Runtime in order for the screen to be able to be called up. If, in the **General** tab, the **Show this dialog in Runtime** option is activated, the complete dialog is called up.

If configured in the Editor, the filter dialog for configuration is offered for screen switching. This dialog is also called up if the **Filter** button is activated. The options generally correspond to those that are also available in the editor. Settings that can no longer be changed in Runtime are hidden. Additional information, such as the shift list, may be shown.

11.7 Extended Trend

11.7.1 Automatic orientation of Gantt curves (F50127)

If, for curves in the Extended Trend, the checkbox of the Adjust Gantt curves automatically property has been ticked, these can now be positioned automatically.

Configuration is carried out by means of settings in the following properties:

▶ Distance

▶ Reference position

▶ Arrangement

11.7.2   "Print Extended Trend diagram" function dialog available in Runtime (F 50154)

The "**Show this dialog in Runtime**" checkbox is available for the **Print Extended Trend diagram** function in the **Display** tab of the **Filter...** dialog.

11.7.3   Enhanced project configuration for mouse operation of the ETM in Runtime (F 50142)

Functions of mouse application can now be configured in the properties of the ETM diagram window.

This makes operation of the ETM easier in Runtime.

11.7.4   Filter dialog of the YT display optimized in Runtime (F 50154)

If, for the configuration of a YT display, the **Comparison with 2nd time period (only for an archive with filter for time range or lot)** checkbox has been activated, there is now a separate tab with the name **Time** available for the filter dialog in Runtime for each of the two time periods. However, this behavior only takes effect if the filter dialog in Runtime is opened using the **Diagram...** button.

11.7.5   New buttons: Activate curve, Activate axis, Curve color (F 50142)

There are three new buttons available for the **Extended Trend** view:

1. **Activate curve**
   Activates the selected curve in the extended curve list.

2. **Activate curve**
   Activates the selected axis in the extended curve list.

3. **Curve color**
   Allows the setting of the desired color for the respective curve.

11.7.6 New dialog for archive variable selection and instancing of archive variables (F 5029)

A new dialog for the selection of an archive variable has been created in order to use this as a curve or as an X-axis of an XY display. This selection dialog is available in the Editor as well as Runtime.

In the **Archive variable selection** dialog, you can now create one or more instances of an archive variable for curves using the **Instance** button.

The new entries are shown in the lower list and given ascending numbers. The copies made this way can be shown in Runtime as curves.

11.7.7 New line in the variable list in the archive variable selection dialog (F5029)

In the archive variable selection dialog, a line with the name automatic archive selection is now created in the variable list for each selected variable. If, the checkbox of the Active Curve property has been activated in this line, the archive is automatically selected in Runtime in order to find the best-possible solution for the time period displayed.

11.7.8 New display text for axis title and curve name (F23620)

There is a new display text available for the X-axis and Y-axis for the axis title.

There is a new display text available in Runtime for the curve name.

The following possibilities can be selected:

- ▶ **Static**
- ▶ **From variable**
- ▶ **Variable name**

The following is the case by default:

- ▶ **Static** is selected for the X-axis and Y-axis
- ▶ **Variable name** is selected for curves

11.7.9 New features for tick display, edit YT display and Y axis in the Curve dialog (F 23620, F 50099)

NEW POSSIBILITIES FOR THE SELECTION OF THE TICK CALCULATION

The selection has now been supplemented with the new possibilities **Fitted** and **Automatically optimized**.

- ▶ **Fitted:** Minimum value and maximum value are the extreme limits of the axis. The labeling of the display is automatically rounded to visually-clear figures and the ticks are amended accordingly. A small deviation to the set main ticks can occur if the space available on the axis can be divided evenly.

- ▶ **Optimized automatically** This setting is only applied in Runtime and does not change any configured axis limits in the Editor. To round the values, the most obvious round range of values is used.

NEW FEATURES FOR THE YT DISPLAY AND AUTOMATIC AMENDMENT OF THE TICKS TO THE TIME AXIS

For the YT display, the main ticks are now delivered so that they are at the more easily-readable time points. The start time of the day, for example, is shown with `00:00:00`.

If the **comparison with 2nd time period** property is activated, the time stamp of the two comparison curves is automatically synchronized. The data points with the same time stamp are now also marked at the same position.

11.7.10 New features for YT and XY display of the X-axis (F50099)

The **real-time display** property has been renamed to **Variable time stamp**.

The axis type can now be selected in the XY-display using a drop-down dialog. The following are available:

- ▶ **Linear**
- ▶ **Logarithmic**

The **Scale style** text field and the attendant **...** selection button has been newly added.

Note: If a selection is made under **Scale style**, properties that belong to the style can no longer be changed.

The text field of the scale style and the ... selection button have been deactivated in Runtime.

11.7.11 Possibility to control the curve visibility by means of a variable (F 50127)

The visibility of a curve can be controlled using a variable.

11.7.12 New "Source" column for curve lists in the screen switching filter dialog (F 50127)

In the filter dialog under **Data** and **Column settings**, the **Source** column is available in the curve lists. Information about the original location of the variables is visible under **Source**. These can only be read from archive variables however.

11.7.13 Curve selection is retained after editing (F50142)

If, in the filter dialog in the **Data** tab, a curve in the curve list is edited, the cell entry remains highlighted after closing the **Edit curve** dialog. This provides a clearer overview when editing several curves.

11.7.14 New possibility to substitute filter objects (F 50127)

Filter objects can now be substituted in the filter dialog of screen switching to a **Extended Trend** screen.

Linkings and indices can be replaced.

The dialog to **substitute filter objects** can be opened using the **Replace...** button.

The **Replace...** button is only available if, in the screen switching filter dialog, **Data** has been selected under **Archive data origin**.

11.7.15 Shift filter for Extended Trend (F 57964)

Extended Trend screens can now also be filtered by shifts. Only entries whose time stamp is within a shift from the shift filter and its equipment linking matches are shown. If no equipment filter is defined for a shift, only the time is taken into account for this shift.

Note:

▶ The shift filter requires a configured time filter. If the time filter is set to the **No time filter** option, the shift filter is deactivated. A notice of the cause of the deactivation is shown.

▸ If the lot filter is activated, the shift filter is automatically deactivated. Both filters mutually exclude one another. A notice of the cause of the deactivation is shown.

BEHAVIOR IN RUNTIME

The following is applicable for the shift filter in Runtime:

▸ When comparing to a second time range, one shift filter is used for both time ranges. If a new filter is assigned to the screen, both filters are overwritten with the same value.

▸ A `No time filter` time filter is not permitted.

▸ If the time filter is configured as relative, the shifts to be displayed are only determined at the time they are called up.
That means:

- At a later point in time, shifts that have been newly added or that have already expired are not taken into account.

- Only the shift that was the earliest or latest at the time of being called up is taken into account. This then moves from the display of the scale as time passes.

▸ If the shift filter is configured as **apply shift filter directly** plus **switch to the "show shift selection" mode**, but no shifts can be found, then:

- The time filter becomes invalid

- The time filter is automatically set to the default value: **Relative time period** `1 h`

- The automatic correction is shown in the filter dialog when the **Diagram** button is pressed

11.7.16 Improvements in the curve list configuration when screen switching (F 50127)

Both curve lists in the ETM screen switching filter dialog can now be configured individually under **Data** and **Column settings**.

11.7.17 Styles

New Extended Trend properties node available for variables and data types

The **Extended Trend** node is now available in the properties of the variables and the data type.

The selected styles are used in Runtime for the display of the Extended Trend.

Assignment of a scale style and linking of styles to the ETM screen switch function (F 50099)

In the editor, you can define the appearance of axes and displays and assign them a scale style, which is then applied in Runtime.

This possibility exists for the following axes and displays:

- ▶ X-Axis

  - • YT display

  - • XY display

- ▶ Y-Axis

In the **Filter…** dialog of the screen switch function of Extended Trend, you can now select a style group and different styles with the **Display** tab.

The selected style group and the selected styles are used in Runtime for the display of the Extended Trend.

11.7.18 Time filter

Display of the selected filter setting improved (F 53723)

If you select **No time filter** for the filter configuration, the start time is shown as *.

This is applicable for:

- ▶ The **alarm message list** screen type

- ▶ The **chronological event list** screen type

- ▶ The **report viewer** screen type

- ▶ The **report generator** screen type

- ▶ The **archive revision** screen type

The behavior of the switching of time and lot dialogs of the screen types listed have been amended to the default behavior.

In individual cases, this can lead to the configuration of a time filter as known from zenon 7.60 acting differently in Runtime in zenon Version 8.00. The corresponding places are amended automatically during project conversion.

New behavior when opening archive revision, Extended Trend and report viewer screens (F 50154)

The following screen types now perform the default action when the screen is opened:

- ▶ Archive revision
- ▶ Extended Trend
- ▶ Report Viewer

The behavior of the switching of time and lot dialogs of the screen types listed have been amended to the default behavior.

In individual cases, this can lead to the configuration of a time filter as known from zenon version 7.60 acting differently in Runtime in zenon Version 8.00. The corresponding places are amended automatically during project conversion.

Time filter behavior for functions harmonized (F 50154)

the behavior of the time filter of the following functions has been amended:

- ▶ Acknowledge alarms
- ▶ Confirm alarm acknowledgment
- ▶ Export AML
- ▶ Print AML/CEL
- ▶ Export CEL
- ▶ Report Viewer: Export/Print

11.8   Load Management (S 42694)

PEAK PERFORMANCE OF A GENERATOR CAN BE LINKED TO A VARIABLE

To do this, the new **Variable peak output** property has been added to the **Capacity/Priority** properties group. A variable can be linked using this property. As a result, it is possible to use a value from a variable in zenon Runtime for calculation instead of the fixed defined peak performance.

If a variable is linked, the value of each variable is applied at the start of the tariff interval in zenon Runtime. If the variable is not valid at this point in time, the configured fixed value (**Peak performance** property) is retained and the generator is not taken into account in Runtime for the calculation until the end of the tariff interval (planning of switching).

11.9   Message Control

11.9.1   Messages to user groups with shift filter (F 8974)

Users that have been assigned to a shift can be notified of alarms and events automatically using the **Message Control** module.
Notification is configured by means of the two modules **shift management** and **Message Control**.

When being sent to the user, there is now a check before the message is generated to see whether the user is linked to a current shift.

The behavior in Runtime depends on which method has been selected for the user group with shift filter:

Method	Procedure
Group call:	▸ The message goes to all users of the selected type who are linked in a shift for which the function execution is active. All users found are notified at the same time.
Following call:	▸ An attempt is made to find the first user of the group who is linked to the shift. ▸ This user is notified. ▸ If they cannot be contacted, the next user is searched for and notified. ▸ This is run through once until the end of the group has been reached. ▸ If no shift is still active during the process or no relevant users are linked in the active shifts, the notification is ended. A corresponding entry is made in the CEL.
Endless following call:	▸ An attempt is made to find the first user of the group who is linked to the shift. ▸ This user is notified. ▸ If they cannot be contacted, the next user is searched for and notified. ▸ This is run through cyclically until a user is reached. ▸ If no shift is still active during the process or no relevant users are linked in the active shifts, the notification is ended. A corresponding entry is made in the CEL.

Note:

▸   Only users who are linked in a shift for which the execution of the function is active are notified.

▸   Depending on the configuration, messages can also be sent to users who are no longer active in any shift.
Example: The type of sending (such as telephone) needs some time for sending. If, at the time of

generating the message, there are already further messages in the queue, the new message is sent with a delay. If the shift has been switched in the mean time, the message is nevertheless sent to the user of the previous shift.

▶ `Following call` and `Endless following call`: No further messages are created if no user of the selected group is active in the active shift.
Example: The message sending with endless subsequent call starts during **Shift 1**.
None of the recipients acknowledges the message. A shift-free time of 5 seconds is configured between **Shift 1** and the subsequent **Shift 2**.
If the timeout for acknowledgment runs out in precisely these 5 minutes, there is no recipient available at this point in time. The sending of the message is aborted. No messages are sent to their users after the start of the late shift either.

11.9.2   SMTP sending: Any desired attachment possible (F 59664)

With the **Mail message (SMTP)** dispatch type, it is now possible to use any desired file format for the **Appendix** option. The folder that contains the file can also be freely selected.

11.10   Process Gateway

11.10.1   General - The graphical user interface has been revised (S 58281)

The start dialog for the **Process Gateway** has been enhanced with information about the configured module, as well as the name of the configuration file.

11.10.2   AccessDNP3_SG (F 50446)

A completely new **AccessDNP3_SG** module for the **Process Gateway** was implemented on the basis of the new COPA-DATA DNP3 stack.

▶ The **AccessDNP3_SG** module is compliant with subset level 1, 2 and 3 and supports file transfer.

▶ The configuration is not compatible with the existing **AccessDNP3 Process Gateway** of zenon versions before version 8.00.

▶ The previous **AccessDNP3** module in the **Process Gateway** can no longer be selected for new configurations.

▶ Existing configurations with the previous **AccessDNP3** module continue to be able to run.

▶ The **AccessDNP3** module will no longer be supplied in a future version. The old module will be completely replaced by the **AccessDNP3_SG** module.

An XML device profile is supplied with the new module. This document with a description of the abilities of the **AccessDNP3_SG** module is included with the installation of the Editor.
Installation path:
`C:\ProgramData\COPA-DATA\zenon8.00\CommunicationProfiles\Dnp3\ProcessGateway`

11.10.3 AccessICCP

ICCP-TASE.2 - direct control (non-SBO) send to ICCP control object (51096)

In the ICCP-TASE.2 **Process Gateway**, as an ICCP client, it is now possible to configure, for a `Control Object` ICCP, a zenon command variable each, as well as a zenon response variable.

To do this, the **client variables** configuration was supplemented with the **Object type** option.

If a value for the command variable is written in zenon Runtime, the `Process Gateway` sends an `Operate Request` to the remote ICCP server. The response from the server (`Indication`) is sent to the response variable.

ICCP-TASE.2 as Client (S 63135)

When creating the datasets, the maximum PDU size is taken into account.

It is ensured that the maximum PDU size is adhered to for `defineVariableNameList` (create data sets) and `informationReport` (reports).

If the ICCP client reaches the maximum PDU size due to the data quantity, the ICCP client automatically creates several data sets.

11.10.4 Access OPC UA

Update to latest stack (S 68880)

The **OPC UA client driver** and the **AccessOPCUA Process Gateway** have been updated to **OPC UA Ansi C Stack**, version 1.03.341.

Action in the event of an empty configuration (S 57506)

With a new or empty configuration, the **OPC UA Server** no longer automatically adds all variables of all projects to the address area. Configurations where variables have not been explicitly defined in the address area must be amended accordingly after the update.

NamespaceIndex (S 57506)

Variables use the `NamespaceIndex 2`, that is incorporated into the `NamespaceArray`.

In earlier configurations, the NameSpaceIndex 80 is used, which was not included in the NameSpaceArray.

After the update, the addressing may need to be amended in **OPC UA Client** under certain circumstances

Support for archive variables (S 57506)

The **OPC UA Server** now allows the configuration of any desired archive variables, including aggregation archives. A **OPC UA client** can read raw values from the basic archive and calculated values from the aggregated archive using `Historical Access`.

An existing configuration, when selecting an individual archive and defined variables of the address area, remains valid.

For configuration, the configuration dialog of the **OPC UA Process Gateway** has been enhanced with the **Archives** tab.

11.10.5 IEC870 Slave (F 49542)

The configuration dialog was redesigned.
In addition, the following enhancements have been implemented:

870-101 slave supports balanced mode (S 59907)

870-101 connections now also support `Balanced Mode` in addition to `Unbalanced Mode`.
A **Link Address Size** 0 can now be configured for `Balanced Mode`.

In the configuration dialog of the device for 870-101, the field for the time configuration of **Master Polling Timeout** has been renamed to **Test Timeout**; the new default for this is 20 seconds.

Configure the import and export of Process Gateway configurations using a .CSV file (S 61671, 61791)

Configurations of the lists of the IOs for the IEC870 slave can be taken from a .CSV file in the Process Gateway module configuration or an existing project configuration can be exported to a .CSV file.

During import, new IOs can be created or existing IOs can be updated or deleted.

870 Gateway supports secured communication (S 50768)

The **IEC870 Slave** module of the **Process Gateway** now supports secure communication by means of TLS in accordance with the IEC 62351-3 standard.

The parameters for the configuration for this are set in the INI file of the module configuration.

DPI mapping for C_RC_NA_1 and C_RC_TA_1 (S 50563)

Double Point Mapping Now also applicable for the commands `C_RC_NA_1` and `C_RC_TA_1` (`Regulating step commands: T47, T60`):

▶ Value `0` corresponds to RCO=next step LOWER

▶ Value `1` corresponds to RCO=next step HIGHER

The behavior was thus amended to the pre-existing behavior for the commands `C_DC_NA_1` and `C_DC_TA_1` (`Double command`); value `0` is OFF and value `1` is ON.

Note: it is now possible to link the IOs that were configured in Process Gateway with **Type ID** `T47` and `T60` directly to the IEC850 `BOOL` type variables, for example to `*/TapChg/Oper.ctlVal[CO]`.

Restarting of the Process Gateway via status variable T00 with IOA 8 (S 50233)

In the IEC870 slave process gateway, it is possible to link a zenon variable to the `IOA 8` with type ID `<0>`
`Internal Status`.

If this linked variable has the value `1` in zenon Runtime, the configuration for the Process Gateway is
read in again. After successful reading-in of the new configuration, communication is restarted in the
IEC870 slave and the value of the variable is reset to `0`. If this process is unsuccessful, the Process
Gateway continues to work with the configuration loaded when the program started.

Select routing depending on variable configuration in zenon (S 50396)

The Process Gateway only reacts to a command variable by carrying out a **Select Routing** if the variable
is configured in zenon with active **Select Before Operate**.

The following is applicable for variables without an activated **Select Before Operate**:

- ▶ If, in **Process Gateway**, the sector setting **Select routing** is activated, this is ignored
- ▶ A received `Select` is responded to positively by the **Process Gateway**.
- ▶ A received `Select` is not routed to the `Auto/Remote section` of the **Command Processing**
  module.

Mapping of status bit N_UPDATE to NT Quality (73102)

The `N_UPDATE` status bit `(not updated in zenon network)` is mapped to NT (not topical) Quality in
ASDUs in the Monitoring direction.

Higher tolerance for queries for ASDUs with incorrect configuration (S 50232)

Queries from an 870 master with unknown ASDUs or unconfigured data points are handled with
increased tolerance. These no longer lead to the closing of a connection but are responded to with
corresponding error codes in COT. The connection remains intact in the process.

- ▶ ASDU An unknown **Type-ID** is responded to with `COT 44`.
- ▶ `C_IC_NA_1` (general query) is:
  - On receipt of an invalid `COT`, it is responded to with `COT 45`.
  - on receipt of `QOI != 20`, `Station interrogation` is triggered. This corresponds to the
    behavior the same as for `QOI = 20`.
  - On receipt of `IOA !=0`, handled like `IOA=0`.
- ▶ For `C_CI_NA_1` (`Counter interrogation`) it is the case that `RQT != 5` is treated the same as
  `RQT = 5`.

The receipt of invalid values is logged in the LOG file.

11.10.6 MODBUS Slave supports Function Codes (S 59019)

The MODBUS Slave Process Gateway module now also supports function codes:

- `01 Read Coils`
- `15 Write Multiple Coils`
- `05 Write Single Coil` (optional)
- `06 Write single Register`

To do this, the configuration dialog for the **MODBUS** module in the **Process Gateway** has been enhanced:

- New **Connection** tab
  Contains the connection settings

- **Holding Registers**
  Configuration of the assignment for variables

- New **Coils** tab
  Configuration of the assignment for variables for Coils

UNIT IDENTIFIER CAN BE CONFIGURED UP TO 247 (D 38234)

The **Modbus Slave** in the **Process Gateway** now uses the `Unit Identifier "1"` by default. This value can be configured between 1 and 247. In requests, the **Modbus Slave** also accepts, in addition to the configured `Unit Identifier`, `Unit Identifier 0` and `255`.

11.11 Production and Facility Scheduler (PFS)

11.11.1 Bank holidays: No selection (F 48763)

The dialog to enter national holidays has been amended. The `None` entry does not set any national holidays and removes all pre-existing national holidays.

11.12 Process Recorder

11.12.1 Logging and recording of limit value breaches (F 53263)

The **Process Recorder** module supports the logging and playback of limit value breaches.

On the basis of this, the flashing behavior of the display in zenon Runtime is recorded for the Process Recorder and visualized in playback.

The configuration and display of the flashing behavior in zenon has been documented in the **Variables** manual in the **Flashing** chapter.

11.12.2 Visualization of configuration changes in playback mode (F 59113)

Configuration changes in zenon screens are visualized in playback mode.

If, in the configuration of a zenon screen, a screen element is repositioned or added, for example, this is also visible during playback in playback mode.

The **text color for system messages** was added in the screen switch function for the **Process Recorder** modules for this reason.

> ⚠ **Attention**
>
> *The general project property **Versioning active** must be activated when configuring for correct display in playback mode.*

11.13 Reporting

11.13.1 Report Viewer (F 65701)

Lot filter - enhancement of the filter parameters (S 69039)

The **FilterType** parameters have been enhanced with the display of lots. This display can be configured with new parameters.

Example: `@FilterData(String) = LOT12345; 17.08.2017 15.50.00; 17.08.2017 16.50.00; 0T 01:00:00`

Data sets for archived data take into account data from backup folder (S 65703)

If data sets with archive data as a data source are used for report display, there is the choice to read archive data from the backup folder or from the Runtime folder.

To do this, the **Screen switch** function of the configuration dialog of the data sets with `archive values` as a data source has been enhanced with the **Use archives from backup folders** option.

New control elements for navigation in report viewer screens (S 65704, 65705, 65708,)

The content of the internal report viewer tool bar for navigation can be configured with control elements in the `Report Viewer` screen. These zenon control elements are primarily suitable for visualization with Multi-Touch operation.

The internal tool bar of the report viewer can be hidden for display in zenon Runtime. To do this, the new **Show integrated Report Viewer tool bar** property was introduced for configuration in the zenon Editor for the `report window` control element in the **Representation** screen properties group.

Shift filter for Report Viewer (F 51862)

`Report Viewer` screens can now also be filtered for shifts. This filter is also used for the **Report Viewer: export/print** function.

Note:

- ▶ The shift filter requires a configured time filter. If the time filter is set to the **No time filter** option, the shift filter is deactivated. A notice of the cause of the deactivation is shown.

- ▶ If the lot filter is activated, the shift filter is automatically deactivated. Both filters mutually exclude one another. A notice of the cause of the deactivation is shown.

Support for current report formats (S 65702)

As a result of the implementation of the current Microsoft report control, reports can now be displayed with the current standard for report definitions. This allows, for example, the display of reports that have been created with Report Builder version 3.0.

.RDL files that have been created with version 2.0 of Report Builder can continue to be used in the current zenon version.

Supported report schemes:

- ▶ `schemas.microsoft.com/sqlserver/reporting/2010/01/reportdefinition` (Microsoft Report Builder 3.0)

▶ schemas.microsoft.com/sqlserver/reporting/2008/01/reportdefinition
(Microsoft Report Builder 2.0)

Attention: If current zenon projects contain RDL data that has been created with Report Builder 3.0, a warning dialog is shown during compilation for older zenon versions. Conversion of these .RDL files into version 2.0 is not carried out.

11.14 Recipegroup Manager

11.14.1 New DynProperty ZValue (FS 38688)

Scaled recipe values can be read via the zenon API with VBA.

The **ZValue** DynProperty has been implemented for this.

Example:

```
public void Macro_RgmScaledValue()
{
var RgmGroup = this.RGMGroups().Item("Test");
var RgmRecipe = RgmGroup.RecipeItem("Recipe0");
var RgmValue = RgmRecipe.ValueItem("ScaledVar[2]");


var Value1 = RgmValue.get_DynProperties("ZValue");
var Value2 = RgmValue.get_DynProperties("Value");


MessageBox.Show(string.Format("Scaled value = {0} value = {1}", Value1, Value2));
}


public void Macro_RgmScaledValueWrite()
{
var RgmGroup = this.RGMGroups().Item("Test");
var RgmRecipe = RgmGroup.RecipeItem("Recipe0");
var RgmValue = RgmRecipe.ValueItem("ScaledVar[2]");


var Value1 = RgmValue.get_DynProperties("ZValue");


RgmValue.set_DynProperties("ZValue", "2");
```

```
RgmRecipe.Save();

RgmGroup.Save();

}
```

11.14.2 Default value for "Invalid recipes writeable via function" (Def. 36766)

In the Recipegroup Manager, the default value for the **Invalid recipes writeable via function** property in the **Recipe Group Manager** property group was already set to `inactive` in version 7.60.

The default value is thus also set to `inactive` after a project conversion.
Invalid recipes are thus no longer set using the function.
If invalid recipes can no longer be written, activate the checkbox for the property.

11.14.3 Signature can be set in the event of parameter changes

When parameters are changed in recipes, it is now possible to set whether a signature is necessary for this.

Signature text editable can be used to select whether the signature is editable or not, or whether the project setting is to be transferred.

11.15 Shift Management

11.15.1 CEL entry in the event of a message not being sent (F 8974)

If, when notifying shift users with the `subsequent call` or `endless call` method, no further users can be notified, a corresponding entry is generated in the CEL.

11.15.2 Display (F 50730)

Colors and fonts for the display of the calendar in Runtime have been amended. The shift management can thus also be displayed correctly with 4k resolution.

11.15.3 Bank holidays (F 48763)

National holidays are shown in the calendar as work-free time. The name of the national holiday is shown instead of the day.
The language of national holidays can be switched.

National holidays for a country can now be entered into shift calendars automatically. to do this, a country is preselected for each project in the Editor using the new **Shift Management/National holidays** project property. Its national holidays are automatically applied in the shift calendar.

National holidays can also be taken into account or excluded when creating shift models.

11.15.4 Hierarchical filter for equipment groups (F 48758)

When linking shifts to equipment groups, a hierarchical filter can be activated for the equipment groups. This allows the inclusion of subordinate equipment groups.
Behavior of the filter:

- ▶ `Active`: For all equipment groups displayed in the **filter list**, their subordinate equipment groups are also selected and used.

- ▶ `Inactive`: Only the equipment groups shown in the **filter list** are used.

11.15.5 Filter information with shifts (F 49288, 57964)

Shifts can serve as filters for the display of information in other screens.
The following can be filtered:

- ▶ AML
- ▶ Archive revision
- ▶ CEL
- ▶ Extended Trend
- ▶ Function **Report Viewer: export/print**
- ▶ Report Viewer
- ▶ Time/Lot/Shift filter

To filter information with shifts, use:

- ▶ Filter for screens in screen switching for the shift management
- ▶ Shift filter in screen switching on one screen

Note:

▶ The shift filter requires a configured time filter. If the time filter is set to the **No time filter** option, the shift filter is deactivated. A notice of the cause of the deactivation is shown.

▶ If the lot filter is activated, the shift filter is automatically deactivated. Both filters mutually exclude one another. A notice of the cause of the deactivation is shown.

▶ The following is applicable for **AML**, **CEL** and **archive revision**: If the **Show shift selection** option has been selected for the shift, a shift must be selected in Runtime in order for the screen to be able to be called up. If, in the **General** tab, the **Show this dialog in Runtime** option is activated, the complete dialog is called up.

11.15.6 Shifts in time filter screens (F 50821, 69893)

Filter screens can be filtered for shifts. This applies for:

▶ AML filter

▶ CEL filter

▶ Time/Lot/Shift filter

Note:

▶ The shift filter requires a configured time filter. If the time filter is set to the **No time filter** option, the shift filter is deactivated. A notice of the cause of the deactivation is shown.

▶ If the lot filter is activated, the shift filter is automatically deactivated. Both filters mutually exclude one another. A notice of the cause of the deactivation is shown.

11.15.7 Insert shift model (F 8886)

An equipment group must now be selected when adding shift models to the calendar. The time period for which the shift model is inserted must also be selected.

11.16   Industrial Maintenance Manager

11.16.1   Tree and list view can be individually configured in the Editor (F 64792)

The tree and list view that is familiar from Runtime can now also be individually configured in the Editor.

11.16.2   Configurable list for screen switch function (F 64792)

The dialog for a **Screen switch** on a zenon `Industrial Maintenance Manager` screen has been redesigned.

The content to be displayed and the display when screen switching can be configured with a configurable list.

11.16.3   New equipment modeling node (F 64791)

The new **equipment modeling** node is now available in Runtime:

▸   Equipment models and equipment groups are displayed in this node.

▸   New devices can only be created for equipment groups.

▸   It is possible to filter in Runtime using the checkboxes of the equipment groups.

▸   The filter results are displayed in a list.

11.16.4   Moving of devices by means of drag&drop (F 64791)

Devices from a list can be moved by means of dragging & dropping from the **Master Data** node to the **Equipment Modeling** node and vice versa.

Note: Devices can only be assigned to equipment groups, not the equipment models or the equipment modeling itself.

12. zenon Web Server

12.1 zenon Web Client Starter (F 49774)

The new **zenon Web Client Starter** allows zenon Web Client to be started from any desired browser, via the command line or a link to arguments. The **zenon Web Client Starter** is also installed when the zenon Web Client is installed.

It is also possible to establish the folder for the Runtime files in the **global_vars.js** configuration file.

START VIA BROWSER

Start from any desired browser:

1. User starts zenon Web Client via a link in the landing page.

2. Operating system finds zenon Web Client Starter.

3. Operating system starts zenon Web Client Starter and transfers link arguments.

4. zenon Web Client Starter analyzes link arguments.

5. zenon Web Client Starter shows zenon Web Client in accordance with link arguments

START VIA COMMAND LINE OR A LINK TO ARGUMENTS

Start via command line or a link to arguments::

1. User starts the zenon Web Client Starter via the command line or a link to arguments.

2. zenon Web Client Starter analyzes arguments.

3. zenon Web Client Starter shows zenon Web Client according to the arguments transferred.

13. zenon Logic

zenon Logic is the programming environment integrated into zenon in accordance with IEC 61131. It is available as Editor and Runtime (Soft-PLC) for zenon Supervisor and zenon Operator.

13.1 Increased number of shared-memory connection (S 60794)

For shared-memory connections between zenon Runtime and zenon Logic Runtime, a maximum number of 256 connections is supported.

13.2 zenon Logic version 9.2 (FS 39439)

The zenon Logic Workbench integrated into zenon Editor has been updated to version 9.2.

13.3 OPC UA Server as fieldbus driver is no longer available (S 76188).

Since version 7.60 the **zenon Logic OPC UA Server** with article number `ST-K5BusOPCUASrv` has not been in the price list. With version 8.00 the zenon **zenon Logic OPC UA Server** as **fieldbus driver** is no longer supported..

Ther **OPC UA Server** in the **Process Gateway** with article number `ZGW-OPCUA` is still available and can be used as an alternative solution.

13.4 New keyboard shortcuts (CDFR QC 1388)

The following new keyboard shortcuts are available in zenon Logic Workbench:

- ▶ New keyboard shortcuts to expand and collapse the tree view:

  - `Ctrl + Shift + right cursor key`:
    Expand all nodes

  - `Ctrl + Shift + left cursor key`:
    Collapse all nodes

- ▶ ST Editor

  - `Ctrl + K`:
    The selected line is converted into a comment. consecutive lines can be highlighted together and converted into a comment.

  - `Shift + Ctrl + K`:
    If the selected line is a comment, the comment syntax is removed.

13.5   Multiple selection of objects in a project (CDFR QC 1461)

In the tree view of a project, several objects can be selected for deletion or moving by means of dragging & dropping.

Selection is carried out with a mouse click when holding down the Ctrl key. Selection, deletion or moving over several levels of the tree structure is possible.

13.6   Program execution - cycle, exceptions and graphical display (CDFR QC 1446)

In order to not execute programs automatically with each PLC scan, corresponding time periods and phases for programs can be configured. As a result, programs no longer have to be executed with each cycle.

The view of the **Cycle** dialog has been enhanced with statistical information. This information is shown as both text and graphics. For each cycle, there is a visualization of whether the program has been executed in the respective cycle or not.

13.7   Cross references - hierarchical display of search results (CDFR QC 1363)

The results of the **Search in files...** command is shown in a hierarchically-subdivided tree view for display in the **cross references** window. The nodes of the view can be expanded or collapsed.

13.8   Compile standard presettings for natives C with Microsoft Visual C++ 2017 Compiler (BUG 109112)

The presetting for the native compiling of the IEC 61131-3 program code (C compiling) was amended to the current **MSVC2017 Compiler**.

13.9   UML-like display of the program call structure (CDFR QC 1396)

Programs that have been created in a project can be visualized in a graphical view. To do this, the **program call structure** has been added to the graphical user interface of the Workbench.

- ▶ The visualization is similar to the display of the **Unified Modeling Language**.

- ▶ The program calls are shown in a tree structure.

- ▶ Each program is shown in its hierarchical structure.

- ▶ The substructures contained in a displayed structure element are shown in a detailed view by clicking on it.

13.10 Drivers

13.10.1 DNP3 driver reimplemented (F 50446)

The drivers for **DNP3 Master** and **DNP3 Outstation** have been reimplemented. These are based on a new **DNP3** stack from **COPA-DATA** and are compliant with the new **DNP3, Level 3** standard.

13.10.2 Ethernet/IP Scanner - Explicit Messaging for "Read Single" attribut and "Write Single" attribut-(CDFR QC 1374)

The Ethernet/IP scanner now supports `Explicit Messaging` for the reading and writing of `Read Single` and `Write Single` attributes.

14. Network

14.1 Flashing (F 49952)

The flashing action has been revised:

- ▶ In zenon it has been ensured that screen elements in a zenon screen visualize the same flashing behavior as on the server.

  - • This display is made at the same timer and with no delay to any clients in the network.

- ▶ The performance in the zenon network has been optimized:

  - • The communication between the primary server, standby server and clients has been optimized.

- Superfluous network telegrams have been removed.

- Information about the flashing status is synchronized in zenon.

▶ zenon Runtime is compatible with servers of versions before 8.00.

▶ The reload behavior for changes to the configuration with regard to flashing has been amended and optimized.

▶ The configuration and display of the flashing behavior in zenon has been documented in the **Variables** manual in the **Flashing** chapter.

14.2 Correct display of variable values in an evaluated network (S 41498)

In older versions, it may be the case that variables of an evaluated network have not been correctly displayed in a zenon screen in Runtime. Values of these variables are only updated after a new screen is called up.

This possible malfunction has been corrected.

14.3 Performance optimization (RQ 41498)

STANDBY BUFFER

The standby buffer necessary for seamless redundancy has been revised. The saving of data has been optimized in this buffer.

Optimizations for the standby buffer in redundant networks:

▶ The filling and processing of the buffer in the event of redundancy switching has been revised.

- Buffer values are processed on the standby server directly.
  In doing so, these buffer values are received directly by the standby server's driver.

- No data sync is carried out in the event of a primary server failure. This is no longer necessary with the new system. The values are taken from the standby server's buffer, which it also maintains, in the event that it becomes the primary server.

▶ The network load for the data sync in the event of redundancy switching has been heavily reduced.

14.4 Server 2 as data server - property removed in the Editor (S 58392)

The **Server 2 as data server** functionality (project property in the **Network** properties group) has been removed from the zenon Editor for version 8.00.

15. Programming interface

15.1 Add-Ins (F 35673, F 26233)

New in version 8.00 for Add-Ins in general:

▶ Activation of VBA/VSTA is no longer necessary in order to be able to use add-ins.

▶ Amended and new Add-Ins can be reloaded in Runtime.

15.1.1 Stipulate development environment for add-ins (F 54533)

In the settings of the zenon Editor, it is now possible to establish which development environment is used to edit Add-Ins. The setting is in the new **Add-Ins**.

When clicking on **Open add-in Editor...** in the **Extras** menu, the pre-set programming environment is opened.

15.1.2 Compatibility (F 51729, 54538)

zenon add-ins are forwards and backwards compatible.

▶ Backwards compatibility:
Add-Ins Are also compatible with newer versions of zenon. If, for example, an Add-In is compiled for version 7.60, it can also run in version 8.00.

▶ Backwards compatibility:
Add-Ins Can be compiled for earlier zenon versions. Thus, for example, an Add-In with zenon 8.00 can be created for zenon 7.60.

15.1.3 Minimum zenon version (F 54538)

For each Add-In, you must define the zenon version for which it is to be compiled. These are defined in the respective programming environment via **Add-In Framework Settings** => **Minimal Scada Version**.

In the Editor, it is possible to check the version for which it is compiled with the Add-In **Minimum version** property. This is also shown in the detail view of the add-ins.

15.1.4 zenon functions (F 35673)

There are now zenon functions available for add-ins:

▸ **Display dialog "Manage Runtime services"**: Opens the dialog to manage the services.

▸ **Execute Project Wizard Extension**: Allows the selection of a wizard that can be executed in Runtime.

15.1.5 COPA-DATA Developer Tools (F 58098)

New features and changes for the **COPA-DATA Developer Tools** in zenon 8.00:

▸ Add-ins can be created for zenon versions from version 7.60.

▸ Support for **Microsfot Visual Studio 2017** (Community, Professional, Enterprise).

▸ Provides project templates for WPF wizards for zenon Runtime and zenon Editor.

15.1.6 New storage location for the "\EditorAddInDeploy" folder (F 66429)

The `\EditorAddInDeploy` folder can now be found under the following path:

`%Program Files (x86)%\COPA-DATA\zenon8.00\EditorAddInDeploy`

15.2 Driver - enhanced access via zenon API (F 127220)

The configuration of drivers with the zenon API has been enhanced.

The API interface was enhanced for version 8.00 with the following drivers:

▸ **Allen Bradley RS-Linx driver**
`ALLANBNT.exe`

- ▶ **BACnetNG**
  BACnetNG.exe

- ▶ **Bernecker&Rainer PVI Driver NG**
  BURPVI.exe

- ▶ **Beckhoff TwinCat NG driver**
  BeckhNG.exe

- ▶ **DNP3TG driver**
  DNP3_TG.exe

- ▶ **Hilscher MPI driver**
  CIFMPI.exe

- ▶ **Modbus Energy Driver**
  MODBUS_ENERGY.exe

- ▶ **Modbus RTU driver**
  MODRTU.exe

- ▶ **OPCUA Client driver**
  OPCUA32.exe

- ▶ **PHLOEM driver (OEM ABB only)**
  PHLOEM32.exe

- ▶ **Pozyton Energy Meter Treiber** (new driver)
  POZYTON.exe

15.3 API - New Events for the ALCEngine Interface (S 62882)

3 events have been added to the zenon API **ALCEngine** Interface.

These are always executed if the following is applicable in zenon:

- ▶ Updated
  An **ALCEngine** calculation has been completed.

- ▶ LoadflowCalculated
  The calculation of the statistical load flow analysis - in the **Load Flow Calculation** module - has been completed.
  Note: A load flow analysis of the topological interlocking check or the N-1 analysis of the **State Estimator** do not trigger this event.

- ▶ NMinus1Calculated
  The N-1 calculation - in the **State Estimator** module - has been completed.
  Note: An N-1 analysis of the topological interlocking check does not trigger this event.

15.4   New API online help

The graphical interface of the help for the zenon API interface has been redesigned. The content was revised and supplemented with examples for the add-in.

API ONLINE HELP FOR ZENON DRIVERS

The general configurations that can be achieved for all zenon drivers with the API programming interface has been redocumented in a separate area for drivers.

In addition, the dynamic properties for each driver that is available in the interface have been compiled in a separate group. This part of the documentation has also been revised.

The online help for the zenon API is also included in the **COPA-DATA Developer Tools**.

> 💡   **Information**
>
> The most recent version of the API online help can also be called up in the internet at the URL http://onlinehelp.copadata.com.

15.5   Visual Studio

15.5.1   Integration of the COPA-DATA Developer Tools into the Microsoft Visual Studio programming environment (F 77383)

The functionality of the **COPA-DATA Developer Tools**   available in the **Visual Studio Marketplace** has been enhanced:

▶   You can use the **F1 key** in the **Microsoft Visual Studio** to call up the zenon API documentation directly.

▶   A direct search query on the basis of the zenon API documentation in **Microsoft Visual Studio** is not possible.

▶   Starting from search results in **Microsoft Visual Studio**, you can now jump directly to the corresponding chapters in the zenon API documentation.

15.6 Conversion of colors between .NET and zenon

Colors in the zenon API have been saved as a numerical data type. You can use the following examples of code to carry out a conversion to .NET colors (or vice versa) via VSTA; furthermore, using color palettes is shown.

HELPER CLASS TO CONVERT .NET COLORS IN ACCORDANCE WITH ZENON AND VICE VERSA

C# code

```csharp
namespace WorkspaceAddin
{
class ZenonColorHelper
{
// convert .net Color to zenon API color
public static uint Color2Api(System.Drawing.Color color)
{
// zenon stores as ABGR, where A is the highest significant bit
return (uint)((color.A << 24) | (color.B << 16) | (color.G << 8) | (color.R << 0));
}


// convert zenon API color to .net Color
public static System.Drawing.Color Api2Color(uint apiColor)
{
// zenon stores as ABGR, where A is the highest significant bit
byte a = (byte)(apiColor >> 24);
byte b = (byte)(apiColor >> 16);
byte g = (byte)(apiColor >> 8);
byte r = (byte)(apiColor >> 0);


return System.Drawing.Color.FromArgb(a, r, g, b);
}


// convert hex color string (#FFFFFF) to zenon API color
public static uint ColorString2Api(string colorString)
{
System.Drawing.Color color = System.Drawing.ColorTranslator.FromHtml(colorString);
return Color2Api(color);
}
```

```csharp
// convert zenon API color to hex color string (#FFFFFF)
public static string Api2ColorString(uint apiColor)
{
System.Drawing.Color color = Api2Color(apiColor);
return System.Drawing.ColorTranslator.ToHtml(color);
}


// extracts the index in the color palette, which is encoded in the zenon API color
// returns -1 if the given color does not represent a color palette index
public static int GetColorPaletteIndex(uint apiColor)
{
System.Drawing.Color color = Api2Color(apiColor);
if (color.A == 128) // alpha value == 128 -> ind from color palette
{
// G defines the nr of 8bit chunks, R is the actual index
return color.G * 256 + color.R;
}


return -1;
}


// encodes the index in the color palette into a zenon API color
public static uint EncodeColorPaletteIndex(int index)
{
int chunks = (int)(index / 256.0);
int idx = (int)(index % 256);


System.Drawing.Color color = System.Drawing.Color.FromArgb(128, idx, chunks, 0);
return Color2Api(color);
}
}
}
```

EXAMPLE FOR THE USE OF THE HELPER CLASS

C# code
`using zenOn;`

```
public void Macro_ColorHandlingSample()
{
// first we need a project to work with

IProject project = Application.MyWorkspace.ActiveDocument;
if (null == project)
{
System.Diagnostics.Debug.Print("Error! No project loaded / activated");
return;
}


// create a sample picture with a filled rectangle in it (color defined directly in
the element)


// create sample frame for picture if it not yet exists
if (null == project.Templates().Item("SampleFrame"))
project.Templates().Create("SampleFrame", false);


// delete and recreate sample picture or create a new one if it not yet exists
IDynPicture pic = null;
IDynPictures pictures = project.DynPictures();
for (int i = 0; i < pictures.Count; i++)
{
IDynPicture picture = pictures.Item(i);
if (picture.get_DynProperties("Title").ToString() == "SamplePicture")
{
pic = picture;
break;
}
}
if (null != pic)
{
pic.Save();
pictures.Delete(pic.Name);
}
pic = project.DynPictures().Create("SamplePicture", "SampleFrame",
tpDynPicturesTypes.tpStandard);
```

```
// create a rectangle for demo and assign a background color

IElement ele = pic.Elements().Create("Rectangle", tpElementTypes.tpRectangle);
ele.BackColor = ZenonColorHelper.Color2Api(System.Drawing.Color.Aqua);

System.Diagnostics.Debug.Print("zenon color: " + ele.BackColor);
System.Diagnostics.Debug.Print("As Html: " +
ZenonColorHelper.Api2ColorString(ele.BackColor));

System.Drawing.Color color = ZenonColorHelper.Api2Color(ele.BackColor);
System.Diagnostics.Debug.Print("As Color: r" + color.R + " g" + color.G + " b" + color.B
+ " a" + color.A);

// create a filled rectangle (color defined directly from a color palette)

// look for existing color palette in current project or create a new one
IColorPalettes palettes = project.ColorPalettes();
int nrOfPalettes = (int)palettes.get_DynProperties("Palette");
if (0 == nrOfPalettes)
{
//Create a new palette
palettes.CreateDynProperty("Palette");
//Define a color in the new palette
String htmlColor = "#00FFFF"; // System.Drawing.Color.Aqua
palettes.set_DynProperties("Palette[0].PaletteColors", htmlColor);
}

// link color palette entry to element background color
ele.BackColor = ZenonColorHelper.EncodeColorPaletteIndex(0);
System.Drawing.Color index = ZenonColorHelper.Api2Color(ele.BackColor);
System.Diagnostics.Debug.Print("Index encoded in color: r" + index.R + " g" + index.G
+ " b" + index.B + " a" + index.A);
System.Diagnostics.Debug.Print("Palette index:" +
ZenonColorHelper.GetColorPaletteIndex(ele.BackColor));


// output of the sample:
//
// zenon color: 4294967040
```

```
// As Html: #00FFFF
// As Color: r0 g255 b255 a255
//
// Index encoded in color: r0 g0 b0 a128
// Palette index: 0
}
```

15.7   Warning dialog when the VSTA editors is opened (FS 37529)

A warning dialog is shown before the VSTA editor is opened in the zenon editor.

16. Styles

16.1   New possibility to create style group (F 50099)

In the context menu of the **Styles** node and in the tool bar, it is now possible to create empty style groups with **Create new style group**.

16.2   New "Select style types" dialog (F50099)

A new dialog to select style types when a new style is created is available.

The style types are in alphabetical order. Multiple selection is possible.

17. Drivers

17.1   New drivers

The following new drivers have been developed for zenon version 8.00.

17.1.1   ADP - Alternative Data Points (F 54597)

With the new **ADP - alternative data points** driver, a zenon variable can receive values from different sources.

Variables of the ADP driver are configured for display in zenon Runtime. The parameters for these variables with the `alternative variables` driver object type are set up in the zenon Editor in the **Alternative variable** variable properties group.

- ▶ A list of **Source variable** is configured in the process. These variables can be configured for different drivers.
- ▶ Each variable from which the values can be received can be selected with a **Selector** variable.

In doing so, the **Alternative variable** is available for **ADP driver** variables.

17.1.2   PHLOEM (S 62782)

The **PHLOEM driver** communicates with an ABB engine control unit directly.

These control units are called `Drives`:

- ▶ Several configured `Drives` can be connected by means of Ethernet or a serial interface.
- ▶ The parameters of a `drive` are sub-divided into groups. The parameters of a group have a corresponding index.
- ▶ zenon variables can be assigned to the parameters of a `Drives` by means of a correspondingly-set data block and offset.

API INTERFACE

The dynamic properties of the drivers are also available in the zenon API.

17.1.3   POZYTON (F 69050)

The **Pozyton driver** reads values from the POZYTON energy meter via an RS 232/485 interface.

▶ Communication takes the `IEC62056-21 protocol standard modified for Pozyton sLAB` into account.

▶ Addressing of the variables is by means of the **Net address**:

▶ The configurations for drivers can also be configured via API or XML access.

The following object types are available for the driver:

Driver Object Type	Description
Counter	Counter Information (VI())
DateTime	Time and Date (T())
ReceiverAccount	Receiver Account Handle (K())
ProgrammingInfo	Meter Programming Infos (LW())
EnergyCounter	Energy Counter Meter Value (PXXX())
Exceed_EnergyCounter	Exceed of Energy Counter (EQ())
Frequency	Frequency (F())
ActivePower	Momentary Active Power of Meter (P())
PassivePower	Momentary Passive Power of Meter (Q())
VoltagePhase	VoltagePahse (U())
MeterCurrent	Current which flows into Meter (I())
IncreasingPowers	Increasing Powers (PN())
PrevCyclePowers	Powers from porevious cycle (PO())
StrongMagneticPowers	Energy P+ counted in the presence of strong magnetic field (ENP())
StrongMagneticInfluence	INformation about strong magnetic field influence for meter (FM())

API INTERFACE

The dynamic properties of the drivers are also available in the zenon API.

17.2   64-bit time stamp for internal time stamp [μs] (S 51288)

The internal time stamp of variables supports the full resolution of microseconds (μs).

17.3   New API online help

The graphical interface of the help for the zenon API interface has been redesigned. The content was revised and supplemented with examples for the add-in.

API ONLINE HELP FOR ZENON DRIVERS

The general configurations that can be achieved for all zenon drivers with the API programming interface has been redocumented in a separate area for drivers.

In addition, the dynamic properties for each driver that is available in the interface have been compiled in a separate group. This part of the documentation has also been revised.

The online help for the zenon API is also included in the **COPA-DATA Developer Tools**.

> 💡 **Information**
>
> The most recent version of the API online help can also be called up in the internet at the URL http://onlinehelp.copadata.com.

17.4   3S V3 driver for PLC handler

17.4.1   3S V3 - detection of the byte sequence on the basis of the RDA type (S 51083)

The **3S V3** driver uses the RDA types used as a basis to automatically detect the byte sequence (Intel or Motorola) in which the data is transferred by the controller and processes it accordingly.

17.4.2 Amended behavior with advise for RDA trigger variables (S 76714)

RDA triggers are no longer advised for asynchronous reading but explicitly read by the driver every 100 milliseconds. If a trigger is 1, the RDA mechanism is started and then reset to the value 0 directly afterwards.

17.5 Allen Bradley RS Linx driver (F 127220)

The API interface for the driver has been enhanced.
The dynamic properties of the driver have been documented accordingly in the new zenon API online help.

17.6 Azure drivers can receive JSON messages in the IoT Hub format (S 54805)

The **Azure driver** can now also process JSON messages in the IoT Hub format of the **Azure Process Gateways**. The detection of the format of the message taken from the service bus queue is automatic.

17.7 BACnet NG

17.7.1 API expansion (D 38190)

The API interface for the driver has been enhanced.
The dynamic properties of the driver have been documented accordingly in the new zenon API online help.

17.7.2 Performance

REQUEST FOR SEVERAL OBJECTS POSSIBLE (S 73759)

`present-value` and `status-flags` are read for several objects in a **Read-Property-Multiple Request**. In doing so, the driver attempts to use the configured **Max. APDU** size.

These **Object-Properties** were previously used individually for each **Object**.

OBJECT_LIST AND OBJECT_NAMES

The attendant objects are initially determined with a `read-Request` to **Object_List-Property** and **Property list** . A list of IDs that can be resolved into names is received from the **BACnet Device** as a response.

UNSUBSCRIBE COV SUBSCRIPTION IN THE EVENT OF UNADVISE

- ▸ `COV Subscriptions` Are advised if variables are inactive and when the driver is closed.

- ▸ If a variable is configured as **COV**, but is not contained in the notification, the INVALID bit is set and a corresponding LOG message is created.

- ▸ If no `Notification` is received after a `Subscription` from the **BACnet Device** (APDU Timeout x (Retries +1)), the INVALID bit is set for the variable and a corresponding LOG message is created.

17.7.3   Disable Read-property-multiple (RPM) service (S 76108)

If `Read-Property Multiple` is not supported by the BACnet Device, this option can be deactivated and the `Read-Property` BACnet service can be used.

This behavior can be manually configured in the **Device** tab with the new **Disable Read-property-multiple (RPM) service** option.

17.7.4   Online variable import amended to BACnet standard (S 71856)

Online variable import from a BACnet Device has been revised and optimized.

The information in the dialog for variable import (such as variable name, IEC data type, BACnet data type, ...) are read from the BACnet Device and shown in the dialog (BACnet standard 14/2012).

17.7.5   Driver is BACnet Device (S 71855)

The **BACnetNG driver** is, from zenon 8.00, also a BACnet device and responds to queries.

17.7.6 Assignment of incoming notifications via Process Identifier (S 71862)

The assignment of incoming `COV-Notifications` and `Event-Notifications` to a configured BACnet device is carried out by means of a configurable **Process Identifier**.
To do this, the new **Process ID** option can be configured in the configuration dialog for the driver in the **Devices** tab.

Assignment was carried out using the source address in the previous version.

17.8 BURPVI - Bernecker & Rainer PVI Driver - API enhancement (F 127220)

The API interface for the driver has been enhanced.
The dynamic properties of the driver have been documented accordingly in the new zenon API online help.

17.9 BeckhNG - Beckhoff TwinCat NG driver - API enhancement (F 127220)

The API interface for the driver has been enhanced.
The dynamic properties of the driver have been documented accordingly in the new zenon API online help.

17.10 CIFMPI - Hilscher MPI driver - API enhancement (F 127220)

The API interface for the driver has been enhanced.
The dynamic properties of the driver have been documented accordingly in the new zenon API online help.

17.11 DNP3TG driver (F 127220)

The API interface for the driver has been enhanced.
The dynamic properties of the driver have been documented accordingly in the new zenon API online help.

17.12   EibV2_32 - support for 4-byte data types (S 48171)

Enhancements for the EibV2_32 driver

- ▶ The **driver** now also supports the data types DINT and UDINT.

- ▶ Configuration for **group address** is possible in the `0 to 31` area. this was only possible from `0 to 15` in previous versions).

- ▶ Support of **Main Group Address** `16 to 31` is available from ETS version 4.

17.13   Internal driver - new licensing model (S 63009)

The licensing for the variables of the internal driver has been amended:

- ▶ Does not require a license:
  - No display of the external time stamp and limited visualization of the status (`spontaneous` and `time internal` only) in Runtime
  - **Internal driver** variables are not calculated as **TAG** for licensing
- ▶ Requires a license:
  - Complete visualization of all time stamps (internal and external) and all statuses in Runtime
  - Each **internal driver** variable is counted as a **TAG** for licensing

When configuring, the display type and licensing are determined with the new **State/Timestamp for Intern Driver variables** property in the **Runtime settings** properties group.

> ⚠ **Attention**
>
> *When converting projects, the new State/Timestamp for Intern Driver variables property is not activated. To visualize the time stamp and status in Runtime, this property must be activated in the zenon Editor.*

17.14   IEC850 (F 59057, FS 38674)

Enhancements and amendments for the **IEC850 driver**:

- ▶ **GetNameList on DO**
  The driver reads the data model from an IED during the `Association` (establishment of the connection). If the IED uses a smaller TCP frame size (PDU size) than is necessary to read the

data model, this is automatically detected and the procedure is amended dynamically.
The **GetNameList on DO** option, which is thus obsolete, has been removed from the **Server** configuration dialog, **Application Layer (ACSE)** option group, but not from the configuration file (TXT) of the driver.
Note: If the zenon project is compiled for a version before version 8.00, a corresponding entry in the configuration file continues to be taken into account.

▶ Support for key words `$SCADA_SERVER1` and `$SCADA_SERVER2` instead of the computer name in the **Hostname** property.

▶ Improved speed when reading the RCBs during online RCB assignment via online selection (**Statically assigned RCB**):
When assigning RCBs in the configuration dialog, the RCB configuration is no longer read in from the IED in full for each RCB assignment. The information loaded for the first RCB assignment is cached The time-intensive reading is not carried out for further RCB assignments.
However, another read is still carried out when the IP address is changed or the driver dialog is opened.

▶ Display of the data set name for RCB assignment
In the configuration dialog to assign RCBs by means of online selection, the data set assigned to the RCB is shown in the new `Data set` column.

▶ RCB with invalid linked data set
An RCB with an invalid data set is not used in Runtime. A corresponding warning entry is generated in the LOG file for this automatic deactivation.

▶ Evaluation of data from reports without data set name.
If the driver receives a report without the name of the data set, it attempts - starting from **RptID** - despite the lack of a data set name, to assign the received data of the existing zenon variables. In doing so, the driver assumes that the IED in this report uses a data set that has been read during the association.
In the configuration dialog of the **Optional Fields**, the **Data set name** option can now be deactivated. A prerequisite is that the IED has a unique RptID for each RCB.

▶ The **Statically assigned RCB** dialog for the configuration of RCBs has been revised. The **RptID** option was added.

▶ The status of the Resv attribute is ignored when validating the availability of the RCB for statically-assigned URCBs if a **ClientLN** is configured for the driver configuration.

▶ LOG entry on receipt of a report with `BufOvfl` that has not been triggered by `PurgeBuf`
If a report is received by a driver with the `BufOvfl` flag set and this received report is not the first report that has been received (after activation of the RCB with `PurgeBuf`), this is logged by the driver in the LOG file with a corresponding warning message:

▶ The driver opens the TCP socket in Windows with 20 seconds KEEP_ALIVE timeout.
If the driver does not send a TCP telegram within 20 seconds (no `MMS request`), the Windows operating system sends a TCP `keep_alive` message every second. If no response is received for 10 such `keep_alive` messages, the connection to the 850 server is closed.

▶ Automatic interruption in communication with pending `MMS requests`:
If the IEC850 receives no response for the sent `MMS requests` from the 850 server and is

blocked for more than 50 seconds as a result, the connection to the IED is disconnected automatically. An 850 client does not offer any alternative solution for this non-standard-compliant IED behavior. The cancellation creates an error message in the LOG file.

▶ `TimeAccuracy` Ignore for received time stamps
The **Ignore time accuracy** option in the driver configuration dialog can be used to deactivate the rounding precision of the time stamp received.

▶ Pseudo `Select confirm` and `Cancel confirm`:
A Select and Cancel for the values `ctlModel=1` or `3` (direct control) is positive responded by the driver for the **command processing**.

▶ **DPI/DPC Mapping** For Bitstring MMS in command direction to `BOOL` data type variables.
If the data type has been amended to `BOOL` data type for the command variable, the driver maps the value `FALSE` to `0x40 (lower)` and the value `TRUE` to `0x80 (higher)` for **write set value**.

17.15   IEC870 (F 8941)

SET S/E BIT ON "DEACTIVATION" (S 51852)

If the IEC870 driver sends a `Cancel` for a `Select` command (Deactivation), the S/E bit is set in the ASDU.

This bit is irrelevant for `Deactivation` in accordance with the IEC60870 standard. This behavior is to support some PLCs that evaluate this bit nevertheless.

END OF INITIALIZATION STARTS NEW GENERAL INTERROGATION (S 70361)

If, on a sector (COA), during an ongoing general interrogation (actterm not yet received), an End of Initialization (M_EI_NA_1) is received, the GI request is canceled and started again.

SECURE COMMUNICATION VIA SSLTLS (S 50768)

The **IEC870 driver** now supports secure communication by means of TLS in accordance with IEC the 62351-3 standard.

Configuration of the TLS connection is configured in the text file of the driver configuration.

17.16   IEC62056 Supports historical values (FS 38179)

17.17   Modbus Energy driver (F 127220)

The API interface for the driver has been enhanced.
The dynamic properties of the driver have been documented accordingly in the new zenon API online help.

17.18   Modbus RTU driver (F 127220)

The API interface for the driver has been enhanced.
The dynamic properties of the driver have been documented accordingly in the new zenon API online help.

17.19   New driver commands - simulation modes on driver start (S 47843)

The driver commands were enhanced with the different driver modes when the driver starts.

New driver commands:

- ▶   **Driver in programmed simulation mode**
- ▶   **Driver in static simulation mode**
- ▶   **Driver in counting simulation mode**

17.20   OPC UA client

17.20.1   Update to latest stack (S 68880)

The **OPC UA client driver** and the **AccessOPCUA Process Gateway** have been updated to **OPC UA Ansi C Stack**, version 1.03.341.

17.20.2 API expansion (F 127220)

The API interface for the driver has been enhanced.
The dynamic properties of the driver have been documented accordingly in the new zenon API online help.

17.20.3 Multiple subscriptions (S 68867)

The **OPCUA32 driver** now supports the possibility to create several subscriptions.

▸ In doing so, the same settings are used for all `Subscriptions`.

▸ Assignment is carried out manually using the data block number of the variables.

▸ A `Subscription` is created by the driver for each unique data block number.

▸ If variables are configured and created with a different data block number and the driver creates several `Subscriptions`, the number of `PublishRequests` is amended automatically as a result.

Note: In order for the number of the **Data block** to be able to be used for the configuration of the **subscriptions** for the variables, the driver configuration must be opened again and confirmed with **OK**.

17.20.4 Operation limits configurable (S 68873)

The **OPCUA32 driver** now takes **operation limits** into account. The **OPCUA32 driver** now uses a maximum of 500 elements in a `Request` by default. Provided the OPC UA server supports a higher number and a higher number has a beneficial effect on communication, this limit can be increased accordingly.

The **Operation Limits** tab was added in the configuration dialog of the **OPCUA32 client driver**.

▸ The driver allows the configuration of **Operation Limits**. These are taken into account during communication in the zenon Editor when browsing and in Runtime.

▸ As an option, the **Operation Limits** can also be read from the `Server Address Space` by activating the new **consider "server Operation Limits"** option, provided the server supports this.

17.20.5 Automatic unique "client session name" (S 68204)

The **OPCUA32 driver** now uses a unique `ClientSessionName` on the primary server and standby server for the connection in Runtime.

17.21 S7TIA - Enhancements (S 65765, 74007)

VARIABLE IMPORT TAKES COMMENTS INTO ACCOUNT

The variable selection dialog has been amended. The comments that were entered for this project are now displayed. This comment is applied for the zenon project configuration for the variable in the **Identification** property.

VARIABLE IMPORT OF TIA 14 PROJECTS

The **S7 driver for S7-1500 1200** (**S7TIA driver**) now also supports TIA 14 projects.
In driver configuration, TIA 13 and TIA 14 projects can be selected in the file selection dialog in the **Options** tab of the **TIA Project (must be present on runtime machine)** option.

17.22 SNMPNG - display of SNMP agent data as a table (S 67684)

Data from SNMP agents can be displayed in table form on zenon variables.

▶ The variable import dialog has been supplemented with the "**Tables**" tab, with which the table variables can be created by means of import in the zenon Editor.
In doing so, the SNMP tree of a selectable SNMP agent is shown below a selectable start OID in a tree display. If an individual node of this tree represents the tree of a table, this node can be selected as a table variable. All subnodes of the selected node are automatically created in the zenon Editor variables of the `SNMP - Table` driver object type.

▶ New driver object type: `SNMP - tables`
Interprets all SNMP variables that are in the structure of the SNMP tree below the OID of the zenon variable configured in the `SNMP - tables` driver object type as a table. If this interpretation is successful, the table content is written to a string variable in columns.

DISPLAY OF OCTET STRINGS

In previous versions, there were sometimes problems when displaying string content.
For example, the byte values of the ASCII codes were displayed for these if they contained certain

special characters or control characters. Conversely, MAC addresses were displayed as text if the address bytes happed to be able to be convertible into alphanumeric characters. This problem was rectified from zenon version 8.00 by using an SNMP API function that automatically undertakes the necessary conversion.

However, to do this, the MIB files must be stored in the CommunicationProfiles directory:
`RegistryKey:ProgramData_8100 +"\CommunicationProfiles\SNMP-MIBS".`
Example: `C:\ProgramData\COPA-DATA\zenon800\CommunicationProfiles`

17.23 System driver

17.23.1 New system driver variables for load flow calculation (S 70788)

The **[Automatic Line Coloring]** has been added to the system driver variables. These contain information about the optional **Lad Flow Calculation** and **State Estimator** modules:

▸ **[Automatic Line Coloring] load flow calculation error**
Information on results of the load flow calculation.
Example: Error if a variable relevant for the **Load Flow Calculation** module has an invalid value (`INVALID`, `OFF` status).

▸ **[Automatic Line Coloring] load flow calculation status**
Information about the configured type of load flow calculation.

17.23.2 System driver (S 53890)

The following system driver variables have been removed from the list of configurable system driver variables:

▸ **[Project information] driver queue overflow**
▸ **[Project information] driver queue overflow name**

These variables are retained for existing project configurations from previous versions. However they are no longer supplied with values from version 8.00.

> 💡 **Information**
>
> Due to the optimizations to the connection to the zenon, there can no longer be a queue overflow.

18. Tools

18.1 3D Integration - Enhancements

SUPPORT FOR FILES IN JT FORMAT (S 75570)

The **3D configurator** supports the import of files in JT format.
Monolithic JT files can be imported into versions 8.0, 8.1, 9.0, 9.5 and 10.

LINKING TO VARIABLES OF STRUCTURE TYPES (DEF. 127545)

zenon variables of structure types can also be linked during configuration in the **3D configurator**.

19. Variables

19.1 64-bit time stamp for internal time stamp [µs] (S 51288)

The internal time stamp of variables supports the full resolution of microseconds (µs).

19.2 Flashing (F 49952)

The flashing action has been revised:

▶ In zenon it has been ensured that screen elements in a zenon screen visualize the same flashing behavior as on the server.

- This display is made at the same timer and with no delay to any clients in the network.

▶ The performance in the zenon network has been optimized:

- The communication between the primary server, standby server and clients has been optimized.

- Superfluous network telegrams have been removed.

- Information about the flashing status is synchronized in zenon.

▶ zenon Runtime is compatible with servers of versions before 8.00.

▶ The reload behavior for changes to the configuration with regard to flashing has been amended and optimized.

▶ The configuration and display of the flashing behavior in zenon has been documented in the **Variables** manual in the **Flashing** chapter.

19.3 Reaction Matrix

N_UPDATE STATUS BIT CAN BE EVALUATED VIA REACTION MATRIX

The "not updated" [**N_UPDATE**] status bit can be evaluated with a reaction matrix.

TEST BIT [TEST] IN MULTI-BINARY AND MULTI-NUMERIC REACTION MATRICES CAN BE EVALUATED

The test bit [TEST] can be evaluated in a multi-binary or multi-numeric reaction matrix.

To do this, a new checkbox for the test bit TEST has been added in the dialogs for the corresponding reaction matrix for the **variable status** option.

19.4 System driver (S 53890)

The following system driver variables have been removed from the list of configurable system driver variables:

▶ **[Project information] driver queue overflow**
▶ **[Project information] driver queue overflow name**

These variables are retained for existing project configurations from previous versions. However they are no longer supplied with values from version 8.00.

> 💡 **Information**
>
> Due to the optimizations to the connection to the zenon, there can no longer be a queue overflow.

19.5   Flashing (F 49952)

The configuration and display of the flashing behavior in zenon has been documented in the **Variables** manual in the **Flashing** chapter.

19.6   Substitution of interlocking variables (F 12433)

You can now substitute interlocking variables at the symbol in order to use a previously-created interlocking for many other applications, each with their own interlocking variable.

Substitution is effected using the **Element entry** dialog.

20. Wizards

20.1   GIS Editor (F28850)

The GIS editor is automatically installed during the standard zenon installation. You get to it in the zenon Editor via **Start Editor wizards...** and **GIS Editor**. The **GIS Editor** allows the configuration of a GEO-data based ALC configuration.

The result of this configuration is saved in a file that contains all information for display in zenon Runtime.

The locations of errors in line networks can be localized and visualized in map views with the **GIS Editor**.

For project configuration, lines, areas and markers are supported. In doing so, variables, functions and ALC objects from zenon primary and subprojects can be used.

Data in ***.KML** and ***.KMZ** format can be imported into a folder in the **GIS Editor**. The lines, areas, markers and layer information in this folder are then applied in the project.

20.2   Metering point administration included when installing zenon web client (F 8627)

Metering point administration with all dependencies is also installed when installing the zenon web client.

ActiveX control must still be registered manually after installation.

20.3 Metering Point Administration - chinese user interface (Def. 38624)

The graphical user interface of the **metering point administration** is now available in Chinese for the wizard as well as for display in zenon Runtime.

20.4 "\ExtensionAddons" folder - new save location (F 66352)

The `ExtensionAddons` folder, necessary for the installation of **Messstellenverwaltung**, can now be found under the following path:

```
%Program Files(x86)%\COPA-DATA\zenon8.00\ExtensionAddons\
```

20.5 WPF wizards - significantly shortened loading time (F 54540)

The loading time when opening WPF wizards in Runtime has been optimized.

20.6 System text wizard supports Chinese (Def. 108543)

The **system text wizard** now also supports `Chinese`. Chinese text from zenon Editor and zenon Runtime is thus available in the language table when configuring language-switchable projects.

21. Important information

21.1 ActiveX Controls

If special ActiveX controls are developed, the following has to be considered:

If the DISPATCH – which is passed in the **zenonInit** event of zenon – is saved in the ActiveX control, an **AddRef** has to be carried out because this DISPATCH is only valid within the **zenonInit** event. If "**AddRef**"

is not called, a crash of the entire Runtime will be the result. Additionally, a release has to be performed in the "**zenonExit**" event.

21.2   Buttons and screen elements with screen-type specific functions

Buttons and elements with screen type-specific functions may only be used once on a screen. If there are identical elements on a screen, all duplicates are removed during compilation.

Example: If a button is copied and pasted in the same screen, the copy is removed during compilation.

Exception: Several containers can be created in a `Faceplate` screen.

21.3   Integration of VBA wizards and VSTA wizards

All VBA wizards are saved in the file called `ZenWorkspace.vba`   by the zenon Editor. All VSTA wizards are saved in `Workspace AddIn`.

When performing a new installation, these files will only be copied to your computer if they do not already exist in the installation folder. Existing VBA/VSTA files are not overwritten, because all your changes would be deleted in this case. If you want to use our new wizards or modified ones, you can import them manually via the menu "**File** – **Update Wizards**" in the Editor. At this you can decide yourself which wizards you want to overwrite.

21.4   Complex vector graphics

Please note when configuring process screens. When using many or complex vector graphics, loading screens in the Runtime can take a long time.

21.5   Converting projects

When a project created using a program version prior to version 5.50 is activated, the profiles contained in the structure schedules are converted. Profiles are no longer supported in version 5.50 or later versions. For each profile, a structure schedule is created, containing the linked schedules. The linked schedules contain all the times which fall within the profile's activation/deactivation times.

The day information contained in the profile schedule is input into the calendar.

The configuration process can be viewed in the Editor's output window.

21.6 MS-ActiveX element DBGrid32.ocx does not work

There a several problems known in context with the use of Microsoft ActiveX element **DBGrid32.ocx** in the Runtime. Therefore please use other ActiveX elements such as **MSDATGRD.ocx**.

21.7 Reload of projects with Simulator driver variables

Simulator driver variables, not projected as HD variables, are reset to the value `0` with the function "**Reload**". Only HD simulator driver variables keep their value after reloading.

21.8 Network access - Firewalls

Different components of zenon   try to access the network and can cause an alarm by firewalls or personal firewalls. If you want to use the network or the zenon Remote Transport, you have to unlock the according TCP/IP ports.

The following zenon components result in network access:

- ▸ Administration service (`zenAdminSrv.exe`)
- ▸ Editor (`zenone32.exe`)
- ▸ Database server (`zendbsrv.exe`)
- ▸ Diagnosis Server (`zenLogSrv.exe`)
- ▸ OPC Server (`zenOPCsrv.exe`)
- ▸ Process Gateway (`zenProcGateway.exe`)
- ▸ Remote Desktop (`zenVncSrv.exe` and `zenVncCli.exe`)
- ▸ Network server (`zennetsrv.exe`)
- ▸ Transport service (`zensyssrv.exe`)
- ▸ Drivers with TCP/IP connections
- ▸ zenon Web Server (`zenWEBsrv.exe`)
- ▸ zenon Logic Workbench
- ▸ zenon Logic Runtime

21.9   Process Desk – killing tasks

The **zenon Process Desk** allows you to end frozen tasks.

Attention: Some drivers need a certain follow-up time, because they write a process image on closing. Premature closing can result in data loss! Use this option only in case of emergency, when you are really sure, that the task will not close on its own.

21.10   Saving reports of the Report Generator in the Runtime

Please be aware that on saving reports in the Runtime, all functions are replaced by the current contents of the cells (numbers). The functions in these reports (**.xrs** files) are no longer available. Additionally, these reports can no longer be edited in the Editor. So please use the MDI function "**Save as**" so that the original reports from the Editor are not overwritten. Moreover, we recommend to define the original reports as `read-only`.

21.11   The database server service must be entered correctly in the Startup Tool

Beside the versions you can also change the data base server with the **Startup Tool**. If you use this function, please note:

Between version 6.21 SP0 and 6.22 SP0 the SQL Service was entered incorrectly in the `zendb.ini` by the setup. This was no problem because the **zenDBSrv** did not consider the value. As of 6.22 SP1 this is the case again.
If you read the values using function **Read from zenDB.ini**, the values are stored wrongly in the **Startup Tool**. You must check existing entries and change them if necessary.

21.12   zenon Logic Intellisense is slow

For large programs the Intellisense function of the zenon Logic Workbench can cause the project to open very slowly. In this case you should deactivate the Intellisense function in the straton Workbench.

21.13   Transport service Autostart

The transport service (**zensyssrv.exe**) is normally started automatically by the operating system when a user logs in. If the transport service is not started, the computer cannot be reached via the Remote

Transport.

At a new installation it is restarted after the computer has rebooted.

If you accidentally delete the entry for the automatic start from the registry, you can restore it with the help of command `Register` in the Startup Tool. At this the transport service is also automatically restarted.

21.14   Overwriting Runtime files

When creating Runtime files in the zenon Editor it can happen, that files changed in online operation are overwritten. This occurs with the following modules:

▶   Recipegroup Manager

▶   Production & Facility Scheduler or Scheduler

▶   User administration

▶   Standard Recipes

In order to guarantee that data created in runtime (recipes, schedules etc.) is not lost when creating Runtime files, there is a new tab in the dialog for project configuration: **Runtime changeable data** For the modules mentioned above you can define here whether the concerned files should be overwritten when Runtime files are created. If the checkboxes are not active, the files are overwritten!

This behavior is also true for the Remote Transport, when the Runtime files are to be transferred to another computer. So these checkboxes also apply here. If you want to transport all files to the remote system, deactivate all checkboxes. Otherwise the corresponding data will not be transported.

When creating Runtime files and when using Remote Transport, a message appears in the output window indicating that the concerned files were not overwritten.

The standard setting is: Runtime Files are not overwritten!

21.15   zenon Web Client: No support for Google Chrome from version 42

From version 42, Google Chrome no longer supports NPAPI plugins. Chrome can thus no longer be used as a zenon web client from version 42.