zenon manual

Reporting

v.7.10

COPADATA
do it your way

Contents

1. Welcome to COPA-DATA help

GENERAL HELP

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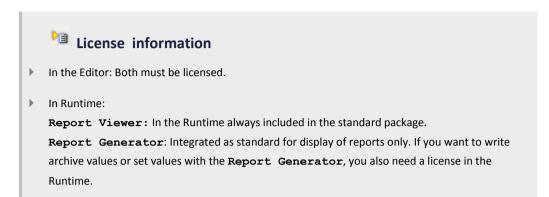
2. Reporting

The integrated reporting in zenon serves as documentation, evaluation and presentation of process data on the basis of online and archive data from zenon. Reporting is divided into:

- ▶ Report Generator (on page 5)

- ▶ Report Viewer (on page 197)

Neither reporting variant is available with Windows CE.

> ### ▶ License information
>
> ▸ In the Editor: Both must be licensed.
>
> ▸ In Runtime:
>   **Report Viewer:** In the Runtime always included in the standard package.
>   **Report Generator**: Integrated as standard for display of reports only. If you want to write
>   archive values or set values with the **Report Generator**, you also need a license in the
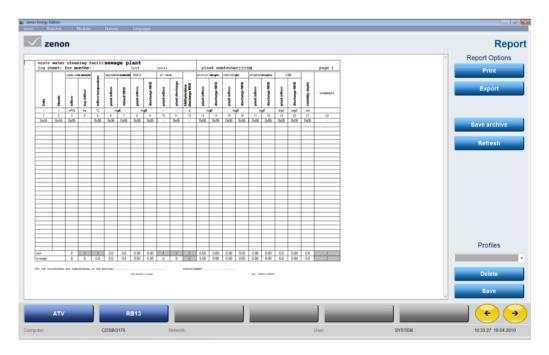>   Runtime.

zenon Analyzer is available for comprehensive, platform-independent reporting of the process level through to ERP. Your COPA-DATA sales partner would be happy to inform you about this.

2.1   Report Generator

The Report Generator documents, evaluates and presents process data on the basis of online and archive data. The report generator works on a cell-orientated basis. Each cell can be assigned certain functions and formats.

Reports are configured using the **report** (on page 7) screen. The data is filtered, calculated, issued and displayed via report functions (on page 45). These are entered using dialogs or manually. Values can be read and - if permitted on the screen - also edited. The report generator meets the requirements of rule sheet and ATV H260.

PROJECT MANAGER CONTEXT MENU

Menu item	Action
New report	Creates a new report in the list in detail view.
Import existing report...	Opens the File Manager to select an existing file.
Editor profile	Opens the drop-down list which includes pre-defined Editor profiles.
Help	Opens the online help.

Info

The status bits NORM and N_NORM are not displayed in the report. The status bit NORM cannot be changed using the report.

2.1.1   Engineering in the Editor

To be able to use reports in Runtime, the following must be carried out in the editor:

▶   A report type screen (on page 7) is created

▶   A report (on page 9) is created and configured

▶   A switching function (on page 158) is configured in Runtime

Recommended procedure for planning the report:

▶   Create report (on page 9)

▶   Format report (on page 22)

▶   Create archive filter (on page 43)

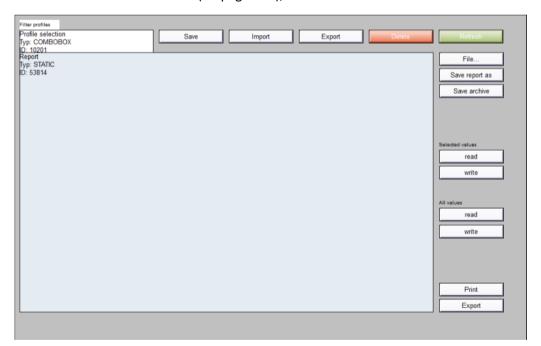▶   Configure report functions (on page 45)

Create screen of type Report Generator

In the type report screen, the reports created with the report generator are displayed in Runtime and configured with the corresponding rights. The display size depends on the template size selected. The report that is displayed can be determined in screen switching or in Runtime.

In order to create a report screen:

1. Select the `New screen` command in the `Screens` node

2. In the `Screen type` column, select the `Report` drop-down list

3. in the menu bar, select *Control elements -> Add template*

4. A report with standard settings is created

5. If necessary, add further control elements from the `Control elements` menu

6. Create a screen switch (on page 158), to select the screen in Runtime

Parameter	Description
`Insert template...`	Opens the dialog for selecting a template for the screen type.
	Templates are shipped together with zenon and can also be created by the user.
	Templates add pre-defined control elements to pre-defined

	locations in the screen. Elements that are not necessary can also be removed individually once they have been created. Additional elements are selected from the drop-down list and dragged onto the screen. Elements can be moved on the screen and arranged individually.
`Report`	Report window to display the report.
`Display: Set filter`	Displays the the currently-configured time filter in Runtime.
<u>Buttons</u>	Buttons for controlling in the Runtime.
`Refresh`	Recreate build report completely (values and display).
`Print`	Print report in Runtime.
`File`	Select report file and change filter conditions.
`Export`	Export report.
`Save archive`	Write changed values to the archive.
`Save report`	Report is saved in Runtime (*.xrs). **Attention:** When saving, formulas and functions are replaced by the current values. The functions in these reports (.xrs files) are no longer available. These reports can also not be edited in the editor anymore. **Recommendation:** Set original report files to "write-protected" status and save changes with `Save as` in a new file.
`Save report as`	Report is saved under a new name during Runtime (*.xrs), formulas and functions are replaced by current values.
`Read all values`	All values are read in again from the driver.
`Read selected values`	Selected values are read in again from the driver.
`Write all values`	All values are sent to the driver.
`Write selected values`	Selected values are sent to the driver.

| Line up	Scroll text in list element up
Line down	Scroll text in list element down
Column right	Scroll text in list element to the right
Column left	Scroll text in list element to the left
Page up	Scroll text in list element up
Page down	Scroll text in list element down
Page right	Scroll text in list element to the right
Page left	Scroll text in list element to the left
Filter profiles	Buttons for filter profiles.
Profile selection	Display of the available filter profile and input for new profile names.
Save	Saves current setting as a profile.
X (save)	Deletes selected profile.
Import	Exports profiles.
Export	Imports profiles.

Create a new report
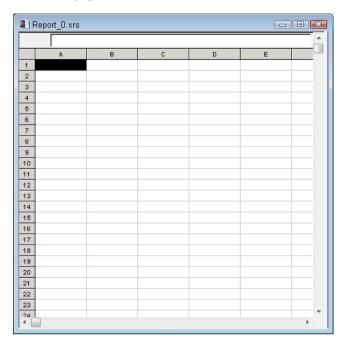
Before a report is created, the report printer should be defined because the page formatting of a report depends on the printer that has been set. Subsequent changing of the printer can necessitate reformatting of the existing table.

In order to create a new report:

▸ Navigate to the report generator node in the project manager

▸ Select the **New report** command in the context menu

▸ a new report is created in the detail view

▶ An empty worksheet is created in the main window

Functionalities of the worksheet:

▶ Assignment of process information using report functions (on page 45)

▶ Formatted display of content of cells, columns and rows

▶ Subtotals

▶ Assignment of editing blocks

💡 Info

The following applies for the input in cells: A decimal value can be entered with a `comma` as well as with a `period` as a separator. The decimal separator is automatically changed to a `period`.

🔍 See also

Set up page (on page 23)
Printer settings (on page 18)
Formatting (on page 22)

Report menu

The `Report` menu is shown when a `report` is selected:

Parameter	Description
Save report	Saves all changes in the active report. Changes in other reports are not saved.
Export report functions	Exports all report functions from the active report to a text file. Structure: Column / row =function(parameter)
Refresh report	Draws a new report.
Calculate report	Calculates a new report.
Set up page (on page 23)	Opens the dialog to set up the print page.
Print report	Prints the report using the printer defined in the standard settings..
Print area	Defines printed areas or cancels them: ▸ Define printing area: Defines marked areas of the table as a print area. ▸ Release printing area: Cancels defined print area
Report - Page view	Switches from layout view to page view. The view can be zoomed using the left mouse button. Leaving page view: Press Esc key.
Report file Info (on page 20)	Opens the dialog to administer the report file information.

Menu Edit

When a report is selected, the `Edit` menu offers special functions for reports:

Command	Description
Cut	Cuts out highlighted area and copies it to the clipboard.
Copy	Pastes the contents of the clipboard.
Insert	Copies highlighted areas to the clipboard.
Paste enhanced	Provides advanced paste options for cell operations:
▸ All	Inserts everything from the clipboard.
▸ Formulas	Only inserts formulas
▸ Value	Only inserts values. The relationship to formulas is lost.
▸ Format	Only accepts the format.
Fill	Copies the content of the first cell of a marking and transfers it to all further marked cells.
▸ down	Filling is carried out in a downward direction.
▸ right	Filling is carried out from left to right.
Remove	Removes from highlighted cells:
▸ All	Everything.
▸ Contents	Contents only.
▸ Format	Formats only.
Delete column/row	Deletes highlighted columns or rows.
Insert column/row	Inserts new columns or rows. Insertion is carried out above the row or to the left of the column. As many rows/columns are inserted as are highlighted in the worksheet.
Search/replace	Opens the dialog (on page 20) for searching and replacing in a worksheet.

Format menu

The Format menu provides properties and functions for formatting the report table:

Parameter	Description
Cells	Formatting options for cells (on page 29): ▸ Font ▸ Pattern ▸ Border ▸ Alignment ▸ Numbers ▸ Lists
Line	Formatting options for rows: ▸ Row height: Opens dialog to define the row height ▸ Row format: Opens dialog for all row properties (on page 42)
Column	Formatting options for columns: ▸ Column width: Opens dialog to define the column width: ▸ Column format: Opens dialog for all column properties (on page 41)
Report	Opens the dialog (on page 26) to format the report.
Function wizard	Opens assistant to create report functions (on page 45).
Filter	Opens report filter (on page 43) to select archives.

Toolbar Report Generator

A toolbar is available in the Editor for the Report Generator. If it is not displayed yet, you can load it:

1. Open the menu `Options -> Settings -> Tab: Toolbar`

2. Tick the checkbox next to Report Generator

3. confirm this with `OK`.

Symbol (from left to right)	Function
Print report	Prints the report on the default printer.
Report – Page view	Opens the report in a preview. The report can be enlarged or minimized in the preview (zoom), you can leaf through it and it is printable.
Refresh report	Refreshes the display.
Recalculate functions	Recalculates the formulas in the report.
Function wizard	Opens a dialog that helps with the configuration of functions.
Font	Opens the dialog for cell formatting - focused on the **Font** tab.
Text alignment	Opens the dialog for cell formatting - focused on the **Alignment** tab.
Border	Opens the dialog for cell formatting, with focus on the tab **Frames**.
Pattern	Opens the dialog for cell formatting - focused on the **Pattern** tab.
Options for symbol bar	Clicking on the arrow opens the submenu: Active: Tool bar is displayed If the toolbar is not displayed, it can be activated using the Menu *Options -> Toolbar*. Note: For free placed tool bar (undocked from the Editor) options are not displayed. The tool bar can be closed by clicking on button X.

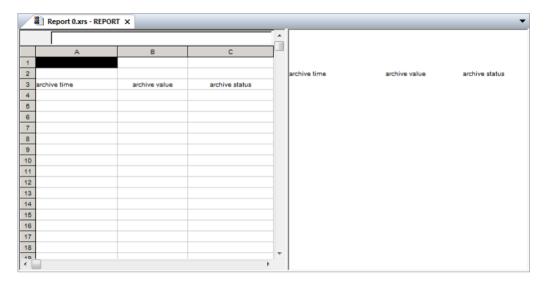**Detail view of context menu and toolbar**

Menu item	Action
`New report`	Creates a new report in the list in detail view.
Open report	Opens the selected report or focuses on the one that is already open.
Create standard function (on page 158)	Creates a zenon function for report generator. The action is documented in the output window.
Jump back to starting element	If you entered the list via function `linked elements`, the symbol leads back to the start element. Only available in the context menu when all linked elements are opened.
Copy	Copies selected entries to the clipboard.
Insert	Pastes the contents of the clipboard. If an entry with the same name already exists, the content is pasted as "Copy of".
Delete	Deletes selected entries.
`Import existing report...`	Opens the File Manager to select an existing file.
Remove all filters	Removes all filter settings.
Edit selected cell	Opens the selected cell for editing. The binocular symbol in the header shows which cell has been selected in a highlighted line.
Replace text in selected column	Opens the dialog for searching and replacing texts.
Properties	Opens the property window.
`Help`	Opens the online help.

Creation of table window

Reports are displayed in two parts when being edited in the Editor in the main window:

- ▸ The left area contains the table for the draft

- ▸ The right part displays a preview of the output

The size of the areas can be amended individually. To do this, move the mouse to the right edge of the table scroll bar until the mouse pointer becomes a moving tool and move the scroll bar to the desired position.

INPUT ROW

The table has an input row at the upper edge. If, in the report Properties (on page 27), the `Edit in the cell` property is activated, this row is not displayed. Input must then be made in the row itself.

Data sheet context menu

Menu item	Action
Cut	Deletes highlighted entries and places them on the clipboard.
Copy	Copies selected entries to the clipboard.
Insert	Pastes the contents of the clipboard.
Expand insert	Enables the contents of the clipboard that are to be inserted to be selected: ▸ All: transfers complete contents ▸ Formula: Transfers only formulas ▸ Value: Transfers only the value ▸ Format: Transfers only the format
Fill	Copies the content of the first cell of a marking and transfers it to all further marked cells. It is possible to select if they are filled downwards or to the right.
Remove	Makes it possible to select the content of a cell to be deleted: ▸ All: Deletes the complete contents with format ▸ Content: Removes the content, but leaves the format ▸ Format: Deletes only the format, but leaves the content
Insert column row	Inserts a new row above a highlighted row or a new column to the left of a highlighted column. If several rows/columns are highlighted, the number of highlighted rows/columns is inserted.
Delete column row	Deletes the highlighted rows or columns
Report...	Opens the dialog to format the report.
Cell...	Opens the dialog to format the a cell.
Column...	Opens the dialog to format the columns.
Row...	Opens the dialog to format the rows.
Functions...	Opens the assistant to create report functions (on page 45). Note: The assistant offers a selection of over 150 functions. Further functions can be entered manually into the input line of the data sheet.

Printer settings

The report layout (side margins, page sizes) depends on the printer configured. You can find the settings for the report printer via *File -> General configuration -> Standard -> Printer for values to be given and protocols.*
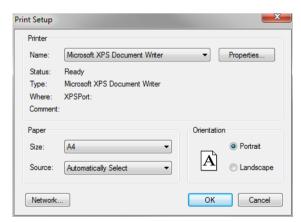
Here you define

▶   the printer and

- Paper format of the print
- Paper source of the printer
- Direction of print

> ⚠ **Attention**
>
> *The desired printer must be selected before creating the report, because the page formatting of a report depends on the printer that has been set. Subsequent changing of the printer can necessitate reformatting of the existing table.*
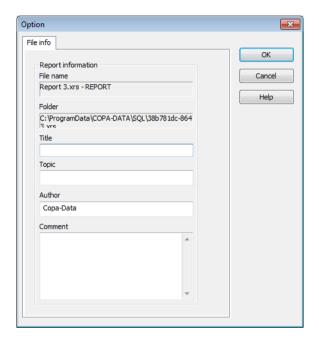
PRINTER CONFIGURATION

Parameters	Description
Printers	Settings for the printer.
Name:	Selection of the printer from the drop-down list. The list contains all printers configured in the operating system.
Properties...	Opens printer configuration dialog.
Status:	Display printer state. For information only.
Type:	Display printer type. For information only
Location:	Display the location of the printer if configured. For information only.
Comment:	Display comment about printer if configured. For information only.
Paper	Configuration of the printout.
Size	Select paper format from drop-down list.
Source	Select paper feed from drop-down list.
Alignment	Select paper alignment. Possible parameter: • Portrait Format • Landscape format
Network	Opens dialog for selecting a printer in the netwok.
OK	Accepts configuration and closes dialog. With this printing is started in the Runtime.
Cancel	Discards configuration and closes the dialog. In the Runtime this also cancels the printout.

File info

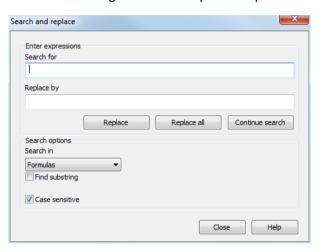In the *Report -> Report file info* dialog, information on the respective report file can be saved.

Parameter	Description
File name	Name of the report file. Information only; cannot be changed here. Can be changed in the detail view.
Folder	Save location of the report file. Information only; cannot be changed here.
Title	Free text input
Topic	Free text input
Author	Free text input
Comment	Free text input

Finding and replacing in the report

To find formulas, character strings or values and replace these:

1. Activate the desired report

2. Select **Find/replace** in the **Edit** menu.

3. The dialog to find and replace is opened

Search and replace

Enter expressions
Search for

Replace by

[Replace]   [Replace all]   [Continue search]

Search options
Search in

Formulas ▼

☐ Find substring

☑ Case sensitive

[Close]   [Help]

Parameter	Description
`Enter expressions`	Input of character string for finding and replacing.
Search for	Character string that is being searched for.
Replace by	Character string that replaces the one that is found.
`Replace`	Replaces currently displayed finding with sequence from `Replace with`.
`Replace all`	Replaces all currently displayed findings with sequence from `Replace with`.
`Continue search`	Looks for next instance of the character string being searched for.
`Search options`	Search options
Search in	Select from drop-down list: ▸ Formulas: Searches in formulas ▸ String: Searches in character strings ▸ Value: Searches in values
Find substring	Also searches partial areas of cells.
Case sensitive	Note capitalization during the search.
Close	Closes the dialog.
`Help`	Opens online help

Formatting

Before creating a report, the necessary formatting should be carried out:

▸ **Page (on page 23):** Set up print properties for the page

▸ **Report (on page 26):** Set up report properties

▸ Cell (on page 29): Define the formatting of the cells (direction, border, font, pattern, type, lists)

▸ Columns and cells (on page 40): Formatting for selected columns or cells

You can find the settings in the `Report` and `Format` drop-down lists in the menu bar or in the context menu of the worksheet.
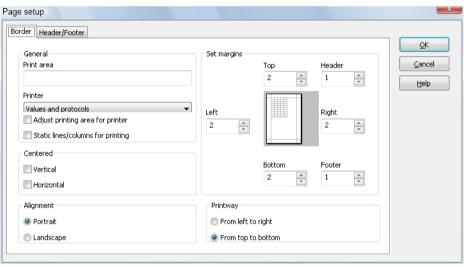
Hint: Ensure that reports with large amounts of data have sufficient lines/columns for output.

Set up page

The `Set up page...` command in the `Report` menu opens the dialog to configure the report page. This contains the two tabs:

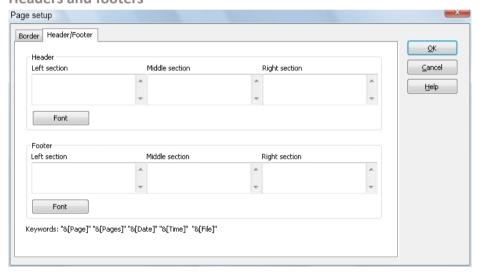▶ Border (on page 23)

▶ Headers and footers (on page 25)

Border

Parameter	Description
Set margins	Define print margin independently of the printer setting (on page 18)
`top`	Distance of the report from the upper edge.
`bottom`	Distance of the report from the lower edge.
`left`	Distance of the report from the left edge.
`right`	Distance of the report from the right edge.
`Header`	Distance of the header from the text. Possible values: `0` to `20`; `0` = hidden. The value selected must be large enough to be able to display the selected font size under header and footer (on page 25).
`Footer`	Distance of the footer from the text. Possible values: `0` to `20`; `0` = hidden. The value selected must be large enough to be able to display the selected font size under header and footer (on page 25).
General	
`Print area`	Defines the print area based on the cell `A1`. Input in the text field: `Coordinates of the end value` of the range. Example: `D5` defines the print area of the cells `A1` to `D5`.
`Printer`	selection of the printer for the report. Clicking on the drop-down list opens the list of all defined printers available in zenon printer settings. Default: `Values and protocols`
`Adjust printing area to printer`	`Active`: The printing area of the report is adjusted to the settings of the selected printer.
`Static lines/columns for printing`	`Active`: Static lines and columns are also printed.
Centered	Orientation of the report on the print page

vertical	Centers the report vertically.
horizontal	Centers the report horizontally.
Alignment	Settings of page format
Portrait	Prints in portrait format.
Landscape	Prints in landscape format.
Printway	Sequence of the page print out for multiple pages
From left to right	Print out several pages from left to right.
From top to bottom	Print out several pages from top to bottom.

Headers and footers

Parameter	Description
Header	Definition of the header area
Left section	Left text for the left third of the header.
Middle section	Centered text for the center of the header.
Right section	Right text for the right third of the header.
Font	Opens the dialog to select the font, font size and emphasis of the footer text.
Footer	Definition of the footer area
Left section	Left text for the left third of the footer.
Middle section	Centered text for the center of the footer.
Right section	Right text for the right third of the footer.
Font	Opens the dialog to select the font, font size and emphasis of the footer text. Note that the settings for the header/footer can be selected in borders (on page 23) in such a way that the font size can be displayed in full.
Keywords	Key words for headers and footers
&[Page]	Page number
&[Pages]	Total number of pages
&[Date]	Printing date
&[Time]	Printing time
&[File]	File name

Format report
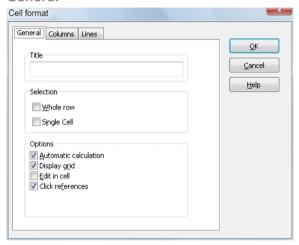
For formatting the whole report, there are settings for:

▶ General (on page 27): General settings for the report

▶ Columns (on page 28): General settings for columns, such as number etc.

▸ Rows (on page 28): general settings for rows, such as number etc.

To format, select either the `Report` command in the `Format` menu or the `Format report` command in the context menu of the report sheet.
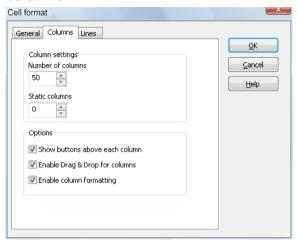
General

| Parameter	Description
Title	Table heading
Select	Defines selection behavior when clicked.
whole row	Active: The whole row is selected.
one cell	Active: Only the cell that has been clicked on is selected; multiple selection of cells is not possible.
Options	
Automatic calculation	Active: Calculations are automatically updated when input values are changed.
Show grid	Active: Shows grid network lines in the editor.
Edit in cell	Active: Entries are made in the cell directly; the input cell in the report window is shaded out.
Click references	Active: The cell that has been clicked is accepted as the report coordinate when formulas are entered. For example: Instead of entering A12, the A12 cell is clicked.
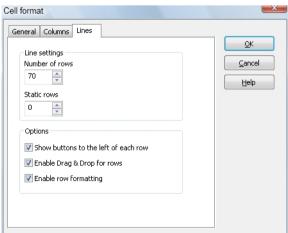
Columns

Parameter	Description
Column settings	
Number of columns	Defines the maximum number of columns. Valid value: `0` to `31000`.
Static columns	Number of columns that are always shown. Scrolling only takes place after the static columns.
Show buttons above each column	`Active`: Shows target numbers.
Enable column formatting	`Active`: The column width can be adjusted using the mouse by dragging the borders of the header.

Lines

Configurable features of the rows are:

Parameter	Description
Number of rows	Defines the maximum number of rows. Valid value: 0 to 31000.
Static rows	Number of rows that are always shown. Scrolling only takes place after the static columns.
Show buttons to the left of each row	Active: Shows column header >(A, B, C, ...).
Enable row formatting	Active: The row width can be adjusted using the mouse by dragging the borders of the row numbering.

Format cell

For formatting the cells of the report sheet, there are settings for:

Alignment (on page 30)

Border (on page 31)

Font (on page 32)

Pattern (on page 33)

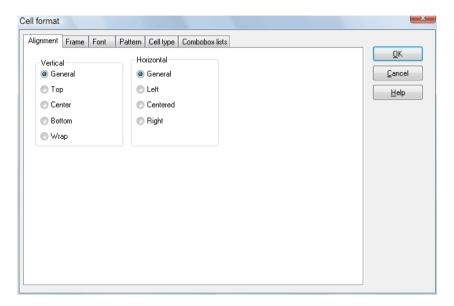Cell type (on page 33)

Lists (on page 38)

For formatting, select:

▸ The `Format cell` command in the context menu of the report sheet to open the dialog

▸ or the desired sub-menu in *Report -> Format cells,* to open the dialog with the corresponding tab

Alignment

The display of cell contents is defined in the `Orientation` tab.

The following are available for vertical positioning:

- ▶ General

- ▶ Top

- ▶ Center

- ▶ Bottom

- ▶ Wrap

The following are available for horizontal positioning:

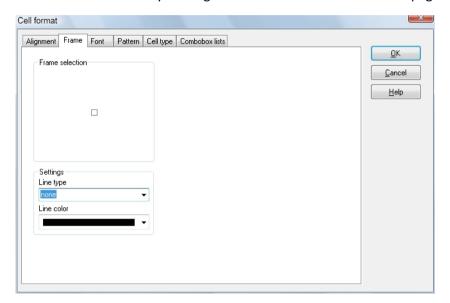- ▶ General

- ▶ Left

- ▶ Center

- ▶ Right

> **Info**
>
> *If text is to be displayed vertically, select the* `vertical` *property in the* **Font** *property.*

Border

In the Frame tab, you define if a frame is displayed around a cell and if so, in what form. You can define each side of the frame (top, bottom, left, right) in terms of type and color:

1. Define the type and color.

2. Allocate these by clicking the whole frame or an individual page.
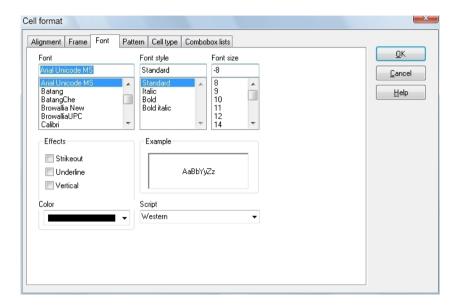

Parameter	Description
Border selection	Clicking in the center activates all our frame sides. Clicking on the frame side activates these.
Line type	Selection of line type from drop-down list.
Line color	Selection of line color from drop-down list.

Font

The standard Windows settings are available for fonts.
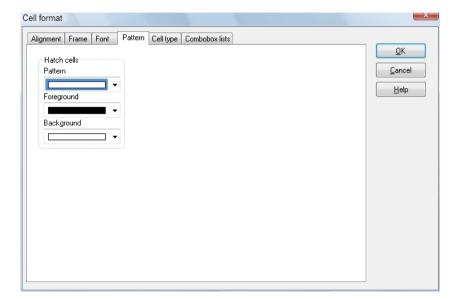
Parameter	Description
Font	Selection of the font.
Font style	Selection of the aspect.
Font size	Selection of aspect in point.
Effects	
Struck out	Active: Text is displayed as strike-through in the cell.
Underlined	Active: Text in the cell is underlined.
Color	Is not taken into account! The color must, for example, be set in a dynamic element when using the font
Script	Is not taken into account! Unicode display is used throughout.
Example	Shows effects of the selection on the text in cells.

Pattern

You define the color and pattern of the cell in the pattern tab.

Parameter	Description
Pattern	Selection of cell pattern from the drop-down list.
Foreground	Selection of foreground color from drop-down list.
Background	Selection of background color from drop-down list.

Cell type

Define the output format in the `cell type` tab. You have the choice between:

▸ General: Output of values, data and binary data
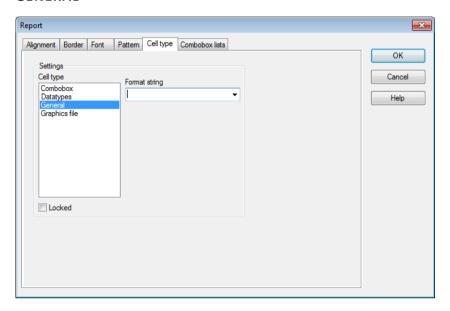
▸ Combo box: Create lists

▸ Data type: Definition of data types

▸ Graphics file: Inserting graphics

The respective format key is described in the Formatting key (on page 154) chapter.

GENERAL

| Parameter	Description
`Format strings`	Selection of the output format
▸ `Value`	Numerical value. For details, see the Numerical data types (on page 154) chapter.
▸ `Date`	Date information. For details, see the Data types for date and time (on page 156) chapter.
▸ `Time`	Time information. For details, see the Data types for date and time (on page 156) chapter.
▸ `Binary value`	binary value. For details, see the Logical data types (on page 155) chapter.
`Locked`	`Active`: Locks the cell for editing in the editor and locks entries in the online report.

📋 Example

Format string for value output:
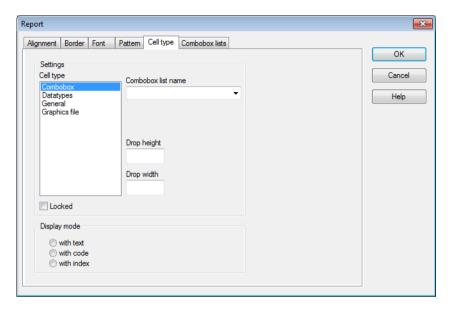
`*0.00` = *any desired value with two decimal points*

`dd/mm/yy hh:mm` = *01/01/10 12:30 (corresponding to the settings in the operating system*

COMBOBOX

The combobox cell type offers drop-down lists to select pre-defined inputs. Before this cell type can be used, a list must be created in the combobox lists (on page 38) tab.
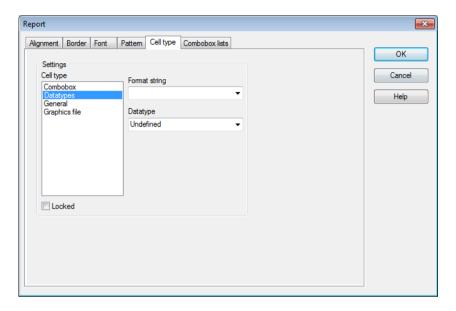
Parameter	Description
`Combobox lists name`	Name of the list file. This must already be defined in the Combobox lists (on page 38) tab.
`Drop height`	Display height
`Drop width`	Display width
`Locked`	`Active`: Locks the cell for editing in the editor and locks entries in the online report.
`Display mode`	Type of display
▸ `with text`	Alphabetically sorted according to text
▸ `with code`	Sorted according to code
▸ `with index`	Sorted according to the order of input of the list entries

DATA TYPE

An input template is created in the report with the data type. Existing functions are removed from this cell. The user receives instructions on how data is to be entered into the cell.
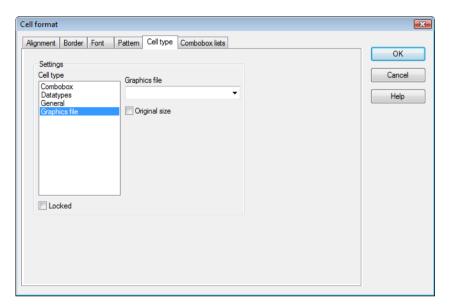
Parameter	Description
Format string	Input of format, depending on data type.
Data type	
▸ Date/Time	Input of a date/time value. For format string details, see the Data types for date and time (on page 156) chapter.
▸ Float	Input of a float value. For format string details, see the Numerical data types (on page 154) chapter.
▸ Integer	Input of an integer value. For format string details, see the Numerical data types (on page 154) chapter.
▸ Logical	Input of a binary value. For format string details, see the Logical data types (on page 155) chapter. Configurable: Yes/no; right/wrong.
▸ Text	Input of text.
▸ Undefined	No data type defined.

Locked	**Active**: Locks the cell for editing in the editor and locks entries in the online report.

GRAPHICS FILE

Graphics files must be created before they are selected in the project manager: *File nodes -> Graphics -> [context menu] Add files.*
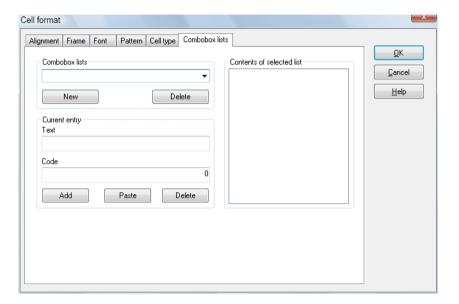
Parameter	Description
Graphics file	Selection of graphics file. BMP, JPG, GIF, PNG and WMF formats are supported.
Original size	**Active**: The graphics are displayed in original size. **Inactive**: The graphics are adapted to the size of the cell. Default: `inactive`.
Locked	**Active**: Locks the cell for editing in the editor and locks entries in the online report.

Combobox lists
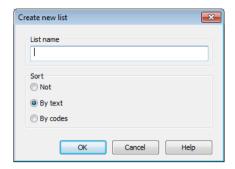
In the **Combobox lists** tab, lists for the cell type **Combobox** are defined.

Parameter	Description
Combobox lists	Drop-down list to select existing lists.
New	Opens the dialog to create a new list.
Delete	Deletes selected list.
Contents of selected list	Shows contents of the selected lists
Current entry	Configuration of the entry.
Text	Input of the text, as it should appear in the list in the Runtime.
Code	Input of a number for sorting the lists. 0: no sorting according to code. There is a choice of sorting lists in Runtime alphabetically or in the order of input. Minimum value: 0 Maximum value: 9999999999
Add	The new entry is inserted depending on the settings in **Content of selected lists** that were selected when the list was created. ▶ Not: Entry is inserted at the end of the list. ▶ According to text: Alphabetic sequence. ▶ According to codes: Code number determines the sequence (increasing) in lists.
Insert	The new entry is added above the entry selected in the **Content of selected lists**:
Delete	Deletes the entry selected in **Content of selected lists**.

CREATE NEW LIST

Parameter	Description
`List name`	Name of list
`Sort`	Sorting criteria for display in **Content of selected lists** and in Runtime
`not`	Entry is inserted at the end of the list.
`According to text` (default)	Alphabetic sequence
`By codes`	Code number determines the sequence (increasing) in lists.

Format columns and cells

If an entire row or an entire column is highlighted, the additional commands **Format column** and/or **Format row** are available in the context menu.

As with **Format cell**, tabs are available for:

Alignment (on page 30)

Border (on page 32)
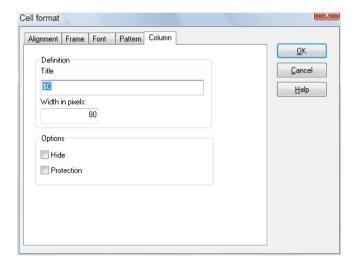
Font (on page 32)

Pattern (on page 33)

Note: These settings always only concern the column heading and/or the row heading. They have no effect on the other cells of a column or row. These must always be formatted using the **Format cells** (on page 29) dialog.

There is also a further Column (on page 41) or Row (on page 42) tab available in this dialog.

Column

In the Column tab, properties that concern the whole column are defined:
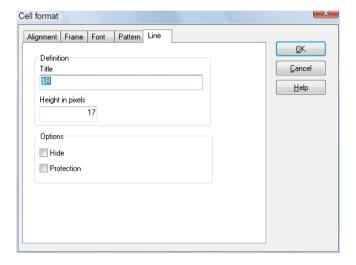
Parameter	Description
Definition	
Title	Defines column heading. $C: Letters in a rising sequence of column numbers E: The column automatically expands to the correct number of columns if an **archive***, **aml** or **cel** is present in one of the cells. If other report functions are used, "E" may not be used. "E" can be used in a report several times. Default: $C
Width in pixels	Width of the column in pixels
Options	
Hide	Active: The column is hidden. Inactive: The column is shown. To show a column that has been hidden: Highlight the two neighboring columns and set the Hide property to inactive. **Attention:** Functions that are in the hidden cells are not executed. To execute a function without it being shown: Set the row height or column width to 0.
Protection	Active: The cells in the column are protected from changes of formatting. Inactive: The cells of this column can be formatted.

Line

This dialog allows defining the properties of the entire selected row.
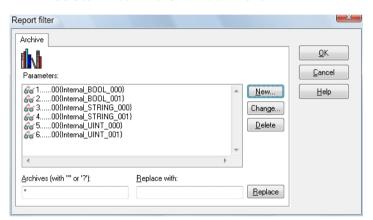
Parameter	Description
Definition	
Title	Defines row heading. $R: Numbers in reverse order of column number E: The row automatically expands to the correct number of rows if an **archive***, **aml** or **cel** is present in one of the cells. If other report functions are used, "E" may not be used. "E" can be used in a report several times. Default: $R
Width in pixels	Height of the row in pixels.
Options	
Hide	Active: The row is hidden. Inactive: The row is displayed. To show a row that has been hidden: Highlight the two neighboring rows and set the Hide property to inactive. **Attention:** Functions that are in the hidden cells are not executed. To execute a function without it being shown: Set the row height or column width to 0.
Protection	Active: The cells in the row are protected from changes of formatting. Inactive: The cells of this row can be formatted.

Set up archive filter

Should archive values be used in the report then these are to be defined before activating the report functions. To set up the archive filter:

▶   Select **Filter** in the **Format** menu.
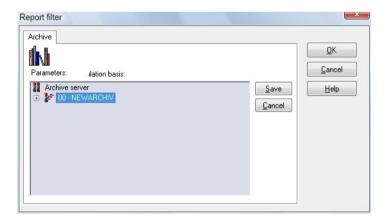
Parameter	Description
Parameter	List of variables selected from the archives. The numbers in front of the variables are used in the archive and protocol functions (on page 49) for classification.
New	Opens the dialog for selecting variables.
Change	Opens the dialog for selecting variables.
Delete	Deletes selected variable from the **Parameters** list.
Archives	Definition of a variable filter. Wildcards * and ? are permitted at the start of an expression.
Replace by	Enter the expression that is to replace the string that is being searched for
Replace	Carries out a search and replace. The result is displayed in a message box.

"NEW FILTER" OR "CHANGE FILTER"

Clicking on the **New** or **Change** buttons opens the dialog to select the variables:

Parameter	Description
Parameter	Display of the archives present. It is possible to select a complete archive or individual variables. Each variable automatically receives an increasing number, which is addressed in the report functions for archive (on page 49). A report can therefore also be used throughout projects.
Save	Provides selected archives/variables for the report.
Cancel	Leaves dialog without selection.

Each variable automatically receives an increasing number, which is addressed in the report functions for archive (on page 49).

Report functions

For the formation of the report, different formatting and calculation functions are available.

▶ `Processing functions:`

Defined access to selected zenon data (online and archive values, user, computer names etc.)

▶ `Report functions`

- formatting functions
- calculating functions

The following groups of functions are available:

▶ Database functions (on page 78)

▶ Date and time functions (on page 86)

▶ Logical functions (on page 91)

▶ Archive and protocol functions (on page 49)

▶ Statistical functions (on page 116)

▶ Mathematical and trigonometric functions (on page 97)

▶ Text functions (on page 126)

> ▶   Recipegroup Manager functions (on page 112)

> ▶   Other functions (on page 135)

The functions that are used most often can be configured using the function assistant (on page 48).

ENTER

The input of the functions is done in the input row of the report.

If several cells are to receive the same input/function:

1. Highlight the starting cell and the target cell

2. Select the **Fill** command in the context menu and the desired direction (`right` or `down`)

References to cell co-ordinates in the function are automatically adjusted.

EXAMPLE FOR INPUTS

Parameter	Description
`=archiv(1,value,down wards)`	All values of the archive variable 1 are displayed downwards from the cell.
`=sum(A4:D4)`	The contents of cells A4 to D4 are added together. The cell must be formatted as a value.
`=now()`	Provides the current date/the current time.

Info

Report functions can lead to error messages in the editor if not all data required to calculate the required data is properly available. To check to see that the configuration is correct, test the report in Runtime.

Direction

Options	Meaning
`right`	Value is entered to the right (row).
	There must be sufficient cells left free for expected entries from further report creation.
	Attention: If an `archivesp` function is created with only one column, the value is entered downwards (in the column).
`bottom`	Value is entered downwards (column).
	There must be sufficient cells left free for expected entries from further report creation.
	Attention: If an `archivesp` function is created with only one row, the value is entered to the right (in the row).

Info

The Report Generator module has to be licensed in the development environment and in the Runtime (standalone, server/standby and clients).

If the module is not licensed in the Runtime, the following functions are not available:

▶ Save archives (for manual inputs)

▶ and writing set values

Read access is possible.

Conventions

The following conventions are valid:

Parameter	Description
`Number:`	`(num);` divider for decimal points depends on the settings in the operating system (1,0 or 1.0).
`Area:`	(A1:A10)
`String:`	Character sequence is separated by a dual apostrophe (**"**), for example "ABC".
`Radians:`	The input of goniometric functions is done in radians.

CONVERSION OF DEGREES INTO RADIANS

degrees	arc measurement	radians
0.00	0	0
45.00	p/4	0.78539816339744828
90.00	p/2	1.5707963267948966
135.00	3p/4	2.3561944901923448
180.00	p	3.1415926535897931
225.00	5p/4	3.9269908169872414
270.00	3p/2	4.7123889803846897
315.00	7p/4	5.497787143782138
360.00	2p	6.2831853071795862

Function wizard
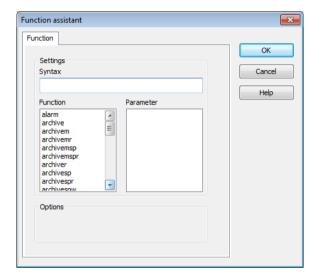
The function assistant supports the planning of report functions. It makes the most frequently used functions available. All other functions are manually entered into the input cell.

To open the function assistant:

1. Click in the desired cell

2. in the context menu, select the `Functions` command

3. The assistant is opened

4. The selection on the assistant is transferred to the input cell;

   any quotation marks that are open may need to be closed manually

Parameter	Description
Syntax	Result of the selection of function and parameter.
Function	Selection of the function.
Parameter	Selection of parameter that is available for the selected function.

Archive and logging functions

BASICS

All archive functions build on the basic `archive(FILTERINDEX,PARAMETER,RICHTUNG)` function.
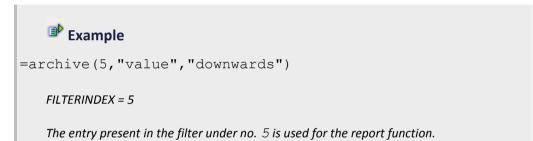
The expansions are distinguishable by additional letters in function names.

Code	Meaning	Comment
r	read only	There is only read access to the archive. Changed values cannot be saved.
sp	Columns	Provides special formatting to define cells to be used.
min	Mathematics	Only has an effect on value and time.
ex	extended	provides special treatments.
w	write	Writing is possible.

FILTER INDEX

The filter index creates a distinction between archive filter numbers (on page 43) used in the report functions and variables from the archives.

Example

```
=archive(5,"value","downwards")
```

FILTERINDEX = 5

The entry present in the filter under no. 5 is used for the report function.

PARAMETER

Options	Possible value
value	[1-n]
status	[1-n]
time	[1-n]
number	[1]

Direction

Options	Meaning
`right`	Value is entered to the right (row). There must be sufficient cells left free for expected entries from further report creation. **Attention:** If an `archivesp` function is created with only one column, the value is entered downwards (in the column).
`bottom`	Value is entered downwards (column). There must be sufficient cells left free for expected entries from further report creation. **Attention:** If an `archivesp` function is created with only one row, the value is entered to the right (in the row).

All archive functions can be created with the function assistant (on page 54).

On outputting the status information, the settings from the `project.ini` file in the **[STATUS]** section or the settings from the internally defined allocation are used. If the status labeling was changed in `project.ini`, the changed name is given. (Tip: makes it possible to keep the project compatible with old status labels.)

In online operation, the query time is either defined in the function query (on page 158) or given in Runtime when a report file is opened.

CHANGE ARCHIVE ENTRIES

If permitted by user authorizations, archive values can also be changed in Runtime. Changes are either saved manually via the `Save` control element or automatically saved when saving a report file. The status of an amended entry is automatically set to `manual value`.

> ### 💡 Info
>
> *Changing string fields in the report: If a text consists only of numbers, spaces are removed by default when it is saved. Spaces that are desired must be displayed by ASCII character 255 (`Alt+255`).*

Handling of archive entries

Archive entries can be displayed in a report and also be changed and saved in the archive again with sufficient user authorizations. There is a function assistant (on page 54) available to create the report functions.

READING AND WRITING ARCHIVE ENTRIES

The following contents of an archive entry can be read for display in a report:

- ▶ Value

- ▶ State

- ▶ `Time`

The following archive entries can be changed and saved via calculations and new entries in the report:

- ▶ Value

- ▶ `State`

Amended archive entries are saved in Runtime using the `Save archive` button. Saving changes can be stopped by:

- ▶ Activating the `Read only` option in the zenon `Screen switching` function for the report screen

- ▶ Activating the `Locked` option in the format (on page 33) dialog cell

If the value of an archive entry is changed, the status is set as `MAN_VAL` when it is saved in the archive.

Exception: If the status of the archive entry is changed, only the changed status is entered. If safety regulations such as those of the FDA are to be met, no status change may be permitted as a result of the report when configuring. Archive entries with `MAN_VAL` status are displayed in `blue` in a report.

REPRESENTATION

Archive entries can be represented as archive value rows or as row-formatted archive value rows. The archive entries which lie in the time range of the opened log are represented rightwards or downwards from the cell in which the archive function is configured. If no archive entry is found then a – is output in the cell in which the archive function is defined.

ARCHIVE VALUE ROWS

The read-in archive entries are shown as a row, one after the other, without blank spaces.

ROW-FORMATTED ARCHIVE VALUE ROWS

The read-in archive entries are shown with a defined number of:

- ▶ Archive entries per row

- ▶ Blank spaces between two entries

- ▶ Empty rows between the rows.

.

CONTROL CELLS

The parameters   P1, P2 and P3 set out which cells of the report can be filled:

- ▶ P1: Number

- ▶ P2: Empty cells between two value outputs, depending on DIRECTION (down, right) either empty rows or empty columns

- ▶ P3: Empty cells between two value outputs, depending on DIRECTION (down, right) either empty rows or empty columns

All entries of the time range of the report are shown in the report, going downwards or tothe right from the cell in which the function was given, with the row format defined by the parameters P1-P3.

WHEN SELECTING DOWN, THE FOLLOWING APPLIES:

- ▶ P1 = number of rows to be filled with values

- ▶ P2 = empty rows between the values

- ▶ P3 = empty columns

WHEN SELECTING RIGHT, THE FOLLOWING APPLIES:

- ▶ P1 = number of columns to be filled

- ▶ P2 = empty rows between the values

- ▶ P3 = empty rows between the values

Configuration with function assistant

You can configure report functions for the archive with the function assistant:

1. Select the desired cell in the context menu with the `Functions` command.

2. The assistant is opened.

3. Select the desired archive function.
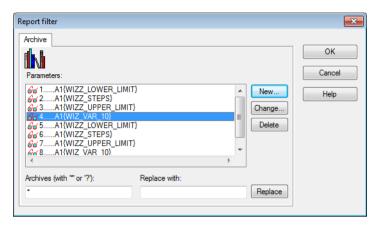
4. The archive function is entered into the syntax.

5. Clicking on archive opens the archive filter (on page 43).
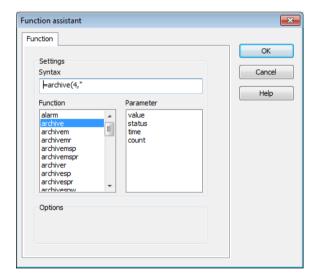
6. Select the desired entry.

7. The reference number of the entry (for example `4`) is transferred to the string of the syntax.

8. Select the desired parameter, for example `value`).
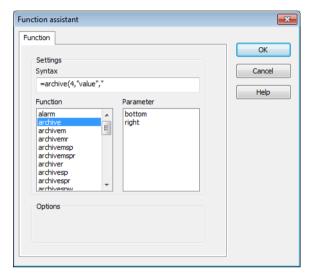
9. The parameter is transferred into the string.

10. Select the desired direction of the entry in the table:

- `below`: The function fills the column
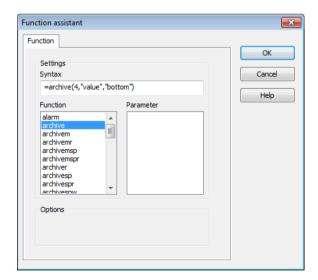
- `right`: The function fills the row

  Attention: When using the `archivesp` function, corresponding columns/rows are envisaged! If for example, to the right is selected, but there is only one column available, then the values are entered downwards (in the only column available).

11. The direction is transferred into the string.

12. The brackets are closed.

13. The function is transferred to the cell by clicking on OK.

archive

Value, Status or time of the archive entry can be read. Changes are either saved manually via the `Save` control element or automatically saved when saving a report file. The status of an amended entry is automatically set to `manual value`.

`Syntax: =archive`(FILTERINDEX,PARAMETER,DIRECTION)

Transfer parameters	Valid range	Comment
FILTERINDEX		Corresponding entry from the archive filter (on page 43).
PARAMETER	value, status, time, count	The selected parameter is read from the archive and displayed.
DIRECTION	bottom, right	Defines the direction in which filling is to take place, starting from the cell in which the function was created.

Example

In a report, all archive values from the archive variable 1 should be given downwards.

=archive(1,"value","downwards")

archiveex

`Syntax:`
`=archiveex`(FILTERINDEX,PARAMETER,DIRECTION,STATUS,SUBSTITUTETEXT,NN_TEXT,NN_SUBSTITU
TEVALUE,CYCLEOFFSET)

Transfer parameters	Valid range	Comment
FILTERINDEX	[1-n]	Corresponding entry from the archive filter (on page 43).
PARAMETER	value, status, time, count	The selected parameter is read from the archive and displayed
DIRECTION	bottom, right	Filling in the corresponding direction.
STATUS	SPONT, MAN_VAL, INVALID	Status of the values to be displayed.
ALTERNATETEXT		Text to be displayed if the status of the archive variables does not correspond to the STATUS.
ND_TEXT		Text for the "non-detectability limit"; not case-sensitive.
ND_ALTERNATEVALUE		Cell reference or entered value are possible. If the ALTERNATETEXT is entered into the cell and then the table is saved, the NN_ALTERNATEVALUE is written to the archive.
CYCLEOFFSET		Is only taken into account for filter type: ▸ Time, day ▸ Time, month ▸ Time, year ▸ Time, week ▸ Time, 2week ▸ Time, 15min. ▸ Time, 30min. ▸ Time, 60min.

Example

=archiveex(1,"time","downwards",SPONT,MAN_VAL,"-","NN",G2,-2)

archiveexr

Syntax:

=archiveexr(FILTERINDEX,PARAMETER,DIRECTION,STATUS,SUBSTITUTETEXT,NN_TEXT,NN_SUBSTITUTEVALUE,CYCLEOFFSET)

Transfer parameters	Valid range	Comment
FILTERINDEX	[1-n]	Corresponding entry from the archive filter (on page 43).
PARAMETER	value, status, time, count	The selected parameter is read from the archive and displayed
DIRECTION	bottom, right	Filling in the corresponding direction.
STATUS	SPONT, MAN_VAL, INVALID	Status of the values to be displayed.
ALTERNATETEXT		Text to be displayed, if the status of the archive variables does not correspond to the STATUS.
ND_TEXT		Text for the "non-detectability limit"; not case-sensitive.
ND_ALTERNATEVALUE		Cell reference or entered value are possible. If the ALTERNATETEXT is entered into the cell and then the table is saved, the NN_ALTERNATEVALUE is written to the archive.
CYCLEOFFSET		Is only taken into account for filter type: ▸ Time, day ▸ Time, month ▸ Time, year ▸ Time, week ▸ Time, 2week ▸ Time, 15min. ▸ Time, 30min. ▸ Time, 60min.

archivem

This function makes it possible to automatically change values or times of an archive entry by adding a constant to a value. Changes are either saved manually via the Save control element or automatically

saved when saving a report file. The status of an amended entry is automatically set to `manual value`.

Syntax: =archivem(FILTERINDEX,PARAMETER,DIRECTION,MATH)

Transfer parameters	Valid range	Comment
FILTERINDEX	[1-n]	Corresponding entry from the archive filter (on page 43).
PARAMETER	value, status, time, count	The selected parameter is read from the archive and displayed
DIRECTION	bottom, right	Filling in the corresponding direction.
MATH		▸ Constant: signed ▸ Decimal points: Point separator (.) ▸ Correction of a time: In minutes

The value is read-in and shown in the report with the appropriate manipulation. The corrected values can be saved using **Save archive**.

> ⚠ **Attention**
>
> *Every time the archive is saved, the values are corrected by the defined constant!*

archivemr

This function makes it possible to automatically change values or times of an archive entry by adding a constant to a value.

Syntax: =archivemr(FILTERINDEX,PARAMETER,DIRECTION)

Transfer parameters	Valid range	Comment
`FILTERINDEX`	`[1-n]`	Corresponding entry from the archive filter (on page 43).
`PARAMETER`	`value, status, time, count`	The selected parameter is read from the archive and displayed
`DIRECTION`	`bottom, right`	Filling in the corresponding direction.
`MATH`		▸ Constant: signed ▸ Decimal points: Point separator (.) ▸ Correction of a time: In minutes

> ⚠ **Attention**
>
> *Changed values cannot be saved in the archive.*
>
> The value is read-in and shown in the report with the appropriate manipulation. Only `time` and `value` are possible as parameters.`=archivemr(4,value,right,6.5)`
>
> `All archive values read` are increased by `6.50` for the display in the report.
>
> `=archivemr(4,time,right,-1)`
>
> *The time stamps of the archive entries are corrected back by `1` minute for display in the report.*
>
> *Possibility for use: For example, for a calculated monthly value that is only available after the end of the month (time stamp, for example. `00:01:00`).*

archivemsp

This function makes it possible to automatically change the value or times of an archive entry by adding a constant to the value as well as establishing the cells for the entries. The value is read-in and shown in the report with the appropriate manipulation. Changes are either saved manually via the `Save` control element or automatically saved when saving a report file. The status of an amended entry is automatically set to `manual value`.

`Syntax: =archivemsp(FILTERINDEX,PARAMETER,DIRECTION)`

Transfer parameters	Valid range	Comment
FILTERINDEX	[1-n]	Corresponding entry from the archive filter (on page 43).
PARAMETER	value, status, time, count	The selected parameter is read from the archive and displayed
DIRECTION	bottom, right	Filling in the corresponding direction.
P1		Number of archive values in the first row.
P2		Empty cells between two value outputs.
P3		Empty cells until the next value row.
MATH		▸ Signed constant, Constant: signed ▸ Decimal points: Point separator (.) ▸ Correction of a time: In minutes

⚠ **Attention**

The values are corrected by the defined constant with every ***Save archive***.

CONTROL CELLS

The parameters P1, P2 and P3 set out which cells of the report can be filled:

▸ P1: Number

▸ P2: Empty cells between two value outputs, depending on DIRECTION (down, right) either empty rows or empty columns

▸ P3: Empty cells between two value outputs, depending on DIRECTION (down, right) either empty rows or empty columns

All entries of the time range of the report are shown in the report, going downwards or tothe right from the cell in which the function was given, with the row format defined by the parameters P1-P3.

WHEN SELECTING DOWN, THE FOLLOWING APPLIES:

▸ P1 = number of rows to be filled with values

- ▶ `P2` = empty rows between the values

- ▶ `P3` = empty columns

WHEN SELECTING `RIGHT`, THE FOLLOWING APPLIES:

- ▶ `P1` = number of columns to be filled

- ▶ `P2` = empty rows between the values

- ▶ `P3` = empty rows between the values

- ▶

> **Example**
>
> `=archivemsp(4,"time","right",16,1,0,-0.5)`
>
> *All time entries of the archive entry with the filter number 4 are retarded 30 seconds; 16 entries are envisaged with an empty column and no space.*

archivemspr

This function makes it possible to automatically change values or times of an archive entry for display in a report by adding a constant to a value. The value is read-in and shown in the report with the appropriate manipulation. Only `time` and `value` are possible as parameters.

> ⚠ **Attention**
>
> *Changed values cannot be saved in the archive.*

`Syntax: =archivemspr(FILTERINDEX,PARAMETER,DIRECTION)`

Transfer parameters	Valid range	Comment
FILTERINDEX	[1-n]	Corresponding entry from the archive filter (on page 43).
PARAMETER	value, status, time, count	The selected parameter is read from the archive and displayed
DIRECTION	bottom, right	Filling in the corresponding direction.
P1		Number of archive values in the first row.
P2		Empty cells between two value outputs.
P3		Empty cells until the next value row.
MATH		▸ Signed constant, Constant: signed ▸ Decimal points: Point separator (.) ▸ Correction of a time: In minutes

CONTROL CELLS

The parameters P1, P2 and P3 set out which cells of the report can be filled:

- ▸ P1: Number

- ▸ P2: Empty cells between two value outputs, depending on DIRECTION (down, right) either empty rows or empty columns

- ▸ P3: Empty cells between two value outputs, depending on DIRECTION (down, right) either empty rows or empty columns

All entries of the time range of the report are shown in the report, going downwards or tothe right from the cell in which the function was given, with the row format defined by the parameters P1-P3.

WHEN SELECTING DOWN, THE FOLLOWING APPLIES：

- ▸ P1 = number of rows to be filled with values

- ▸ P2 = empty rows between the values

- ▸ P3 = empty columns

WHEN SELECTING `RIGHT`, THE FOLLOWING APPLIES:

- ▶ `P1` = number of columns to be filled

- ▶ `P2` = empty rows between the values

- ▶ `P3` = empty rows between the values

Example

```
=archivemsp(4,"time","right",16,1,0,-0.5)
```

All time entries of the archive entry with the filter number 4 are retarded 30 seconds; 16 entries are envisaged with 1 empty column and 0 spaces.

archiver

This function makes it possible to read the value, status and time of an archive entry. No changes are possible.

Syntax: =archiver(FILTERINDEX,PARAMETER,DIRECTION)

Transfer parameters	Valid range	Comment
FILTERINDEX	[1-n]	Corresponding entry from the archive filter (on page 43).
PARAMETER	value, status, time, count	The selected parameter is read from the archive and displayed
DIRECTION	bottom, right	Filling in the corresponding direction.

This differs from the `archive` (on page 56) function in that no manual entries can be made in Runtime.

Area of application: These functions are employed, for example, if an entry in a report must be shown several times and may only be edited at one place. All entries of the time range of the report are shown in the report going downwards or to the right from the cell in which the function was created.

archivesp

Value, Status or time of the archive entry can be read. Changes are either saved manually via the `Save` control element or automatically saved when saving a report file. The status of an amended entry is automatically set to `manual value`.

```
Syntax: =archivesp(FILTERINDEX,PARAMETER,DIRECTION,P1,P2,P3)
```

Transfer parameters	Valid range	Comment
FILTERINDEX	[1-n]	Corresponding entry from the archive filter (on page 43).
PARAMETER	value, status, time, count	The selected parameter is read from the archive and displayed
DIRECTION	bottom, right	Filling in the corresponding direction
P1		Number of archive values in the first row
P2		Empty cells between two value outputs
P3		Empty cells until the next value row

CONTROL CELLS

The parameters P1, P2 and P3 set out which cells of the report can be filled:

▶   P1: Number

▶   P2: Empty cells between two value outputs, depending on DIRECTION (down, right) either empty rows or empty columns

▶   P3: Empty cells between two value outputs, depending on DIRECTION (down, right) either empty rows or empty columns

All entries of the time range of the report are shown in the report, going downwards or tothe right from the cell in which the function was given, with the row format defined by the parameters P1-P3.

WHEN SELECTING DOWN, THE FOLLOWING APPLIES:

▶   P1 = number of rows to be filled with values

▶   P2 = empty rows between the values

▶   P3 = empty columns

WHEN SELECTING RIGHT, THE FOLLOWING APPLIES:

▶   P1 = number of columns to be filled

▶   `P2` = empty rows between the values

▶   `P3` = empty rows between the values

When outputting the status information, the settings from the `project.ini` file in the **[STATUS]** section or the settings from the internally defined allocation are used. If the status labeling was changed in `project.ini`, the changed name is given. (Tip: makes it possible to keep the project compatible with old status labels.)

archivespr

Value, Status or time of the archive entry can be read.

Syntax: `=archivespr(FILTERINDEX,PARAMETER,DIRECTION,P1,P2,P3)`

Transfer parameters	Valid range	Comment
`FILTERINDEX`		Corresponding entry from the archive filter (on page 43).
`PARAMETER`	`value, status, time, count`	The selected parameter is read from the archive and displayed
`DIRECTION`	`bottom, right`	Filling in the corresponding direction
`P1`		Number of archive values in the first row
`P2`		Empty cells between two value outputs
`P3`		Empty cells until the next value row

CONTROL CELLS

The parameters   `P1`, `P2` and `P3` set out which cells of the report can be filled:

▶   `P1`: Number

▶   `P2`: Empty cells between two value outputs, depending on `DIRECTION` (`down`, `right`) either empty rows or empty columns

▶   `P3`: Empty cells between two value outputs, depending on `DIRECTION` (`down`, `right`) either empty rows or empty columns

All entries of the time range of the report are shown in the report, going downwards or tothe right from the cell in which the function was given, with the row format defined by the parameters P1-P3.

WHEN SELECTING `DOWN`, THE FOLLOWING APPLIES:

▸ `P1` = number of rows to be filled with values

▸ `P2` = empty rows between the values

▸ `P3` = empty columns

WHEN SELECTING `RIGHT`, THE FOLLOWING APPLIES:

▸ `P1` = number of columns to be filled

▸ `P2` = empty rows between the values

▸ `P3` = empty rows between the values

Unlike the `archivesp` (on page 66) function, no manual presettings can be set on the cells. These functions are employed, for example, if an entry in a log must be shown several times and may only be edited at one place. All entries of the time range of thereport are shown in the report going downwards or to the right from the cell in which the function was created

> ### Example
>
> *=archivespr(1,"value","downwards",12,1,3)*
>
> *All values of the archive entry with the filter number `1` are written `downwards`; `12`*
> *entries are envisaged with `1` empty row and `3` empty columns.*

archivespw

This function suppresses the displays of read-in archive entries and makes it possible to define the cells to be used. When saving, the archive entries are overwritten with values and statuses existing at other places (e.g. calculated).

```
Syntax:
=archivespw(FILTERINDEX,PARAMETER,DIRECTION,P1,P2,P3,SZ_WERT,SZ_STATUS)
```

Transfer parameters	Valid range	Comment
FILTERINDEX	[1-n]	Corresponding entry from the archive filter (on page 43).
PARAMETER	value, status, time, count	The selected parameter is read from the archive and displayed
DIRECTION	bottom, right	Filling in the corresponding direction.
P1		Number of archive values in the first row.
P2		Empty cells between two value outputs.
P3		Empty cells until the next value row.
SZ_VALUE	Start-cell values	Cell from which calculated values start to appear.
SZ_STATUS	Start cell status	Defining a fixed status text (for example MAN_VAL), or stating the starting cell from which the status text should be taken.

CONTROL CELLS

The parameters P1, P2 and P3 set out which cells of the report can be filled:

▶ P1: Number

▶ P2: Empty cells between two value outputs, depending on DIRECTION (down, right) either empty rows or empty columns

▶ P3: Empty cells between two value outputs, depending on DIRECTION (down, right) either empty rows or empty columns

All entries of the time range of the report are shown in the report, going downwards or to the right from the cell in which the function was given, with the row format defined by the parameters P1-P3.

WHEN SELECTING DOWN, THE FOLLOWING APPLIES:

▶ P1 = number of rows to be filled with values

▶ P2 = empty rows between the values

▶ P3 = empty columns

WHEN SELECTING `RIGHT`, THE FOLLOWING APPLIES:

▶   `P1` = number of columns to be filled

▶   `P2` = empty rows between the values

▶   `P3` = empty rows between the values

STATUS TEXTS

The configuring is not done via the input help but directly in the cell. For this, the following applies:

▶   When entering the status texts, the defined short texts from the `project.ini` file or from the system-internal default configuration must be used. This applies regardless of whether the statues are input as fixed or read in from another cell.

▶   Several statuses are separated by a comma (`,`).

▶   The calculated values are updated regardless of whether the newly calculated value has changed in comparison to the read-in value.

▶   The status is changed to manual value (`MAN_VAL`).

archivew

This function suppresses the display of the archive entries that have been read in for the report. When saving, the archive entries are overwritten with values and statuses existing at other places (e.g. calculated). This report function can be used as an example for reduction of monthly values to annual values as an archive value.

```
Syntax:
=archivew(FILTERINDEX,PARAMETER,DIRECTION,P1,P2,P3,SZ_WERT,SZ_STATUS)
```

Transfer parameters	Valid range	Comment
FILTERINDEX	[1-n]	Corresponding entry from the archive filter (on page 43).
PARAMETER	value, status, time, count	The selected parameter is read from the archive and displayed
DIRECTION	bottom, right	Filling in the corresponding direction.
SZ_VALUE	Start-cell values	Cell from which calculated values start to appear.
SZ_STATUS	Start cell status	Defining a fixed status text (for example MAN_VAL), or stating the starting cell from which the status text should be taken.

CONTROL CELLS

The parameters P1, P2 and P3 set out which cells of the report can be filled:

▶ P1: Number

▶ P2: Empty cells between two value outputs, depending on DIRECTION (down, right) either empty rows or empty columns

▶ P3: Empty cells between two value outputs, depending on DIRECTION (down, right) either empty rows or empty columns

All entries of the time range of the report are shown in the report, going downwards or tothe right from the cell in which the function was given, with the row format defined by the parameters P1-P3.

WHEN SELECTING DOWN, THE FOLLOWING APPLIES:

▶ P1 = number of rows to be filled with values

▶ P2 = empty rows between the values

▶ P3 = empty columns

WHEN SELECTING RIGHT, THE FOLLOWING APPLIES:

▶ P1 = number of columns to be filled

▶ P2 = empty rows between the values

▶   `P3` = empty rows between the values

STATUS TEXTS

The configuring is not done via the input help but directly in the cell. For this, the following applies:

▶   When entering the status texts, the defined short texts from the `project.ini` file or from the system-internal default configuration must be used. This applies regardless of whether the statues are input as fixed or read in from another cell.

▶   Several statuses are separated by a comma (`,`).

▶   The calculated values are updated regardless of whether the newly calculated value has changed in comparison to the read-in value.

▶   The status is changed to manual value (`MAN_VAL`).

Example

Reductions from monthly values to annual values as an archive value:

In a monthly report with daily values, the amount of days on which a limit x was exceeded is calculated. The sum of the limit violations is written into a cell of the report.

▸   In an archive, a value was defined that gets one entry per month.

▸   This entry is read in via `archivew` in the report.

▸   `=archivew(4,"value","downwards",B12,C12)`

- `4`: Variable from archive filter
- `value`: Parameter of variables that are to be displayed
- `bottom`: Direction in which values are written
- `B12`: Start cell, corresponds to the calculated cell
- `C12`: Defines status text to be given

▸   The sum is written to the archive value when the archive values are saved.

---

archivetime

The `archivetime` function supplies, depending on the time filter that has been set, the time information for the start and end of the archive interrogation in online operation.

Syntax: `=archivetime(FILTERINDEX,PARAMETER)`

Transfer parameters	Valid range	Comment
FILTERINDEX	[1-n]	Corresponding entry from the archive filter (on page 43).
PARAMETER	absolutetimefrom, absolutetimeto	Start time or end time of the archive request.

batch

This function outputs the first batch names for the archive selected in the filter area of the report.

Syntax: `=batch(FILTERINDEX)`

Transfer parameters	Valid range	Comment
FILTERINDEX	[1-n]	Corresponding entry from the archive filter (on page 43).

Example

=batch(1)

First batch name for the archive of the variables placed at position 1 of the archive filter.

batch no.

This function displays different information about batches. The output is displayed as a list which is evaluated from top to bottom or from left to right. The following are displayed:

- ▶ Name of the batches

- ▶ Start time

- ▶ End point

Syntax: `=batchnr(FILTERINDEX,PARAMETER,DIRECTION)`

Transfer parameters	Valid range	Comment
FILTERINDEX	[1-n]	Corresponding entry from the archive filter (on page 43).
PARAMETER	name, time from, time to	The selected parameter is read from the archive and displayed
DIRECTION	bottom, left	Filling in the corresponding direction.

wmz

The =wmz wertet function evaluates counter values saved in the archive. Counter overflow is not automatically checked.

Syntax:
=wmz(FILTERINDEX,ACTIVATED,OVERFLOW,STARTVALUE,ENDVALUE,COUNTERINFO,ERRORCOUNTER)

Transfer parameters	Valid range	Comment
FILTERINDEX	[1-n]	Corresponding entry from the archive filter (on page 43).
ACTIVATED	0 = deactivated >1 = activated	The function is not executed with 0.
OVERFLOW	Value or cell reference	With an overflow of 1000, an error is recognized with an archive value of 1000 or higher.
STARTVALUE	Cell reference	First valid value from the archive.
ENDVALUE	Cell reference	Last valid value from the archive.
COUNTERINFO	Cell reference	Start of the counter change. Right direction from the cell reference. Consists of four cell entries: ▶ Date and time of the first value after a counter change ▶ Value before the counter change ▶ Value after the counter change ▶ Empty cell
ERRORCOUNTER	Cell reference	States the number of errors found in a cell. Causes of the error: ▶ Value >= as defined in OVERFLOW ▶ Values <= 0 ▶ With two consecutive archive values, the first value is greater than the subsequent value.

The sum of all valid values is written in the cell of the function.

wmzz

The wmzz function (filter index, direction, P1, P2, P3, activated, overflow) lists the difference between two consecutive archive values.
The output is '0' if

▶ The first value is >= the subsequent value.

▶ One of the two values is >= OVERFLOW.

▶ One of the two values is = 0.

Syntax: `=wmzz(filter index, direction, P1, P2, P3, activated, overflow)`

Transfer parameters	Valid range	Comment
`FILTERINDEX`	[1-n]	Corresponding entry from the archive filter (on page 43).
`DIRECTION`	`bottom, left`	Filling in the corresponding direction.
`P1`		Number of archive values in the first row
`P2`		Empty cells between two value outputs
`P3`		Empty cells until the next value row
`AKTIVATED`	0 = deactivated 1 = activated	The function is not executed with `0`.
`OVERFLOW`	Value or cell reference	With an overflow of `1000`, an error is recognized with an archive value of `1000` or higher.

CONTROL CELLS

The parameters `P1`, `P2` and `P3` set out which cells of the report can be filled:

▶ `P1`: Number

▶ `P2`: Empty cells between two value outputs, depending on `DIRECTION` (`down`, `right`) either empty rows or empty columns

▶ `P3`: Empty cells between two value outputs, depending on `DIRECTION` (`down`, `right`) either empty rows or empty columns

All entries of the time range of the report are shown in the report, going downwards or to the right from the cell in which the function was given, with the row format defined by the parameters P1-P3.

WHEN SELECTING ᴅᴏᴡɴ, THE FOLLOWING APPLIES:

▶ `P1` = number of rows to be filled with values

▶ `P2` = empty rows between the values

▶ `P3` = empty columns

WHEN SELECTING RIGHT, THE FOLLOWING APPLIES:

- ▶ `P1` = number of columns to be filled

- ▶ `P2` = empty rows between the values

- ▶ `P3` = empty rows between the values

Database functions

This function enables data from Access or SQL databases to be read off.

Syntax: `=sqldao (DATABASE,SQL-QUERY,PARA1,PARA2)`
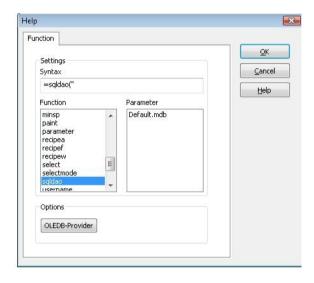
> ⚠ **Attention**
>
> *SQLDAO does not work with:*
>
> ▸ `ntext`
>
> ▸ `sql_variant`
>
> ▸ `nvarchar(MAX)`
>
> ▸ `nchar(MAX)`

Transfer parameters	Comment
DATABASE	Name of the *.mdb database or the OLEDB provider.
SQL query	Standard 'SELECT', 'INSERT', 'UPDATE', 'DELETE' and 'EXECUTE' database queries can be used here. Two-integer parameters can be transferred with %d %d, and two-string parameters can be transferred with %s %s
PARA1	parameters for SQL query
PARA2	parameters for SQL query

CONFIGURATION WITH FUNCTION ASSISTANT
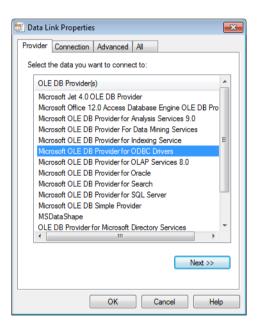
You can use MS Access databases and databases with OLEDB provider. For this, use the following syntax:

Database	Syntax
MS Access database	`=sqldao` (database,SQL query,Para1,Para2)
Databases with OLEDB provider	Replace the reference to a database with an OLEDB initialization string. This string can be ▸ entered manually or ▸ created by clicking on the **OLEDB provider** button.

DATABASES WITH OLEDB PROVIDER

Clicking on the **OLEDB provider** `button` opens the dialog for **data linking properties:**

Here, you define the connection to the desired database:

▸ OLEDB provider type

▸ Data source

▸ Registration information

▸ etc.

After confirmation of the dialog by clicking on **OK**, the initialization string is generated and inserted into the `Syntax` field.

Syntax: `=sqldao (Provider,SQL query,Para1,Para2)`

> ⚠️ **Attention**
>
> *The initialization string must always start with the character sequence* `Provider=`.
> *Note upper case and lower case letters, because the string is* `case sensitive`.

SETTINGS FOR MS ACCESS AND OLEDB

Transfer parameters	Description
Database	Name of the *.mdb database. Storage directory: Project folder which is superordinate to the Runtime folder. For example: Runtime storage location: Subfolder of `C:\Users\Public\Documents\zenon_Projects\Multi\TEST_PROJEKT` Save location *.mdb: saved directly in the folder`C:\Users\Public\Documents\zenon_Projects\Multi\TEST_PROJEKT` The project folder is defined in the project properties.
Provider	OLEDB provider; the initialization string must always begin with `Provider=` (case sensitive).
SQL query	You can use standard 'select' database queries here. Two-integer parameters can be transferred with `%d %d` and two-string parameters can be transferred with `%s %s`
Para1, Para2	Parameters for SQL query.

Example of access

EXAMPLES FOR MS ACCESS

Example database: `Address.mdb`

Name	First name	Street	City	Telephone	CanAddress
Müllner	Egon	Am Rheinberg 22	5020 Salzburg	0662/329354	1
Roider	Jürgen	Mozartgasse 7	5020 Salzburg	0662/329354	2
Leitner	Hans	Strubergasse 6	5020 Salzburg	0662/329354	3
Müller	Peter	Blumengasse 6	80003 München	089/234243	5
Weinberger	ww	Hafnergasse 5	7020 Klagenfurt	0222/2222	6

EXAMPLE 1

All entries from the `Addresses` table with `Name` field names are to be given.

Function: `=sqldao(Adress.mdb,SELECT Adressen.name FROM Adressen ;,1,1)`

`Address.mdb`	Access database (must be in the project database path)
`SELECT`	Tells the Microsoft Jet database module to return information from the database as a group of records
`Address.name`	table.field name
`FROM`	states the table of query in which to find the fields stated in the SELECT statement
`Addresses`	Table

Result:

Müllner
Roider
Leitner
Müller
Weinberger

EXAMPLE 2

▸ Lower limit: Contents of cell `E1`

▸ Upper limit: Contents of cell `E2`

▸ Field name for evaluation: `CanAddress`

- ▶ Entries from the `Addresses` table with `Name` and `First name` field names are to be given.

- ▶ The `CanAddress` condition must be higher than contents of cell `E1` and lower than contents of cell `E2`.

Function: `=sqldao(Adress.mdb,SELECT Address.name,Address.first name FROM Address WHERE (((Adressen.CanAdresse)>%d) AND ((Address.CanAdresse)<%d)) ;,E1,E2)`

Address.mdb	Access database (must be in the project database path)
SELECT	Tells the Microsoft Jet database module to return information from the database as a group of records
Address.name	table.field name
,	SQL separator
Address.first name	table.field name
FROM	states the table of query in which to find the fields stated in the SELECT statement
Addresses	Table
WHERE	States which records of the table defined in the FROM section are affected by a SELECT, UPDATE or DELETE statement
(((Address.CanAddress)>%d)	Condition: table.field name
AND	Linkage
((Address.CanAddress)<%d))	Condition: table.field name
;	SQL statement end
E1, E2:	Cell references (integer)

Result for E1 = 1, E2 = 6

Roider	Jürgen	2
Leitner	Hans	3
Müller	Peter	5

> 💡 **Info**
>
> *For further information on SQL statements refer to the Online-Help of Access.*

SQL example

EXAMPLE FOR DATABASES WITH OLEDB PROVIDER

=sqldao("Provider=SQLNCLI.1;Integrated Security=SSPI;Persist Security Info=False;Initial Catalog=<databasename>;Data Source=<database instance>;Use Procedure for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=<PCNAME>;Use Encryption for Data=False;Tag with column collation when possible=False","SELECT <tablename>.<columnname> FROM <tablename>;",0,0)

Transfer parameters	Comment
DATABASE	"Provider=SQLNCLI.1;Integrated Security=SSPI;Persist Security Info=False;Initial Catalog=<databasename>;Data Source=<database instance>;Use Procedure for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=<PCNAME>;Use Encryption for Data=False;Tag with column collation when possible=False"
SQL query	"SELECT <tablename>.<columnname> FROM <tablename>;"
PARA1	0
PARA2	0

VALUES FOR EXAMPLE

Database: "AdventureWorks"

Installation: AdventureWorksDB.msi

Source: http://msftdbprodsamples.codeplex.com/releases/view/4004

Computer name: "PC1"

Authentication: "Windows NT"

SQL server instance: "SQLEXPRESS"

SELECT

SELECT

=sqldao("Provider=SQLOLEDB.1;Integrated Security=SSPI;Persist Security Info=False;Initial Catalog=AdventureWorks;Data Source=.\SQLEXPRESS;Use Procedure for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=PC1;Use Encryption for Data=False;Tag with column collation when possible=False","SELECT TOP 100 [AddressID],[AddressLine1],[AddressLine2],[City],[StateProvinceID],[PostalCode] FROM [AdventureWorks].[Person].[Address] ORDER BY AddressID;",0,0)

SELECT WITH STRING PARAMETER (CELL H2)

=sqldao("Provider=SQLOLEDB.1;Integrated Security=SSPI;Persist Security Info=False;Initial Catalog=AdventureWorks;Data Source=.\SQLEXPRESS;Use Procedure for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=PC1;Use Encryption for Data=False;Tag with column collation when possible=False","SELECT TOP 100 [AddressID],[AddressLine1],[AddressLine2],[City],[StateProvinceID],[PostalCode] FROM [AdventureWorks].[Person].[Address] WHERE (City = N'%s') ORDER BY AddressID;",H2,0)

SELECT WITH NUMERICAL PARAMETER (CELL H2)

=sqldao("Provider=SQLOLEDB.1;Integrated Security=SSPI;Persist Security Info=False;Initial Catalog=AdventureWorks;Data Source=.\SQLEXPRESS;Use Procedure for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=PC1;Use Encryption for Data=False;Tag with column collation when possible=False","SELECT TOP 100 [AddressID],[AddressLine1],[AddressLine2],[City],[StateProvinceID],[PostalCode] FROM [AdventureWorks].[Person].[Address] WHERE (StateProvinceID = %d) ORDER BY AddressID;",H2,0)

INSERT

=sqldao("Provider=SQLOLEDB.1;Integrated Security=SSPI;Persist Security Info=False;Initial Catalog=AdventureWorks;Data Source=.\SQLEXPRESS;Use Procedure for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=PC1;Use Encryption for Data=False;Tag with column collation when possible=False","INSERT INTO [AdventureWorks].[Sales].[Currency] (CurrencyCode,Name,ModifiedDate) VALUES ('AAA','NewValue','');",0,0)

UPDATE

=sqldao("Provider=SQLOLEDB.1;Integrated Security=SSPI;Persist Security Info=False;Initial Catalog=AdventureWorks;Data Source=.\SQLEXPRESS;Use Procedure for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=PC1;Use Encryption for Data=False;Tag with column collation when possible=False","**UPDATE** Sales.Currency **SET** Name = 'ModifiedValue' **WHERE** CurrencyCode = 'AAA';",0,0)

DELETE

=sqldao("Provider=SQLOLEDB.1;Integrated Security=SSPI;Persist Security Info=False;Initial Catalog=AdventureWorks;Data Source=.\SQLEXPRESS;Use Procedure for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=PC1;Use Encryption for Data=False;Tag with column collation when possible=False","**DELETE** Sales.Currency **WHERE** CurrencyCode = 'AAA';",0,0)

EXECUTE

EXECUTE with numerical parameter (cell ʃ2)

=sqldao("Provider=SQLOLEDB.1;Integrated Security=SSPI;Persist Security Info=False;Initial Catalog=AdventureWorks;Data Source=.\SQLEXPRESS;Use Procedure for Prepare=1;Auto Translate=True;Packet Size=4096;Workstation ID=PC1;Use Encryption for Data=False;Tag with column collation when possible=False","**EXECUTE** dbo.uspGetBillOfMaterials %d, '2001-05-08 12:35:29.123';",ʃ2,0)

Date and time functions

This function gives date and time values.

You can find the formatting keys in the Data types for date and time (on page 156) chapter.

date

The date(iYear, iMonth, iDay) function date calculates the date from the numerical values for year (iYear), month (iMonth) and day (iDay).

Syntax: =date(iYear,iMonth,iDay)

Transfer parameters	Valid range	Comment
iYear	[0100-2200]	
iMonth	[1-12]	Values out of the valid range are set to the upper or lower limit
iDay	[1-(28,29,30,31)]	Values out of the valid range are set to the upper or lower limit

Result format: date

Standard format: dd/mm/yyyy

Example

=date(2010,5,12) is **12.05.2010**

Cell A1=2010, cell A2=5, cell A3=12, cell A4=date(A1,A2,A3) is 12.05.2010

datevalue

The datevalue(date_text) function calculates the date from the date text (date_text).

Syntax: =datevalue(date_text)

day

The day(date) function calculates the day from a date value.

Syntax: =day(date)

Result format: Integer

Example

=now() with format string hh:mm:ss dd/mm/yy is 12:46:24 05.06.10
Cell A2=day(A1) is 5

hour

The hour(time) function calculates the hour from a time value.

Syntax: =hour(time)

Result format: Integer

> **Example**
>
> *=now() with format string hh:mm:ss dd/mm/yy is 12:46:24 05.06.10*
>
> *Cell A2=hour(A1) is 12*

minute

The `minute(time)` function calculates the minutes from a time value.

Syntax: `=minute(time)`

Result format: Integer

> **Example**
>
> *=now() with format string hh:mm:ss dd/mm/yy is 12:46:24 05.06.10*
>
> *cell A2=minute(A1) is 46*

month

The `month(date)` function calculates the month from the date value.

Syntax: `=month(date)`

Result format: Integer

> **Example**
>
> *=now() with format string hh:mm:ss dd/mm/yy is 12:46:24 05.06.10*
>
> *Cell A2=month(A1) is 6*

now

The `now()` function displays the current date and time.

The cell automatically gets the format string dd/mm/yy.

Syntax: `=now()`

> 📋 **Example**
>
> *=now() is 05.06.2010*
>
> *=now() with format string hh:mm:ss dd/mm/yy is 12:46:24 05.06.10*

second

The `second(time)` function calculates the seconds from a time value.

Syntax: `=second(time)`

Result format: Integer

> 📋 **Example**
>
> *=now() with format string hh:mm:ss dd/mm/yy is 12:46:24 05.06.10*
>
> *Cell A2=second(A1) is 24*

time

The `time(iHour, iMinute, iSecond)` function provides the time from the numerical values for hour (`iHour`), minute (`iMinute`) and seconds (`iSecond`).

The cell automatically gets the format string hh:mm:ss.

Syntax: `=time(iHour,iMinute,iSecond)`

Transfer parameters	Valid range	Comment
`iHour`	[0-23]	Values out of the valid range are set to the upper or lower limit
`iMinute`	[0-59]	Values out of the valid range are set to the upper or lower limit
`iSecond`	[0-59]	Values out of the valid range are set to the upper or lower limit

Result format: `time`

> ### 🔲 Example
>
> =time(12,24,48) is **12:24:48**
>
> *Cell A1=12, cell A2=24, cell A3=48, cell A4=time(A1,A2,A3) is 12:24:48*

today

The `today()` function displays the current date.

The cell automatically gets the format string dd/mm/yy.

Syntax: `=today()`

> ### 🔲 Example
>
> *=today() is 05.06.2010*
>
> *=today() with format string hh:mm:ss dd/mm/yy is 12:00:00 AM 05.06.10*

weekday

The `weekday(date)` function calculates the weekday from the date value.

Syntax: `=weekday(date)`

value	Weekday
1	Sunday
2	Monday
3	Tuesday
4	Wednesday
5	Thursday
6	Friday
7	Saturday

Result format: Integer

> **Example**
>
> *=now () with format string hh:mm:ss dd/mm/yy is 12:46:24 05.06.10*
>
> *Cell A2=weekday(A1) is 3*

year

The `year(date)` function calculates the year from the date value.

Syntax: `=year(date)`

Result format: Integer

> **Example**
>
> *=now () with format string hh:mm:ss dd/mm/yy is 12:46:24 05.06.10*
>
> *Cell A2=year(A1) is 2010*

Logical functions

This section summarizes all logical functions of the report generator

and

The `and(logical,logical)` function provides the `TRUE` value if both arguments are `TRUE`.

The cell automatically gets the format string `True;False`.

Syntax: `=and(logical,logical)`

EXAMPLE

Cell A1	Cell A2	Cell A3=and(A1,A2)
false	false	false
false	true	false
true	false	false
true	true	true

exact

The `exact(string1,string2)` function returns the value `TRUE`, if both text strings are identical (case-sensitive).

Syntax: `=exact(string1,string2)`

false

The `false()` function returns the value `FALSE`.

The cell automatically gets the format string `True;False`.

Syntax: `=false()`

> **Example**
>
> *The entry in cell A1=`false()` is `False`*

if

The `if(exprTest,valTrue,valFalse)` function provides the content of `valTrue` (value or string), if the argument is `exprTest` `TRUE`, otherwise the content of `valFalse`. It is possible to nest `if` functions, for example by implementing `and`-links.

Syntax: `=if(exprTest,valTrue,valFalse)`

isblank

The `isblank(ref)` function has the logical value `TRUE`, if the reference cell is empty.

Syntax: `=isblank(ref)`

EXAMPLE

ContentA1	A1	Result
Logical expression	No	False
Text	This is a text	False
empty		True
Number	123.00	False
Error	#ARG!	#ARG!

iserror

The `iserror(ref)` function has the logical value TRUE if the reference cell contains an error.

Syntax: `=iserror(ref)`

EXAMPLE

ContentA1	A1	Result
Logical expression	No	False
Text	This is a text	False
empty		False
Number	123.00	False
Error	#ARG!	True

islogical

The `islogical(ref)` function has the logical value TRUE if the reference cell contains a logical value.

Syntax: `=islogical(ref)`

Example

ContentA1	A1	Result
Logical expression	No	True
Text	This is a text	False
empty		False
Number	123.00	False
Error	#ARG!	#ARG!

isnontext

The `isnontext(ref)` function has the logical value `TRUE` if the reference cell does not contain a text string.

Syntax: `=isnontext(ref)`

EXAMPLE

ContentA1	A1	Result
Logical expression	No	True
Text	This is a text	False
empty		True
Number	123.00	True
Error	#ARG!	#ARG!

isnumber

The `isnumber(ref)` function has the logical value `TRUE` if the reference cell contains a numerical value.

Syntax: `=isnumber(ref)`

EXAMPLE

ContentA1	A1	Result
Logical link	No	False
Text	This is a text	False
empty		False
Number	123.00	True
Error	#ARG!	#ARG!

isref

The `isref(ref)` function has the logical value TRUE if the reference cell contains a reference to another cell.

`Syntax: =isref(ref)`

EXAMPLE

ContentA1	A1	Result
Logical expression	No	False
Text	This is a text	False
empty		False
Number	123.00	False
Error	#ARG!	False
Reference	=B1	True

istext

The `istext(ref)` function has the logical value TRUE if the reference cell contains a text string.

`Syntax: =istext(ref)`

EXAMPLE

ContentA1	A1	Result
`Logical link`	No	`False`
`Text`	This is a text	`True`
`empty`		`False`
`Number`	123.00	`False`
`Error`	`#ARG!`	`#ARG!`

not

The `not(logical)` function returns the logical opposite of the argument.

The cell automatically gets the format string `True;False`.

Syntax: `=not(logical)`

EXAMPLE

Cell A1	Cell A2=`not(A1)`
`false`	`True`
`true`	`False`

or

The `or(logical,logical)` function provides the `TRUE` logical value if both arguments are `TRUE`.

The cell automatically gets the format string `True;False`.

Syntax: `=or(logical,logical)`

EXAMPLE

Cell A1	Cell A2	Cell A3=or(A1,A2)
false	false	false
false	true	true
true	false	true
true	true	true

true

The `true()` function returns the logical value `TRUE`.

The cell automatically gets the format string `True;False`.

Syntax: `=true()`

> **Example**
>
> *The entry in cell A1=`true()` is `True`*

Mathematical and trigonometric functions

Common mathematical functions are available for reports. These can not only be planned using the function assistant, but can also be directly entered into the input row.

abs

The `abs (num)` function returns the absolute value of a number as a result. The absolute value of a number is the value without sign.

`Syntax: =abs(num)`

`num:` is the real number, that is the basis for the absolute value

> ### ➡ Example
>
> =abs(2.2) is 2
>
> *=abs(-2.2) is 2*

acos

The `acos (num)` function provides the arc cosine of a number.

Syntax: =acos(num)

acosh

The `acosh (num)` function provides the inverse hyperbolic cosine of a number.

Syntax: =acosh(num)

asin

The `asin (num)` function provides the arc cosine of a number.

Syntax: =asin(num)

asinh

The `asinh (num)`) function provides the inverted hyperbolic sine of a number.

Syntax: =asinh(num)

atan

The `atan (num)` function provides the arc tangent of a number.

Syntax: =atan(num)

atanh

The `atanh (num)` function provides the inverted hyperbolic tangent of a number.

Syntax: =atanh(num)

atan2

The `atan2 (x,y)` function provides the arc tangent based on an x and y coordinate.

Syntax: =atan2(x, y)

ceil

The `ceil (num)` function returns the next higher whole-numbered value of the value, rounded up.

Syntax: =ceil(num)

> 📋 **Example**
>
> =ceil(2.2) is 3
>
> *=ceil(-2.2) is -3*

column

The `column (ref)` function provides the column number of a connection (for example C3=2) in the value range of 0 to n.

Syntax: =column(ref)

columns

The `columns (range)` function returns the numbers of columns in the stated range.

Syntax: =columns(range)

cos

The `cos(num)` function provides the cosine of a number.

Syntax: =cos(num)

EXAMPLE

=cos(Input)

	Enter	Result
degrees	radians()	cosine
0.00	0	1.000
45.00	0.78539816339744828	0.707
90.00	1.5707963267948966	0.000
135.00	2.3561944901923448	-0.707
180.00	3.1415926535897931	-1.000
225.00	3.9269908169872414	-0.707
270.00	4.7123889803846897	-0.000
315.00	5.497787143782138	0.707
360.00	6.2831853071795862	1.000

cosh

The cosh(num) function provides the hyperbolic cosine of a number.

Syntax: =cosh(num)

degree

The degree(num) function converts radians to degrees.

Syntax: =degree(num)

EXAMPLE

=degree(Input)

Enter	Result
Radians()	Degree()
0	0
0.785398163397448 28	45
1.570796326794896 6	90
2.356194490192344 8	135
3.141592653589793 1	180
3.926990816987241 4	225
4.712388980384689 7	270
5.497787143782138	315
6.283185307179586 2	360

even

The `even(num)` function returns the next higher even whole-numbered value. Negative values are rounded from 0 away.
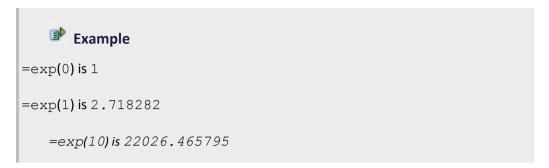
`Syntax: =even(num)`

> 📩 **Example**
>
> =even(2.2) is 4
>
> *=even(-2.2) is -4*

exp

The `exp(num)` function potentializes the basis e with the number stated as argument.

Syntax: =exp(num)

> 🔹 **Example**
>
> =exp(0) is 1
>
> =exp(1) is 2.718282
>
> =exp(10) is 22026.465795

fact

The fact(int) function provides the factorial of a number.

Syntax: =fact(int)

> 🔹 **Example**
>
> =fact(0) is 0 (0)
>
> =fact(1) is 1 (1)
>
> =fact(2) is 2 (1*2)
>
> =fact(3) is 6 (1*2*3)
>
> =fact(4) is 24 (1*2*3*4)
>
> =fact(5) is 120 (1*2*3*4*5)

factdouble

The factdouble(num) function provides the double factorial of a number. In doing so, only the product of each second number is formed.

Syntax: =factdouble(num)

> **Example**
>
> =factdouble(0) is 0 (0)
>
> =factdouble(1) is 1 (1)
>
> =factdouble(2) is 2 (2)
>
> =factdouble(3) is 3 (1*3)
>
> =factdouble(4) is 8 (2*4)
>
> *=factdouble(5) is 15 (1*3*5)*
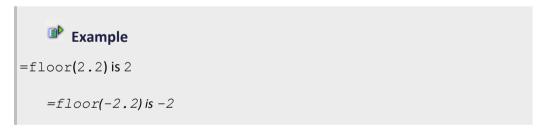
floor

The `floor(num)` function returns the next lower whole-numbered value of the value, rounded down. Negative values are rounded in direction to 0.
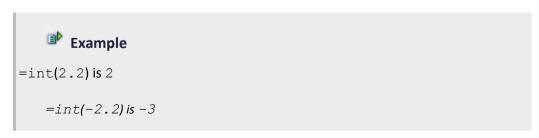
Syntax: =floor(num)

> **Example**
>
> =floor(2.2) is 2
>
> *=floor(-2.2) is -2*

int

The `int(num)` function returns the next lower whole-numbered value of the value, rounded down. Negative values are rounded from 0 away.

Syntax: =int(num)

> **Example**
>
> =int(2.2) is 2
>
> *=int(-2.2) is -3*

ln

The `ln(num)` function provides the natural logarithm of a number.

Syntax: =ln(num)

> **Example**
>
> =ln(1) is 0
>
> =ln(2.718282) is 1
>
> *=ln(22026.465795) is 10*
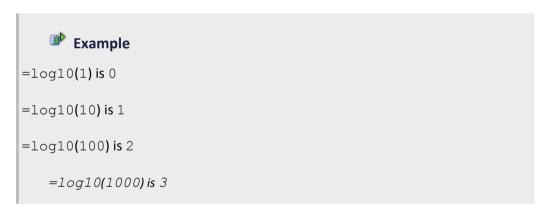
log10

The `log10(num)` function calculates the logarithm of a value on the basis of 10.

Syntax: =log10(num)

> **Example**
>
> =log10(1) is 0
>
> =log10(10) is 1
>
> =log10(100) is 2
>
> *=log10(1000) is 3*

mod

The `mod(num, divisor)` function returns the rest of the division of a value (`num`) by the divisor (`divisor`).

Syntax: =mod(num,divisor)

> **Example**
>
> =mod(0.3) is 0
>
> =mod(1.3) is 1
>
> =mod(2.3) is 2
>
> =mod(3.3) is 0
>
> *=mod(4.3) is 1*

odd

The function odd(num) returns the next higher odd whole-numbered value. Negative values are rounded from 0 away.

Syntax: =odd(num)

> **Example**
>
> =odd(2.2) is 3
>
> *=odd(-2,2) is -3*
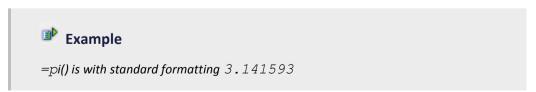
pi

The pi() function pi returns the value for pi: 3.1415926535897931

Syntax: =pi()

Standard format: *0.######

> **Example**
>
> *=pi() is with standard formatting 3.141593*

product

The product (range) function calculates the product of the values in the stated area.

```
Syntax: =product(range)
```

EXAMPLE

Cells A1 to A10: any numbers

Cell `B1=product(A1:A10)`

Cell/column	A	B
1	1	362880 0
2	2	
3	3	
4	4	
5	5	
6	6	
7	7	
8	8	
9	9	
10	10	

The result for the range A1 to A10 is `3628800`.

quotient

The `quotient(num,divisor)` function returns the whole-numbered part of a division.

```
Syntax: =quotient(num,divisor)
```

> ### Example
>
> =quotient(0.3) is 0
>
> =quotient(1.3) is 0
>
> =quotient(2.3) is 0
>
> =quotient(3.3) is 1
>
> *=quotient(4.3) is 1*

radians

The radians(num) function converts degrees into radians.

Syntax: =radians(num)

num: input in degrees

EXAMPLE

=radians(Input)

Enter	Result
0.00	0
45.00	0.78539816339744828
90.00	1.5707963267948966
135.00	2.3561944901923448
180.00	3.1415926535897931
225.00	3.9269908169872414
270.00	4.7123889803846897
315.00	5.497787143782138
360.00	6.2831853071795862

rand

The `rand()` function returns a new random value between `0.000` and `1,000` every time the report is calculated.

`Syntax: =rand()`

randbetween

The `randbetween(min,max)` function returns a new random value between `min` and `max` every time the report is calculated.

`Syntax: =randbetween(min,max)`

round

The `round(num,digits)` function rounds a value (`num`) to a stated number of decimals (`digits`).

`Syntax: =round(num,digits)`

row

The `row(ref)` function returns the row number of a reference (e.g. C3=3) in the value range from `0` to n.

`Syntax: =row(ref)`

rows

The `rows(range)` function calculates the number of empty cells in the defined range.

`Syntax: =rows(range)`

sign

The `sign(num)` function provides the value

▸   `1`, if the number (`num`) is positive

▸   `−1` if the number is negative

▸   `0` if the number is `0`

`Syntax: =sign(num)`

> ⊡ **Example**
>
> =sign(2.2) is 1
>
> =sign(-2.2) is -1
>
> *=sign(0.0) is 0*

sin

The sin(num) function provides the sine of a number.

Syntax: =sin(num)

num: input in radians

EXAMPLE

=sin(Input)

	Enter	Result
degrees	radians()	sine
0.00	0	0.000
45.00	0.78539816339744828	0.707
90.00	1.5707963267948966	1.000
135.00	2.3561944901923448	0.707
180.00	3.1415926535897931	0.000
225.00	3.9269908169872414	-0.707
270.00	4.7123889803846897	-1.000
315.00	5.497787143782138	-0.707
360.00	6.2831853071795862	-0.000

sinh

The `sinh(num)` function provides the hyperbolic sine of a number.

`Syntax: =sinh(num)`

sqrt

The `sqrt(num)` function provides the square root of a number.

`Syntax: =sqrt(num)`

> **Example**
>
> =sqrt(1) is 1
>
> =sqrt(2) is 1.414214
>
> =sqrt(3) is 1.732051
>
> *=sqrt(4) is 2*

sqrtpi

The sqrtpi(num) function provides the square root of a pi value.

Syntax: =sqrtpi()

> **Example**
>
> =sqrtpi(1) is 3.141593 = 1*pi
>
> =sqrtpi(2) is 4.442883 = sqrt(2)*pi
>
> =sqrtpi(3) is 5.441398 = sqrt(3)*pi
>
> *=sqrtpi(4) is 6.283185 = sqrt(4)*pi*

tan

The tan(num) function provides the tangent of a number.

Syntax: =tan(num)

tanh

The tanh(num) function provides the hyperbolic tangent of a number.

Syntax: =tanh(num)

trunc

The trunc(num) function returns the next lower whole-numbered value of the value, rounded down. Negative values are rounded in direction to 0.

Syntax: =trunc(num)

> ### ⮞ Example
>
> =trunc(2.2) is 2
>
> *=trunc(-2.2) is -2*

Recipegroup Manager functions

Variables from other loaded projects that are used in recipes can also be displayed with the report functions for the recipe group manager. These functions can only be used with screens of the recipegroup manager type.

The function assistant (on page 48) is available for configuration.

recipea

The =recipeaInhalt) function displays general data of a recipe of the recipe group manager.

Syntax: =recipea(Inhalt)

Contents	Description
muser	User who last edited the recipe.
mzeit	Time of the last change.
rez	Name of the recipe.
rezg	Name of the recipe group to which the recipe belongs.
reznum	Number of the recipe
text1 to text 8	User comments 1 to 8 of the recipe.
text2	User comment 2 for the recipe.
version	Version number of the recipe.
status	Status of the recipe as text. For example: **1 - sample recipe**

Example

=recipea(recnum)

recipef

The `=recipef(content,variable)` function displays variable-orientated data of a recipe of the recipe group manager. These data can freely be positioned in the report.

`Syntax: =recipef(content,variable)`

Contents	Description
name	Name of the variable.
identification	Identification of the variable.
type	Source type.
unit	Technical unit of measurement the variables.
value	Value for the variable in the recipe.
symbols	Display of limit value texts from the reaction matrix or limit value as text.
min	Minimum of the permitted value range for the variable in this recipe.
max	Maximum of the permitted value range for the variable in this recipe.
actual_value	If the defined variable does not exist in the selected recipe, the cell is marked with –.
Interlocking	Display of Interlocking. Because the names are not available in Runtime, only the IDs of the interlockings selected in the recipe group manager are displayed.
visibility	Shows visibility of the variables.
shadow variable	Shows the name of the linked shadow variable. Only display in the Runtime. Configuration in the Editor.
filter	Filter text that was selected in the RGM.
Keyboard	Shows linked keyboard keyboard.
Set value limit	Yes: Takes over set value limit to the variable. Only display in the Runtime. Configuration in the Editor.

Example

=recipef("value","temperature")

recipew

The =recipew(content,direction) function displays variable-orientated data of a recipe of the recipe group manager in table form.

Syntax: =recipew(content,direction)

Contents	Description
name	Name of the variable.
identification	Identification of the variable.
type	Source type.
unit	Technical unit of measurement the variables.
value	Value for the variable in the recipe.
symbols	Display of values as text.
min	Minimum of the permitted value range for the variable in this recipe.
max	Maximum of the permitted value range for the variable in this recipe.
actual_value	If the defined variable does not exist in the selected recipe, the cell is marked with −.
Interlocking	Display of Interlocking. Because the names are not available in Runtime, only the IDs of the interlockings selected in the recipe group manager are displayed.
visibility	Shows visibility of the variables.
shadow variable	Shows the name of the linked shadow variable. Only display in the Runtime. Configuration in the Editor.
filter	Filter text that was selected in the RGM.
Keyboard	Shows linked keyboard keyboard.
Set value limit	Yes: Takes over set value limit to the variable. Only display in the Runtime. Configuration in the Editor.

Example

=recipew(name,bottom)

Statistical functions

Common statistical functions are available for reports. These can not only be planned using the function assistant, but can also be directly entered into the input row.

avg

The `avg (range)` function calculates the average of the values within the range.

`Syntax: =avg(range)`

EXAMPLE

Cell A4= avg(A1:A3)

Row/column	A
1	1
2	3
3	5
4	3

The average of the cells A1 to A3 is 3

count

The `count (range)` function returns the number of numerical values in the stated range.

`Syntax: =count(range)`

EXAMPLE

Cell `A5= count(A1:A4)`

Row/column	A
1	1
2	Text
3	
4	3
5	2

The number of numerical values in the cells A1 to A4 is `2`.

counta

The `counta (range)` function calculates the number of non-empty cells in a given range.

`Syntax: =counta(range)`

EXAMPLE

Cell `A5=counta(A1:A4)`

Row/column	A
1	1
2	text
3	
4	3
5	3

The number of non-empty cells in the range `A1` to `A4` is `3`.

countblank

The `countblank (range)` function calculates the number of empty cells in a given range.

`Syntax: =countblank(range)`

EXAMPLE

Cell`A5 = countblank(A1:A4)`

Row/column	A
1	1
2	Text
3	
4	3
5	1

The number of empty cells in the range A1 to A4 is `1`.

max

The `max (range)` function calculates the highest value in a range.

`Syntax: =max(range)`

EXAMPLE

Cells A1 to A10: any numbers

Cell `B1=max(A1:A10)`

Row/column	A	B
1	1	10
2	2	
3	3	
4	4	
5	5	
6	6	
7	7	
8	8	
9	9	
10	10	

The highest value in range A1 to A10 is in cell A10 and is `10`.

min

The `min (range)` function calculates the lowest value in a range.

Syntax: `=min(range)`

EXAMPLE

Cells A1 to A10: any numbers

Cell B1=min(A1:A10)

Row/column	A	B
1	1	1
2	2	
3	3	
4	4	
5	5	
6	6	
7	7	
8	8	
9	9	
10	10	

The lowest value in range A1 to A10 is in cell A1 and is `1`.

percentile

The `percentile (range,num)` function returns as result a threshold value, from which the values are accepted.

Syntax: `=percentile(range,num)`

`Range:` Is a matrix or a data range that describes the relative position of the data.

`Num:` Integer from `0-100`

EXAMPLE

Cells A1 to A10: any numbers

Cell B1=percentile(A1:A10,90)

Row/column	A	B
1	1	9.1
2	2	
3	3	
4	4	
5	5	
6	6	
7	7	
8	8	
9	9	
10	10	

The threshold value for the range A1 to A10 is `9.1`.

stdev

The `stdev (range)` function returns an estimation of the standard deviation based on a random sample of the values in the defined area.

Formula:

$$s = \sqrt{\frac{n \sum x^2 - \left(\sum x\right)^2}{n(n-1)}}$$

`Syntax: =stdev(range)`

EXAMPLE

Cells A1 to A10: any numbers

Cell A11=stdev(A1:A10)

Row/column	A
1	745.00
2	730.00
3	732.00
4	768.00
5	769.00
6	747.00
7	718.00
8	723.00
9	766.00
10	739.00
11	18.785633

The standard deviation for the area A1 to A10 is estimated using on the sample on `18.785633` as a basis.

stdevp

The `stdevp (range)` function returns an estimation of the standard deviation based on the total of the values in the defined area.

Formula:

$$s = \sqrt{\frac{n \sum x^2 - (\sum x)^2}{n^2}}$$

`Syntax: =stdevp(range)`

EXAMPLE

Cells A1 to A10: any numbers

   Cell A11= stdevp(A1:A10)

Row/column	A
1	745.00
2	730.00
3	732.00
4	768.00
5	769.00
6	747.00
7	718.00
8	723.00
9	766.00
10	739.00
11	17.821616

The standard deviation for the range A1 to A10 is estimated at `17.821616`.

sum

The `sum (range)` function calculates the sum of the values in the defined area.

`Syntax: =sum(range)`

EXAMPLE

Cells A1 to A10: any numbers

Cell B1=sum(A1:A10)

Row/column	A	B
1	1	55
2	2	
3	3	
4	4	
5	5	
6	6	
7	7	
8	8	
9	9	
10	10	

The sum of all values in the A1 to A10 range is 55.

sumsq

The sumsq (range) function calculates the sum of the squares of the values within a range.

Syntax: =sumsq(range)

EXAMPLE

Cells A1 to A10: any numbers

Cell A11=sumsq(A1:A10)

Row/column	A	B
1	1	1
2	2	4
3	3	9
4	4	16
5	5	25
6	6	36
7	7	49
8	8	56
9	9	81
10	10	100
11	385	385

SUPERVISION

Cell B1 to B10: Squares of A1 to A10

Cell B11= sum(B1:B10)

var

The var (range) function returns an estimation of the variance based on a random sample of the values in the defined area.

Formula:

$$v = \frac{n\sum x^2 - (\sum x)^2}{n(n-1)}$$

Syntax: =var(range)

EXAMPLE

Cells A1 to A10: any numbers

Cell A11=var(A1:A10)

Row/column	A
1	745.00
2	730.00
3	732.00
4	768.00
5	769.00
6	747.00
7	718.00
8	723.00
9	766.00
10	739.00
11	352.9

The variance for the area A1 to A10 is estimated using on the sample on `352.9` as a basis.

varp

The `varp (range)` function returns an estimation of the variation based on the total of the values in the defined area.

Formula:

$$v = \frac{n \sum x^2 - (\sum x)^2}{n^2}$$

`Syntax: =varp(range)`

EXAMPLE

Cells A1 to A10: any numbers

Cell A11=varp(A1:A10)

Row/column	A
1	745.00
2	730.00
3	732.00
4	768.00
5	769.00
6	747.00
7	718.00
8	723.00
9	766.00
10	739.00
11	317.61

The variance for the range A1 to A10 is estimated at `317.61`.

Text functions

Different text functions are available for reports. These can not only be planned using the function assistant, but can also be directly entered into the input row. Exception: paint (on page 130)

char

The function `char(code)` returns the ASCII character that is defined under `code`.

`Syntax: =char(code)`

`code`: number between `1` and `255`.

> **Example**
>
> `=char(88)` is X
>
> *If the cell A1 contains the value* `35`*, the* `=char(A1)` *function gives the character* `#` *as a result.*

To determine the ASCII code for a character, use the code (on page 127) function.

clean

The `clean(string)` function deletes all control characters from a text character chain.

`Syntax: =clean(string)`

code

The `code(string)` function provides the ASCII code for the first character in a text character chain.

If the string is in the brackets of the text to be coded, it is put in quotation marks. If it refers to another row, this is entered.

`Syntax: =code(string)`

> ### Example
>
> `=code("z")` is `122`.
>
> *Cell A1 contains the text* `zoom`*:*
>
> `=code(A1)` is `122`.

To determine the character corresponding to an ASCII code, use the char (on page 126) function.

> ### ⚠ Attention
>
> *The* `code(string)` *function only provides the ASCII code for the first character in a text string.*

find

The `find(keystring,string,start)` function provides the position of a character sequence (`keystring`) within a text string (`string`). The position is found in the text string starting at the start character (`start`) with case-sensitivity.

`Syntax: =find(keystring,string,start)`

> **Example**
>
> Cell A1: `cd`
>
> Cell A2: `ABCDabcd`
>
> Cell A3: `1`
>
> Cell A4=`find(A1,A2,A3)`: Result =`7`

fixed

The `fixed(num,decimals)` function formats a number and converts it into a defined character string.

`Syntax: =fixed(num,decimals)`

- ▸ `num`: Numerical value or cell reference

- ▸ `decimals`: Number of decimals

  > **Example**
  >
  > **Cell A1** = `12345.672`
  >
  > =`fixed(A1,2)` is `12345.67`
  >
  > =`fixed(A1,1)` is `12345,6`
  >
  > The separator and the display also depend on the regional settings and formatting settings in cell (on page 29).

Note: The fixed() function cannot be combined with a format string such as the stating of units.

DIFFERENCE TO ZENON 5.50 AND EARLIER VERSIONS

The `fixed()` function also has the argument `no_seps` in versions up to 5.50. The argument is optional and controls the display of thousand separators.

`Syntax: =fixed(num,decimals,no_seps)`

- ▸ `no-seps`:

- 0: A separator is inserted
- 1: A separator is left out

Figures are generally shown without thousand separators in later versions.

left

The `left(string,num)` function provides the number of characters from a character string defined in (`num`), starting with the first character.

Syntax: `=left(string,num)`

▶ `string`: Character string in quotation marks or reference to a cell with a character string

▶ `num`: Number of characters that are to be read off, starting with the first

> **Example**
>
> Cell A1 = `abcdefgh`
>
> `=left(A1,3)` is `abc`
>
> *`=left(A1,5)` is `abcde`*

To read off characters beginning with the last character of a character string, use the right (on page 132) function.

len

The `len(string)` function provides the number of characters in a character string.

Syntax: `=len(string)`

> **Example**
>
> Cell A1 = `abc`
> `=len(A1)` is 3
>
> *Cell A1 = `abcdefgh`*
> *`=len(A1)` is 8*

lower

The `lower(string)` function replaces all upper case letters of a text character string with lower case letters.

```
Syntax: =lower(string)
```

> **Example**
>
> *Cell A1 = AbCdEf*
>
> *=lower(A1) is abcdef*

To replace lower case letters with upper case letters, use the upper (on page 135) function.

mid

The `mid(string,numStart,numLen)` function extracted

- ▸ From the defined character string `string`

- ▸ A new character string

- ▸ With a number of characters defined by the `numLen` argument

- ▸ From the start position defined by `numStart`.

```
Syntax: =mid(string,numStart,numLen)
```

> **Example**
>
> *Cell A1 = abcdefgh*
>
> *=mid(A1,2,4) is bcde*

paint

The `=paint(referenz)` function continues the character string given in the referenced cell, if there is not enough space for this in the referenced cell. Thus long character strings can be well displayed by being divided into several cells.

```
Syntax: =paint(referenz)
```

> ### ▣ Example
>
> *Cell A1 contains the characters:* `abcdefghijklmn`
>
> *In cell A1, the following is displayed due to the width of the cell:* `abcdefg`
>
> `=paint(A1)` *in cell A2 leads to this display:*
>
> *A1:* `abcdefg`
>
> *A2:* `hijklmn`

This text function can be configured with the function assistant.

proper

The `proper(string)` function transforms the first character of a word into upper case letters and all subsequent characters into lower case letters. The start of the word is the first alphabetical character of a character string. If the character string contains several non-alphabetic characters, a new word is started after each non-alphabetic character.

```
Syntax: = proper (string)
```

> ### ▣ Example
>
> Cell A1 = `AbCdEf`
> `=proper(A1)` is `Abcdef`
>
> Cell A1 = `aBcDeF`
> `=proper(A1)` is `Abcdef`
>
> *Cell A1 = 1aBcD2eF*
> *=proper(A1) is 1Abcd2Ef*

replace

The function `replace(string,numStart,numLen,stringNew)`

- ▶ replaces, in the text character string , `string`

- ▶ from the start position defined by `numStart`,

▶ a number of characters defined by `numLen`

▶ with characters from the new text character string, stringNew

Syntax: `=replace(string,numStart,numLen,stringNew)`

📭 Example

Cell A1 = `abcdefgh`
Zelle A2 = `xyz`
`=replace(A1,3,2,A2)` is `abxyzefgh`

In this example, in the string `abcdefgh`, the characters `cd` (3rd position in the string, 2 characters) are replaced with `xyz`.

---

rept

The `rept(string,num)` function repeats the character string `string` for as many times as is defined in `num`.

Syntax: `=rept(string,num)`

📭 Example

Cell A1 = `abc`
Cell A2 = 3
=rept(A1,A2) is `abcabcabc`

---

right

The `right(string,num)` function provides the number of characters from a character string defined in (`num`), starting with the last character.

Syntax: `=right(string,num)`

▶ `string`: Character string in quotation marks or reference to a cell with a character string

▶ `num`: Number of characters that are to be read off, starting with the last

> ### Example
>
> Cell A1 = `abcdefgh`
>
> `=right(A1,3)` is `fgh`
>
> *=right(A1,5) is defgh*

To read off characters beginning with the first character of a character string, use the left (on page 129) function.

search

The `search(keystring,string,start)` function provides

- ▶ the position of a search text `keystring`

- ▶ within a character string `string`

- ▶ from the start character position `start`

`Syntax: =search(string,keystring,start)`

EXAMPLE

Cell `A4=search(A1,A2,A3)`

Cell/column	A
1	`abcdeABCDEabcde`
2	`cd`
3	`1`
4	`3`

The function in cell A4 finds the text being searched for `cd` in cell A1 at position `3`.

substitute

The function `substitute(string,oldstring,newstring,instance)`

- ▶ replaces, in the character string , `string`

- ▶ an `oldstring` character sequence

- ▶ with a new character sequence `newstring`

- ▶ whereby `instance` determines the position of
  - • `instance=0` replaces all character strings found

The function is case sensitive.

```
Syntax: =substitute(string, oldstring, newstring, instance)
```

Example

Cell A1 = `abcdef ABCDEF abcdef ABCDEF abcdef ABCDEF`
Cell A2 = `XYZ`

Example 1:

`=substitute(A1,"ABCDEF",A2,0)` is `abcdef XYZ abcdef XYZ abcdef XYZ`

In this example, the whole character sequence `ABCDEF` in the entire string `abcdef ABCDEF abcdef ABCDEF abcdef ABCDEF` is replaced by `XYZ`

Example 2:

`=substitute(A1,"abcdef",A2,2)` is `abcdef ABCDEF XYZ ABCDEF abcdef ABCDEF`

In this example, the second character sequence `abcdef` *found in the string* `abcdef ABCDEF abcdef ABCDEF abcdef ABCDEF` *is replaced by* `XYZ`.

Note: Character string within the function are put into quotation marks ("character string").

trim

The `trim(string)` function allows a maximum of one space between two characters. All other spaces are removed.

```
Syntax: =trim(string)
```

Example

Cell A1 = `Too many spaces are reduced to one.`
=trim(A1) is: `Too many spaces are reduced to one.`

upper

The `upper(string)` function replaces all lower case letters of a text character string with upper case letters.

`Syntax: =upper(string)`

> **Example**
>
> *Cell A1 = AbCdEf*
> *=upper(A1) is ABCDEF*

To replace upper case letters with lower case letters, use the lower (on page 129) function.

Other functions

Other functions group together report functions that cannot be allocated to a sub-group.

The function assistant (on page 48) is available for configuration.

Alarm

This function shows information from the alarm administration as a formatted list in a report form.

`Syntax: =alarm(FILTER, PARAMETER, DIRECTION)`

Transfer parameters	Valid range	Comment
FILTER		Configuration via AML filter.
PARAMETER	‣ name ‣ identification ‣ text ‣ status ‣ value ‣ time received ‣ time sent ‣ time acknowledged ‣ time reactivated ‣ status reactivated ‣ number reactivated ‣ active ‣ number_h ‣ number_t ‣ number_m ‣ user ‣ computer ‣ note ‣ class ‣ group	Selection of the filter column to be read. Hints: ‣ project: Only present in the AML filter with multi-user projects ‣ number: only for one project ‣ number_h: only for one project ‣ number_t: only for one project ‣ number_m: only for one project

	▸ `project`	
	▸ `number`	
`DIRECTION`	▸ `bottom`	Filling in the corresponding direction.
	▸ `left`	

The time filter results from the definition in Screen switching filter report (on page 158).

Direction

Options	**Meaning**
`right`	Value is entered to the right (row).
	There must be sufficient cells left free for expected entries from further report creation.
	Attention: If an `archivesp` function is created with only one column, the value is entered downwards (in the column).
`bottom`	Value is entered downwards (column).
	There must be sufficient cells left free for expected entries from further report creation.
	Attention: If an `archivesp` function is created with only one row, the value is entered to the right (in the row).

> 💡 **Info**
>
> *To ensure that the function can dynamically insert lines or columns without overwriting other lines/columns, the labeling of the corresponding row (on page 42) or column (on page 41) must be set to E (for "Expand").*

cel

This function shows information from the chronological event list (CEL) as a formatted list in a report. Configuration with the function assistant works along the lines of configuration of the `aml` (on page 139) function.

Syntax: `=cel(FILTER,PARAMETER,DIRECTION)`

Transfer parameters	Valid range	Comment
FILTER		Configuration via CEL filter.
PARAMETER	▸ name ▸ identification ▸ text ▸ status ▸ value ▸ time received ▸ user ▸ computer ▸ note ▸ class ▸ group ▸ project ▸ number	Selection of the filter column to be read. Hints: ▸ project: Only present in the CEL filter with multi-user projects ▸ number: only for one project
DIRECTION	bottom, left	Filling in the corresponding direction.

The time filter results from the definition in Screen switching filter report (on page 158).

Direction

Options	Meaning
`right`	Value is entered to the right (row).
	There must be sufficient cells left free for expected entries from further report creation.
	Attention: If an `archivesp` function is created with only one column, the value is entered downwards (in the column).
`bottom`	Value is entered downwards (column).
	There must be sufficient cells left free for expected entries from further report creation.
	Attention: If an `archivesp` function is created with only one row, the value is entered to the right (in the row).

Info

To ensure that the function can dynamically insert lines or columns without overwriting other lines/columns, the labeling of the corresponding row (on page 42) or column (on page 41) must be set to E (for "Expand").
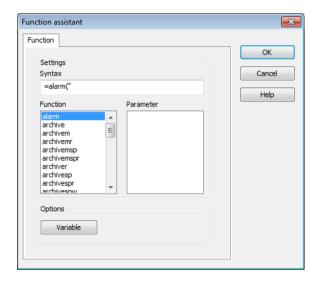
Configuring the alarm and CEL with the function assistant

You can configure functions for AML and CEL with the function assistant. This example is carried out with `aml` and works along the lines of `cel`:
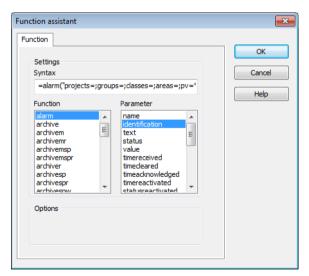
1. Select the desired cell in the context menu with the `Report functions` command.

2. The assistant is opened.

3. Select `Alarm`.
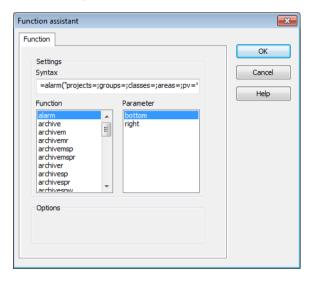
4. The `alarm` report function is entered into the syntax.

5. Clicking on **Variable** opens the Filter for AML. (cel: Filter for CEL.)

6. Configure the filter.

7. Select the desired parameter by double clicking, for example `identification`).

8. The parameter is transferred into the string.

9. Select the desired direction of the entry in the table by double clicking:

   - `below`: The function fills the column

- `right`: The function fills the row

10. The direction is transferred into the string.

11. The brackets are closed.

12. The function is transferred to the cell by clicking on OK.
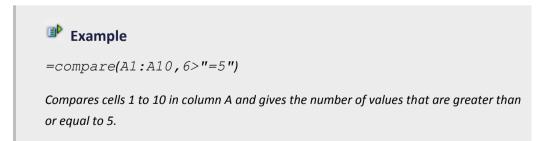
compare

The `compare(range,"condition")` function compares values in the stated range with a condition and returns the number of hits.

`Syntax: =compare(range,"condition")`

`Area:` cells that are used for the comparison

`Condition:` Content for which the examination is relevant (information always in quotation marks)

> 📋 **Example**
>
> *=compare(A1:A10,6>"=5")*
>
> *Compares cells 1 to 10 in column A and gives the number of values that are greater than or equal to 5.*

comparesp

The `comparesp(start cell,direction,P1,P2,P3,P4,"condition")` function compares values in a given area in a special format with a condition and returns the number of the hits from a reference area.

Syntax: `=comparesp(start cell,direction,P1,P2,P3,P4,"condition")`

Parameter	Description
`Start cell`	Cell with the first value.
`Direction`	"right" or "bottom"
`P1`	Number of values in the first row.
`P2`	Empty cells between two value outputs.
`P3`	Empty cells until the next value row.
`P4`	Maximum number of values to compare.
`Condition`	condition. Information always in quotation marks.

CONTROL CELLS

The parameters `P1`, `P2` and `P3` set out which cells of the report can be filled:

- ▶ `P1`: Number

- ▶ `P2`: Empty cells between two value outputs, depending on `DIRECTION` (`down`, `right`) either empty rows or empty columns

- ▶ `P3`: Empty cells between two value outputs, depending on `DIRECTION` (`down`, `right`) either empty rows or empty columns

All entries of the time range of the report are shown in the report, going downwards or tothe right from the cell in which the function was given, with the row format defined by the parameters P1-P3.

WHEN SELECTING ᴅᴏᴡɴ, THE FOLLOWING APPLIES:

- ▶ `P1` = number of rows to be filled with values

- ▶ `P2` = empty rows between the values

> ► `P3` = empty columns

WHEN SELECTING `RIGHT`, THE FOLLOWING APPLIES：

> ► `P1` = number of columns to be filled

> ► `P2` = empty rows between the values

> ► `P3` = empty rows between the values

> 📨 **Example**
>
> *=comparesp(A1,downwards,12,0,0,12,"=5")*

comparespsum

The `comparespsum(start cell,direction,P1,P2,P3,P4,"condition")` function compares values in a given area in a special format with a condition and returns the sum of the hits from a reference area.

Syntax: `=comparespsum(start cell,direction,P1,P2,P3,P4,"condition")`

Parameter	Description
`Start cell`	Cell with the first value.
`Direction`	"right" or "bottom"
`P1`	Number of values in the first row.
`P2`	Empty cells between two value outputs.
`P3`	Empty cells until the next value row.
`P4`	Maximum number of values to compare.
`Condition`	condition. Information always in quotation marks.

CONTROL CELLS

The parameters `P1`, `P2` and `P3` set out which cells of the report can be filled:

▶   `P1`: Number

▶   `P2`: Empty cells between two value outputs, depending on `DIRECTION` (`down`, `right`) either empty rows or empty columns

▶   `P3`: Empty cells between two value outputs, depending on `DIRECTION` (`down`, `right`) either empty rows or empty columns

All entries of the time range of the report are shown in the report, going downwards or tothe right from the cell in which the function was given, with the row format defined by the parameters P1-P3.

WHEN SELECTING `DOWN`, THE FOLLOWING APPLIES:

▶   `P1` = number of rows to be filled with values

▶   `P2` = empty rows between the values

▶   `P3` = empty columns

WHEN SELECTING `RIGHT`, THE FOLLOWING APPLIES:

▶   `P1` = number of columns to be filled

▶   `P2` = empty rows between the values

▶   `P3` = empty rows between the values

Example

`=comparespsum(A1,downwards,12,0,0,12,"=5")`

comparesum

The `comparesum( comparison range,sum reference,"condition")` function compares values in the stated area with a condition and returns the sum of the hits from a reference area.

`Syntax: =comparesum(comparison range,sum reference,"condition")`

▶   `Area:` cells that are used for the comparison

▶   `Sum reference:` value that is given if the condition is met

`Condition`: Content for which the examination is relevant (information always in quotation marks)

> ### Example
>
> *=comparesum(B2:B11,A2,"=5")*
>
> *Cells B2 to B11 are checked for =5. In the lines where this is true, the values of the cell are added to column A.*

computername

The `computername()` function provides the computer names of the computers on which the report is executed.

`Syntax: =computername()`

No transfer parameters are needed.

maxsp

The `maxsp` function determines the highest value in a range und displays it. The format can be a color or underlined.

`Syntax: =maxsp(range,num)`

Transfer parameters	Valid range	Comment
`range`	`A1-ATWH31000`	Maximum number of columns: `31000` Maximum number of rows: `31000`

`num`	`-1`
	
	`0-15`
		`-1` : underlined
		`0 ... 15:` Color according to the font color in the Create new font dialog (screens/fonts). You can find the allocation list under this table.

ALLOCATION OF FONT COLORS

The font colors are taken from the operating system. To make it easier for you, you can find a list of colors as they are used in Windows:

Color	Number
	0
	1
	2
	3
	4
	5
	6
	7
	8
	9
	10
	11
	12
	13
	14
	15

Note: If colors are rearranged or redefined in the operating system, the colors defined for `maxsp` also change.

EXAMPLE

The highest value is to be displayed as underlined in report A1 to B4.

INPUT IN THE EDITOR:

Row/column	A	B	C
1	8	5	=maxsp(A1:B4,-1)
2	20	2	
3	12	10	
4	32	14	

RESULT IN RUNTIME

```
      A    B
01   08   05
02   12   02
03   12   10
04   32   14
```

minsp
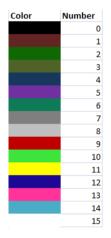
The `minsp(range,num)` functions determine the highest value in a range und displays it. The format can be a color or underlined.

Syntax: `=minsp(range,num)`

Transfer parameters	Valid range	Comment
`range`	`A1-ATWH31000`	

`num`	`-1`
	
	`0-15`
		`-1:` underlined
		`0 ... 15:` Color according to the font color in the Create new font dialog (screens/fonts). You can find the allocation list under this table.

ALLOCATION OF FONT COLORS

The font colors are taken from the operating system. To make it easier for you, you can find a list of colors as they are used in Windows:

Note: If colors are rearranged or redefined in the operating system, the colors defined for `minsp` also change.

Example

Cell C1=`minsp(A1:B4,-1)`

Parameter

Parameters make it possible to define a report that is executed via the zenon `Execute report` function and different transfer parameters used in the process perform different tasks. For example, a single report can be used to display the archives of several controls. Or a report displays values in different units of measurement.`Syntax: =parameter(Number)`

Ten parameters are available:
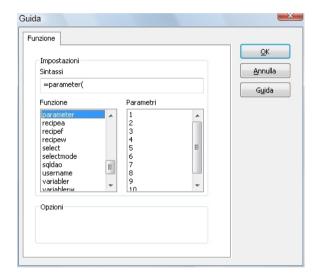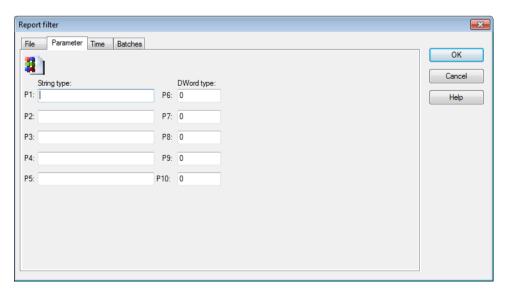
▶ 1 - 5: String type

▶ 6 - 10: DWord type

ENGINEERING

1. Allocate the desired parameters to the appropriate cells in the report

2. Supplement the parameters with any computer operations that may be required or reference them

3. In Runtime, select values for the parameters using the `Execute report` function or the `Open file` button and the `Parameters` tab

Hint: If you would like to display data from different archives in a report, for example five PLCs, then define a joint report. You control the display via five zenon functions that provide the parameters. You can simply copy these functions, however you must always only change the transfer values in the Parameters tab.

EXAMPLES OF PARAMETERS

▶ =parameter(1): allocates parameter (1) to the cell

▶ =parameter(6)*A1/100: allocates parameter (6) to the cell, multiplies the value that was set in the Parameter tab by the value from cell A1 and divides this by 100

select

The =select(cell address) function defines the cell that should be selected on opening the report. As a parameter the cell reference has to be stated.
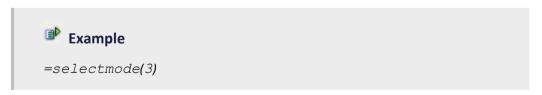
Syntax: =select(cell address)

> **Example**
>
> *=select(B3)*

selectmode

The selectmode(selection mode) function defines the selection mode.

Syntax: =selectmode(selection mode)

> **Example**
>
> *=selectmode(3)*

user name

The username function username returns the user names that are currently logged in to zenon.

Syntax: =username()

No transfer parameters are needed.

variabler

The =variabler(`variable name`,`property`) function provides information on a variable.

Syntax: =variabler(`variable name`,`property`)

Parameter	Description
`name`	Variable name.
`text`	Current limit text in the event of a limit being exceeded.
`value`	Current value of the variable.
`unit`	Unit of the variable.
`identification`	Identification of the variable.
`mmin`	Minimum measuring range of the variable.
`mmax`	Maximum measuring range of the variable
`status`	Current status of the variable.
`gw1`	Limit 1 of the variable.
`gw2`	Limit 2 of the variable.
`gw3`	Limit 3 of the variable.
`gw4`	Limit 4 of the variable.

Info

The report is created more quickly if the variables called up are advised.

CONFIGURATION WITH THE FUNCTION ASSISTANT

1. Select the **Report functions** report table in the context menu

2. Select `variabler`

3. click on the **Variable** button

4. Select the desired variable:

    Several variables can be selected; these are entered in a downward direction, starting from the cell in which the report function is defined

5. select the property to be displayed

> **Example**
>
> *=variabler("WIZ_VAR_10","name")*

variablerw

The variablerw(`variablename,property,action`) function provides information on variables and writes values in variables.

Syntax: =**variablerw**(`variable name,property,action`)

Parameter	Description
name	Variable name.
text	Current limit text in the event of a limit being exceeded.
value	Current value of the variable.
unit	Unit of the variable. Note: may not be used together with unit conversion. Only for use with older projects. If a unit of a variable defined in the unit conversion is changed with this report command in Runtime, configuration must be carried out again in unit switching.
identification	Identification of the variable
mmin	Minimum measuring range of the variable. Note: may not be used together with unit shift.
mmax	Maximum measuring range of the variable Note: may not be used together with unit shift.
status	Current status of the variable.
gw1	Limit 1 of the variable.
gw2	Limit 2 of the variable.
gw3	Limit 3 of the variable.
gw4	Limit 4 of the variable.

🔆 Info

The report is created more quickly if the variables called up are advised.

CONFIGURATION WITH THE FUNCTION ASSISTANT

1. Select the `Report functions` report table in the context menu

2. Select `variabler`

3. click on the **`Variable`** button

4. Select the desired variable:

   Several variables can be selected; these are entered in a downward direction, starting from the cell in which the report function is defined

5. select the property to be displayed

6. Enter the desired action

> **Example**
>
> *=variablerw("WIZ_VAR_10","value","read"*

Formatting keys

The formatting keys define the formatted output of cell details depending on the cell contents (data types).

The definition is made:

▶ In the drop-down list *Format -> Cells -> Numbers -> Cell type tab -> General/format string*

or

▶ In the context menu Report table *-> Cell -> Cell type tab -> General/format string*

Numerical data types

Format keys for numerical data types are symbolic dummies that are replaced by the formatted value.

▶ The comma is used as a thousand-divider; a period is used for decimal points.

▶ Each character which does not represent a format key is transferred into the output as a text character.

▶ If no format key is given, the format *0.* is used as a standard setting.

Note: A maximum of 15 decimal digits are permitted with the standard setting.

Format key	Description
, (comma)	Inserts the thousand-separator three digits after the decimal separator.
. (period)	Inserts the decimal separator.
; (semicolon)	Separates the format key for positive and negative value ranges.
#	Is only replaced if the corresponding place is also available.
*	All places before and after the decimal point are displayed.
0 (Cero)	If the position in the value does not exist then it is filled with zero.
/	The following character is treated as a text character string.

EXAMPLES

Data	Format key	Formatted text	Description
-0.2	*0.00	-0.20	Two decimal places are always output.
123	*00000	00123	The first 5 places are output with leading zeros.
104	->*<-	->104<-	Numbers are basically output between the ->...<- characters.
0.25 00	*0.*	12:25 AM	For values between -1.0 and 1.0, a leading zero is always output.
12	+*0.00;-*0.00	+12.00	Positive values are indicated with + and negative values with -.

Logical data types

Yes/No information can be displayed in different ways with logical data types. The output text is determined using the formatting key.

▶ Format: Text1;Text2

▶ Output:

- Positive values including zero = `Text 1`
- Negative values = `Text 2`

Output information can be described independently of the cell formatting with the `if (on page 92)(exprTest,valTrue,valFalse)` function.

EXAMPLE

Data	Format key	Formatted text	Description
`1`		Yes	If no formatting key is given, true/false is used as a default.
`-1`	Yes;No	No	

Data types for date and time

A variety of formatting keys are available for the output of date and time details. In input mode and for output without formatting keys, the output format defined under the windows country settings is used.

> **Info**
>
> *Descriptions for days and months can be configured as language switchable in Runtime.*

Each character which does not represent a format key is transferred into the output as a text character.

Format key	Value	Description
min	1 – 12	Month numerical short
mm	01 – 12	Month numerical long
mmm	Jan – Dec	Month short
mmmm	January – December	Month long
d	1 – 31	Day short
dd	01 – 31	Day long
ddd	Sun – Sat	Weekday short
dddd	Sunday – Saturday	Weekday long
yy	00 – 99	Year short
yyyy	1700 – 2900	Year long (0x is interpreted as 190x)
/		International date separator
h	0 – 23 or 1 – 12	Hour short, either 24 h or 12 h with ap
hh	00 – 23 or 01 – 12	Hour long, either 24 h or 12 h with ap
mm	00–59	Minutes, only after h or hh
ss	00–59	seconds
ff	000–999	Milliseconds
ap	am, pm (or international equivalent)	am or pm
: (colon)		International time separator

EXAMPLE

Data	Format key	Formatted text
22.09.95 2:05 PM	dddd dd/mm/yyyy	Friday 22.09.1995
22.09.95 2:05 PM	dd/mmmm yyyy	22 September 1995
22.09.95 2:05	dd/mm/yy hh:mm	22.09.95 2:05

PM		PM
22.09.95 2:05 PM	hh:mm:ff	`2:05:000 PM`

zenon functions for Report Generator

zenon provides functions to control the Report Generator in Runtime.

The screen switching (on page 158) function switches to the report screen and makes it possible to define filters.

Special functions make it possible to print, export, and execute a report in Runtime. To create zenon functions for the report generator:

1. In the functions context menu, navigate to: *New function -> Report generator*

2. select the desired function:
   - Print report (on page 172)
   - Export report (on page 174)
   - Report: execute (on page 176)

Report screen switching

To use a report in Runtime, configure screen switching to a report screen type:

1. Create a new **Report** screen type

2. Select the **New function** command in the **Functions** node

3. select the **Screen switching** function

4. select the **Report screen**

5. The report filter (on page 159) is displayed

6. select the desired properties

7. Link the functions with a button in the screen, in order to switch in Runtime
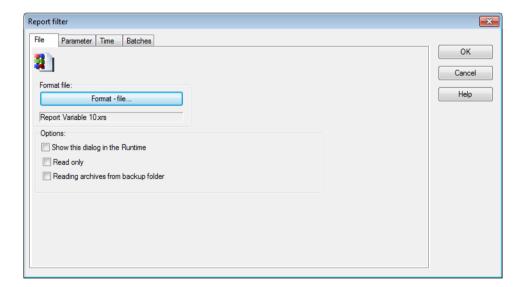
Report filter

Displayed when creating zenon functions for configuring reports:

▶ File (on page 159): Selection of report file and conditions in Runtime

▶ Tag (on page 160): Transfer parameters

▶ Time (on page 161): Time period

▶ Batches (on page 163): Filtering on batches

▶ Links (on page 165): Replace functions

▶ Indices (on page 169): Replace indices

File

In the 'File' tab, you define which report is called up with screen switching or if this is to be decided in Runtime.
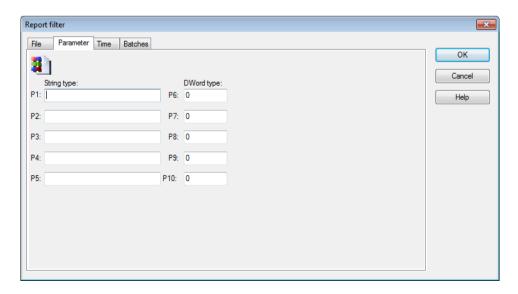
Parameter	Description
`Format file`	selection of the report file configured (on page 9) in the editor (*.xrs)
`Show this dialog in the Runtime`	`Active:` This filter dialog is offered before being called up in Runtime.
`Read only`	`Active:` Entries can only be read. It is not possible to change or save them.
`Reading archives from the read-back folder`	`Active:` Stored archives are read in from the read-back folder defined in project configuration. When loading archive data from the readback folder, the archive data from the Runtime path and from all subfolders of the readback folder is also read.

The file configuration is different for:

- ▶ Print report (on page 172)

- ▶ Export Report (on page 174)

- ▶ Report: execute (on page 176)

Parameter

Parameters make it possible to use a report for different archives. To do this, the fields to be used flexibly are defined with the `parameter(Number)` (on page 148) report function. The parameters to be used are set in the 'Parameters' tab of the screen switching function. For example, a single report can be used to display the archives of several controls. Or a report displays values in different units of measurement.

Ten parameters are available:
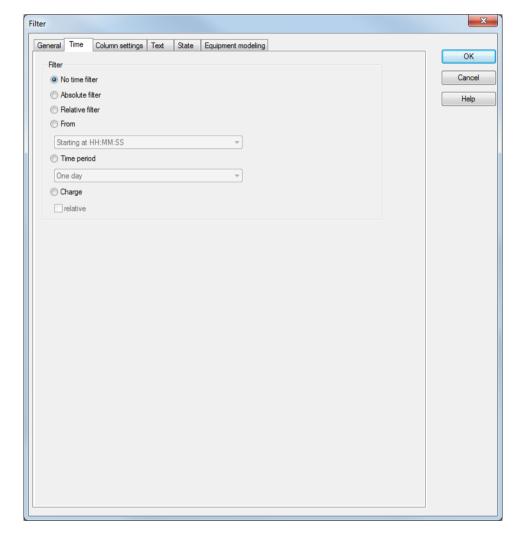
- ▶ 1 - 5: String type

- ▶ 6 - 10: DWord type

If, for example, the archive data of five PLCs is to be displayed, a report is created and the five parameters for the PLCs are transferred via the five switching functions.

Time

Time filters make it possible to limit the data to be displayed or exported. The time filters are very flexible to implement and can be pre-set in the editor or adjusted in Runtime.

Note: Time is saved in UTC. For details see chapter Handling of date and time in chapter Runtime.

Time filtering can be carried out in two ways:

1. Define time period in the Editor

   Fixed time areas are used. A time period is given in the editor. It is only possible to filter according to this time period in Runtime. Other filters - such as filtering according to variable name, alarm/event groups and alarm/event classes etc. can no longer be amended in Runtime.

2. Time filter amendable in Runtime

   Pre-defined times are used. The time filter is defined in the Editor and can be changed in Runtime as desired.

   > ### 💡 Info
   >
   > *If a screen that does not have a time filter is referenced, the time filters are deactivated.*
   >
   > *For the setting* `Set filter for time filter type` *the option* `Display filter dialog in Runtime` *can be selected, but it is not available in runtime.*

FILTER METHODS

You can read details on filtering in the alarm administration manual, in the following chapters:

▶ Specify time area in the Editor

▶ Time filter can be amended in Runtime

▶ Absolute filter

▶ Relative filter

▶ From

▶ Time period

The methods for time filtering are applicable for screen switching as well as export for:

▶ Alarm Message List

▶ Archive revision

▶ Chronological Event List

▶ Extended Trend

▶ Filter screens

▶ Report Generator (on page 5)

- ▶ Report Viewer (on page 210)

- ▶ Function Export Archives.

Lots

Batch filtering corresponds to filtering in the archives:

Parameter	Description
Lot filter	On the left side you can choose the desired archive from the available archives. On the right side the available lots are displayed. You can filter the lots there.
Lot name	In this column the names of the available lots are displayed. By left clicking the top part of the header, the lots are sorted alphabetically in an ascending or descending order. In the bottom part of the header you can enter a character string. Only lots matching the respective character string will be displayed.
Start date	In this column the start date of the available lots is displayed. By left clicking the top part of the header, the lots are sorted in an ascending or descending order. Lots with the same start date are sorted according to their start time. In the bottom part of the header you can enter a start date manually or use the displayed calendar.
Start time	Only available if you entered a start date. In this column the start time of the available lots is displayed. By left clicking the top part of the header, the lots are sorted in an ascending or descending order. In the bottom part of the header you can enter the start time manually. Note: '*' means 12:00:00 AM o' clock.
End date	In this column the end date of the available lots is displayed. By left clicking the top part of the header, the lots are sorted in an ascending or descending order. Lots with the same end date are sorted according to their end time. In the bottom part of the header you can enter an end date manually or use the displayed calendar.
End time	Only available if you entered an end date. In this column the end time of the available lots is displayed. By left clicking the top part of the header, the lots are sorted in an ascending or descending order. In the bottom part of the header you can enter the end time manually. Note: '*' means 11:59:59 PM o' clock.
Duration	This column displays the duration for each available lot. It is only for display.

Info

Still open lots are also displayed if they match the set filter criteria.

Info

The value of the lot variable is written in the index file and in the header of the ARX file at the start of the lot. These entries are adjusted with every change of the variable. When the lot is closed, the value of the lot variable at this moment is finally written in the index file and in the header.

Thus the lot name is final when the lot is closed.

Replace links
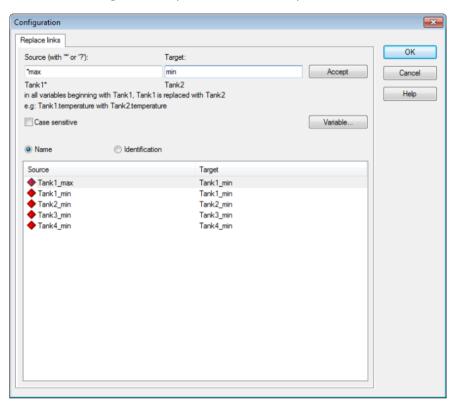
Linked variables or functions can be permanently replaced in the Editor in a rule-based manner. The replacement is not limited to the functions or variables defined in the properties of the `Variable / function` group. All other variables used the screen can, such as variables for `Visibility`, `Flashing`, `Size and rotation dynamic` and others.

To replace variables or functions in a rule-based manner in the element directly:

1. Highlight the desired element and open it by right clicking the element in the context menu

2. select `Replace Links`

3. The dialog for the replacement of links opens

Property	Description
Source	Enter the partial string to be searched for.
	Place holder * and ? can be used. Placeholders are only permitted as prefix or suffix; e.g. *xxx or xxx*.
Target	Entry of the partial string
Name	Swaps information in process variable names.
Identification	Exchanges information in the identification
Note capitalization	When swapping, be sure that any capitalization is an exact match.
Accept	Swaps target strings from the `source` for those defined in the `target`.
Variable/function	Opens the selection list for variables/functions in relation to the selected line in the list. Clicking on the variable in the list defines new target variables. Alternative: Double-click on the source variable in question.

REPLACE

A) REPLACE BY MANUAL SELECTION

▶ select the element from the list that you would like as the source

▶ select a target element via the `variable/function` button

▶ the previous element is replaced by the new one

B) AUTOMATED REPLACEMENT WITH RULES

▶ In the source input field, define the parameters for the element that you wish to replace

▶ define the parameter for the new variable/function in the target input field

▶ specify what is to be replaced via Name/Identification

▶ Click on `Accept`

> 💡 **Info**
>
> *The target variable or target function can also be in a different project as the source variable or source function.*

> ⚠ **Attention**
>
> *When replacing variables, be aware of the type and signal resolution. If you replace a variable with one of an incompatible type, this can lead to errors during execution. You will be warned when making the substitution; the substitution will however be carried out.*
>
> 💡 **Info**
>
> *Replace via Drag&Drop: Elements that can only contain one variable or function can be swapped by dragging & dropping. Drag the new variable or function to the element using the mouse. It automatically replaces the previous one.*
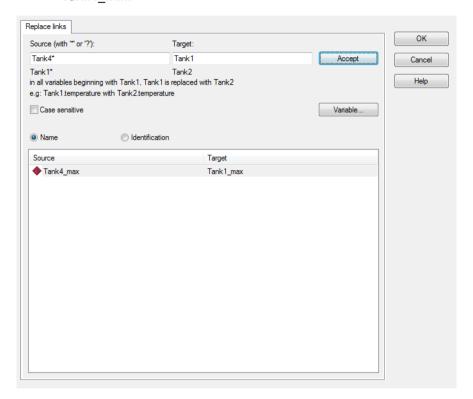
EXAMPLE

All variables with "Tank1" in the name are to be replaced with "Tank4".

1. Enter "Tank1" into the `Source` text field. You can include all variables that start with "Tank1" with the `*` character.

2. Enter "Tank4" into the target text field.

3. Click on Accept.
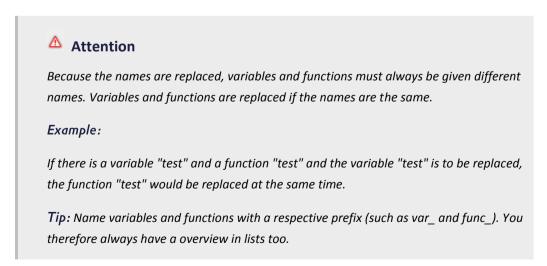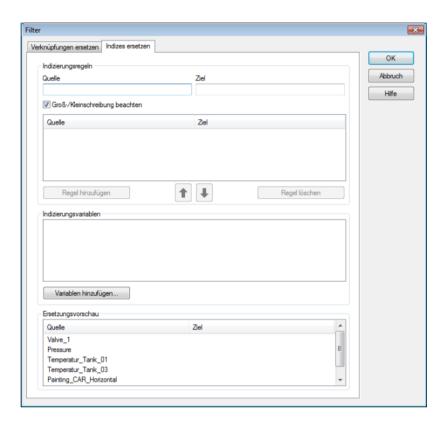
4. The variables Tank1_max and Tank1_min are replaced by the variables "Tank4_max and Tank4_min.

Replace indices

When switching screens in Runtime, variables can be replaced dynamically (virtually) using indexing rules. In doing so, variable values can be used as part of the replacement rule.

> ⚠ **Attention**
>
> *Because the names are replaced, variables and functions must always be given different names. Variables and functions are replaced if the names are the same.*
>
> *Example:*
>
> *If there is a variable "test" and a function "test" and the variable "test" is to be replaced, the function "test" would be replaced at the same time.*
>
> *Tip: Name variables and functions with a respective prefix (such as var_ and func_). You therefore always have a overview in lists too.*

To configure the replacement of indices:

1. All variables and functions used in the screen are displayed in the `Source` field of the **`Replacement preview`**.

2. Left-click the variable or function for which you want to create an indexing rule.

3. In the **`Indexing rules`** area, the variable/function is entered in the fields **`Source`** and **`Target`**. Alternatively, you can also enter the desired variable/function directly into the **`Source`** field in the **`Indexing rules`**.

4. Define the indexing rule.
   The dynamic part of the rule is inserted in curly brackets, for example {X01}.

Example:

The source variable **`Motors1.speed`** is replaced by the target **`Motors{X01}.speed`**, the placeholder **`x01`** is replaced with the respective value of the index variable in the Runtime.
The replacement rule would thus be:

Source: **`Motor1`**

Target: **Motor{X01}**

5. Carry this over to **Add Rule** by left clicking in the indexing rules.

6. All indexing rules that have been created are displayed in this list.

7. You can change the order of the entries using the button with the arrow symbol.

Attention: The substitutions are executed in the order in which they are displayed in the list!

8. Variables that are not present on the screen but are used in the replacement can be inserted via the **Add Variables...** button in the **Indexing Variables** window.

9. The variables which are in this list are always kept in the memory in order to minimize waiting time when the screen is loaded.

> 💡 **Info**
>
> *For the placeholder * the same rules as for Replace links are valid.*

> ⚠ **Attention**
>
> If index variables are used in a network project, the Client must first fetch the value from the Server. Then the calculation is executed. On devices with outdated hardware this may cause delay at screen switching.
> Therefore in a network project index variables should always be set to DDE active.

CROSS-PROJECT

If the target is in a different project to the source, this can be displayed using '#'.

> **Example**
>
> *Source: VAR_1*
>
> *Target: VAR_1 in Project_1*
>
> *You thus enter VAR_1 as the source and Project_1#VAR_1 as the target.*
>
> *Source: SUBPROJECT1#VAR_1*
>
> *Target: {X01}VAR_1*
>
> *The index variable X01 is a string variable and has the project name including # as a value, for example "SUBPROJECT4#".*
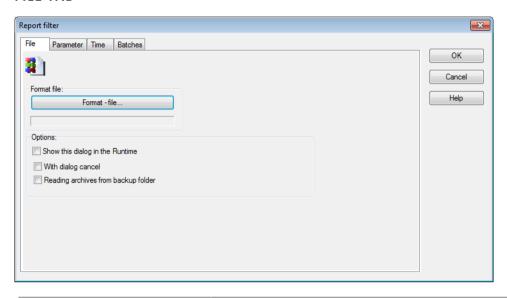
Print report

The `Print report` function makes it possible to print out a report without calling it up beforehand. To configure the function:

1. create a new function

2. Select, in the `Report generator` section, the `Execute report` entry

3. The dialog for the report filter (on page 159) is opened

4. Note: The `File` tab settings are different. The settings that are valid for `Print report` are clarified in this chapter; you can find all further settings in the report filter (on page 159) chapter.

5. select the desired properties

6. Link the functions with a button in the screen, in order to switch in Runtime

FILE TAB

| Parameter	Description
`Format file`	selection of the report file configured (on page 9) in the editor (*.xrs)
`Show this dialog in the Runtime`	`Active`: This filter dialog is offered before being called up in Runtime.
`With dialog cancel`	`Active:` A button to cancel printing is shown whilst the report is being prepared for print.
`Reading archives from the read-back folder`	`Active:` Stored archives are read in from the read-back folder defined in project configuration. When loading archive data from the readback folder, the archive data from the Runtime path and from all subfolders of the readback folder is also read.

Info

Archive entries can only be read with the `Print report` *function. Write actions are not possible.*

Export Report

This function is used to write a table into a file in the background during online operation. The export file is stored in the \Export sub-folder of the project folder. (open from zenon: highlight project, press `ctrl+alt+D`.) Provide the file parameters, the filter and the time as the transfer parameters.

The `Export report` function makes it possible to export a report without calling it up beforehand. To configure the function:

1. create a new function

2. Select, in the `Report generator` section, the `Export report` entry

3. The dialog for the report filter (on page 159) is opened

4. Note: The `File` tab settings are different. The settings that are valid for `Export report` are clarified in this chapter; you can find all further settings in the report filter (on page 159) chapter.

5. select the desired properties

6. Link the functions with a button in the screen, in order to switch in Runtime
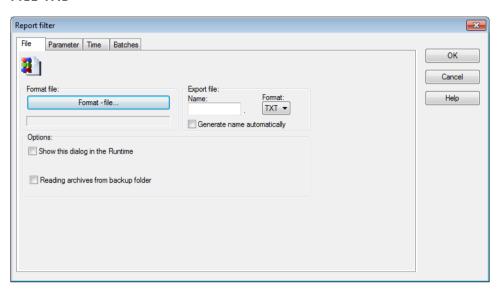
   > ⚡ **Info**
   >
   > *When exporting a report using the* `Export report` *function, the converted units are exported at units of measurement. The basic units are not a part of the export.*

FILE TAB

Parameter	Description
Format file	selection of the report file configured (on page 9) in the editor (*.xrs)
Show this dialog in the Runtime	`Active:` This filter dialog is offered before being called up in Runtime.
Reading archives from the read-back folder	`Active:` Stored archives are read in from the read-back folder defined in project configuration. When loading archive data from the readback folder, the archive data from the Runtime path and from all subfolders of the readback folder is also read.
Export file	Properties of the file that is exported in the report.
Name	Name of the file. Maximum of eight alphanumeric characters. If no name is given and automatic naming is switched off, the name `dummy` is used. **Attention:** The use of special characters may lead to a loss of data.
Format	Format of the export file.
▸ TXT	Text file
▸ XRS	zenon Report file
▸ XML	XML-File

Generate name automatically	`Active`: The filename is created automatically. It consists of the short identifier and the coded date and the chosen file ending.

CODING NAME FOR AUTOMATIC NAMING

Code	Meaning
`xx`	Short identifier: 2 characters The first two characters of the name given under 'Name' or the name issued by the system, `du`.
`YY`	Year: two-digits
`MM`	Month: two-digits
`DD`	Day: two-digits
`HH`	Hour: two-digits
`MM`	Minute: two-digits
`SS`	Second: two-digits

EXAMPLE

▶ The `Report` report file is exported on 9. 6. 6. 2010 at 11:52 in XRS format

▶ Automatic filename: `re100506091152.xrs`

> 💡 **Info**
>
> *Archive entries can only be read with the `Export report` function. Write actions are not possible.*

Execute report

The `execute report` function makes it possible to execute a report automatically in the background in Runtime. If writing functions are configured in the report, calculated values can automatically be written to archives or to the process image.

To configure the function:
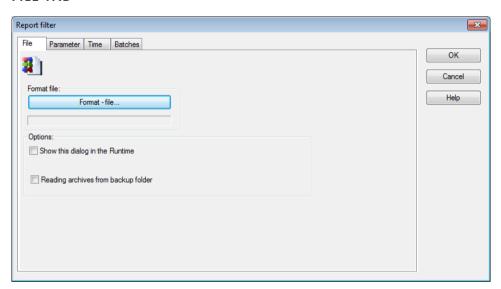
1. create a new function

2. Select, in the `Report generator` section, the `Execute report` entry

3. The dialog for the report filter (on page 159) is opened

Note: The `File` tab settings are different. The settings that are valid for `Execute report` are clarified in this chapter; you can find all further settings in the report filter (on page 159) chapter.

4. select the desired properties

5. Link the functions with a button in the screen, in order to switch in Runtime

FILE TAB

Parameters	Description
`Format file`	selection of the report file configured (on page 9) in the editor (*.xrs)
`Show this dialog in the Runtime`	`Active:` This filter dialog is offered before being called up in Runtime.
`With dialog cancel`	`Active:` A button to cancel printing is shown whilst the report is being prepared for print.
`Reading archives from the read-back folder`	`Active:` Stored archives are read in from the read-back folder defined in project configuration. When loading archive data from the readback folder, the archive data from the Runtime path and from all subfolders of the readback folder is also read.

Info
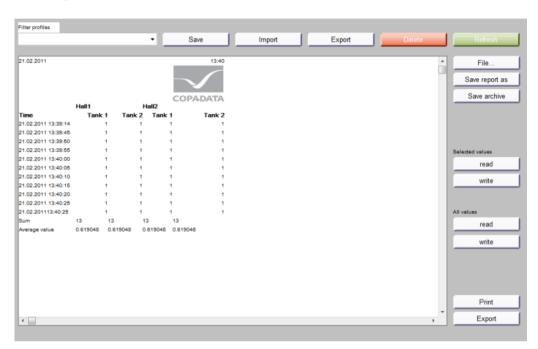
With this function, all write actions are carried out in the report.

2.1.2   Operating during Runtime

The reports on the configured buttons are displayed, configured and administered in Runtime. The following user elements are available in Runtime:

Parameter	Description
Insert template...	Opens the dialog for selecting a template for the screen type.
	Templates are shipped together with zenon and can also be created by the user.
	Templates add pre-defined control elements to pre-defined locations in the screen. Elements that are not necessary can also be removed individually once they have been created. Additional elements are selected from the drop-down list and dragged onto the screen. Elements can be moved on the screen and arranged individually.
Report	Report window to display the report.
Display: Set filter	Displays the the currently-configured time filter in Runtime.
Buttons	Buttons for controlling in the Runtime.
Refresh	Recreate build report completely (values and display).

Parameter	Description
Print	Print report in Runtime.
File	Select report file and change filter conditions.
Export	Export report.
Save archive	Write changed values to the archive.
Save report	Report is saved in Runtime (*.xrs).
	Attention: When saving, formulas and functions are replaced by the current values. The functions in these reports (.xrs files) are no longer available. These reports can also not be edited in the editor anymore.
	Recommendation: Set original report files to "write-protected" status and save changes with **Save as** in a new file.
Save report as	Report is saved under a new name during Runtime (*.xrs), formulas and functions are replaced by current values.
Read all values	All values are read in again from the driver.
Read selected values	Selected values are read in again from the driver.
Write all values	All values are sent to the driver.
Write selected values	Selected values are sent to the driver.
Line up	Scroll text in list element up
Line down	Scroll text in list element down
Column right	Scroll text in list element to the right
Column left	Scroll text in list element to the left
Page up	Scroll text in list element up
Page down	Scroll text in list element down
Page right	Scroll text in list element to the right

Parameter	Description
`Page left`	Scroll text in list element to the left
Filter profiles	Buttons for filter profiles.
`Profile selection`	Display of the available filter profile and input for new profile names.
`Save`	Saves current setting as a profile.
`X (save)`	Deletes selected profile.
`Import`	Exports profiles.
`Export`	Imports profiles.

> ## 💡 Info
>
> *A period is used as a decimal separator. When inputting, both a period or a comma can be used. The separator is automatically converted by the system.*
>
> *Example: `10,5` is turned into `10.5`*

Filter for screen switch

If you switch to a report type screen, the following filter is displayed.

Parameter	Description
Lot filter	On the left side you can choose the desired archive from the available archives. On the right side the available lots are displayed. You can filter the lots there.
Lot name	In this column the names of the available lots are displayed. By left clicking the top part of the header, the lots are sorted alphabetically in an ascending or descending order. In the bottom part of the header you can enter a character string. Only lots matching the respective character string will be displayed.
Start date	In this column the start date of the available lots is displayed. By left clicking the top part of the header, the lots are sorted in an ascending or descending order. Lots with the same start date are sorted according to their start time. In the bottom part of the header you can enter a start date manually or use the displayed calendar.
Start time	Only available if you entered a start date. In this column the start time of the available lots is displayed. By left clicking the top part of the header, the lots are sorted in an ascending or descending order. In the bottom part of the header you can enter the start time manually. Note: '*' means 12:00:00 AM o' clock.
End date	In this column the end date of the available lots is displayed. By left clicking the top part of the header, the lots are sorted in an ascending or descending order. Lots with the same end date are sorted according to their end time. In the bottom part of the header you can enter an end date manually or use the displayed calendar.
End time	Only available if you entered an end date. In this column the end time of the available lots is displayed. By left clicking the top part of the header, the lots are sorted in an ascending or descending order. In the bottom part of the header you can enter the end time manually. Note: '*' means 11:59:59 PM o' clock.
Duration	This column displays the duration for each available lot. It is only for display.

> **💡 Info**
>
> *Still open lots are also displayed if they match the set filter criteria.*

> **💡 Info**
>
> *The value of the lot variable is written in the index file and in the header of the ARX file at the start of the lot. These entries are adjusted with every change of the variable. When the lot is closed, the value of the lot variable at this moment is finally written in the index file and in the header.*
>
> *Thus the lot name is final when the lot is closed.*

2.1.3   Example of archive data in reports

In this example, you create archive data in a report. To do this:

1. Open a new report using the context menu and the `New report` entry.

2. Design the Title area (on page 183)

3. Fill the data area (on page 188)

4. Set up a report screen (on page 194) and create a function

Title area

The title area contains, in the first four rows:

▶ Texts (on page 184) as headings for the data

▶ Date (on page 184) and time

▶ Company logo (on page 185)
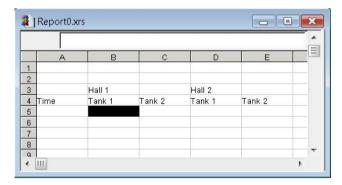
The title area is formatted in such a way that it is fixed for longer reports and does not disappear from view during scrolling.

Texts

Enter the following texts as headings for the report:

▶ Time

▶ Hall 1

  • Tank 1

  • Tank 2

▶ Hall 2

  • Tank 1

  • Tank 2
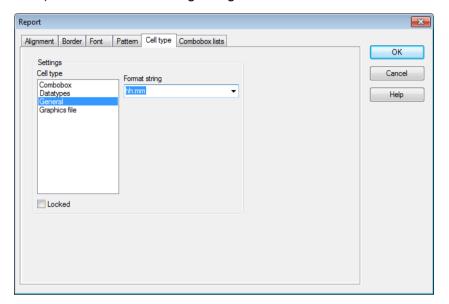
Date and time

The date and time of the report are configured for the for cells A1 and E1 with report functions. To do this:
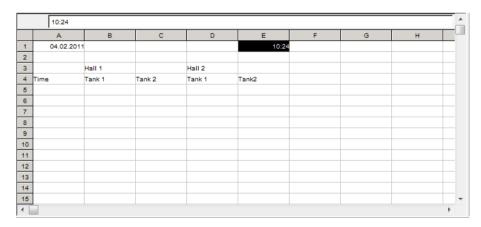
1. Enter the function =now() into the cells A1 and E1.

2. The current date is now displayed in both cells.

3. In order to get the time in cell E1 instead, you have to change the formatting of this cell.

   a) Highlight cell E1

b) Open the format dialog using the Format/Cells/Editing menu.

c) Select the formatting string as shown in the illustration below

4. The result:
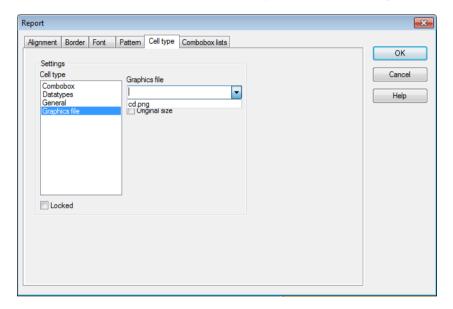
Screens as a company logo

Screens in cells are not treated as cell contents by the Report Generator, but as formatting. Configuration is therefore carried out using cell formatting:
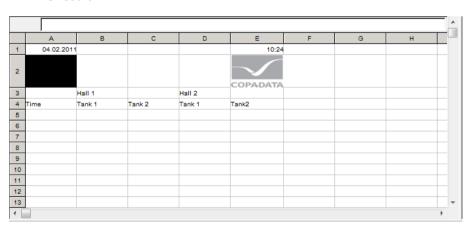
1. Import the desired screen in the Project Manager into the **Files/graphics** nodes

2. Select cell E2 with the left mouse button.

3. Open the context menu of the report with the right mouse button and select the `Cell` command.

4. In the dialog box, change to the `Cell type` page.

5. Change the cell type to `Graphics file`

6. Select the desired screen from the drop-down list on the right side

7. The result:
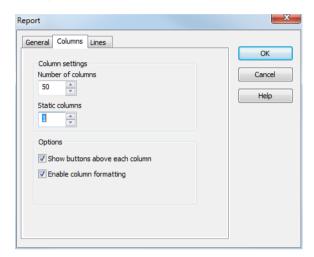
Fixed title area for multipage reports

To always have the row and column titles in view in multi-page reports, you must prevent these also being moved when the report is scrolled. To do this, define fixed title areas.
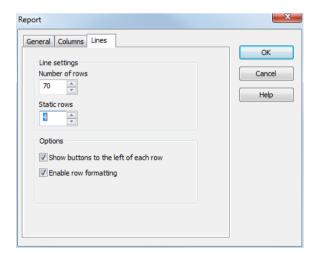
To fix the row title:

1. Select the `Report` command in the Format menu.

2. The configuration dialog is opened

3. Open the `Columns` tab

Next to the number of columns you also can define the number of `Non-scrolling columns` here. The stipulated number of fixed columns is then no longer scrolled out of the visible screen area.

4. Enter 1 for the `Non-scrolling columns`.

5. Change to the `Rows:` tab

Our column headings are in rows 1 to 4 for the lines:

6. Enter 4 for the `Non-scrolling rows`.

7. Confirm all changes by clicking on the **OK** button

Data area

In the data area:

▶   Configure the display of time (on page 188) and values (on page 192)

▶   Have the sum and average (on page 193) displayed

Time

The time of the displayed archive values is displayed in column A.

> 💡 **Info**
>
> *We select a cyclical archive as a basis. The entries of all variables are therefore made at the same time and a time column is sufficient for all entries.*
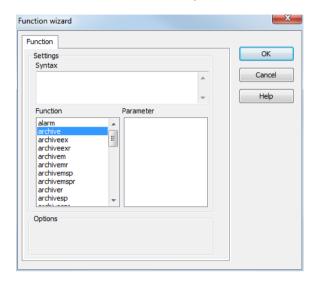
To configure the time:
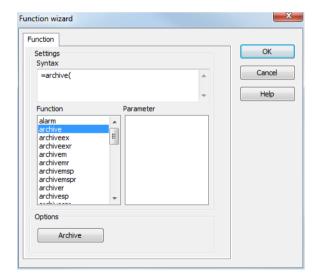
1. Select cell A5.

2. Select the **Function assistant** command in the **Format** menu

3. The function assistant is opened
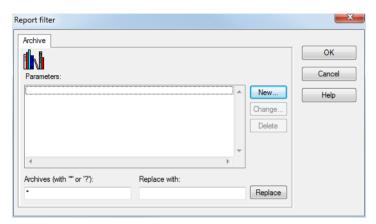
4. Select the **Archive** function with a double-click.

5. The **Archive** button is always displayed in the lower area of the dialog
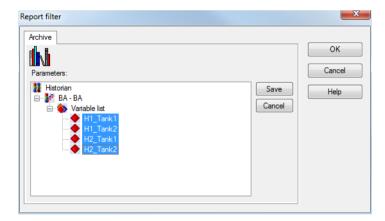
6. Click on **Archive**

7. The dialog for archive selection is opened:
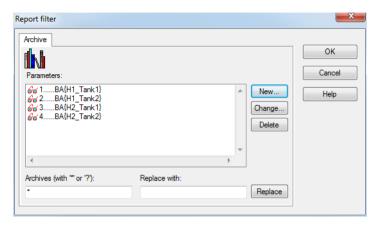
8. Click on **New**

9. The archives stored in the archive server node are displayed

10. Select the variables as shown in the screenshot
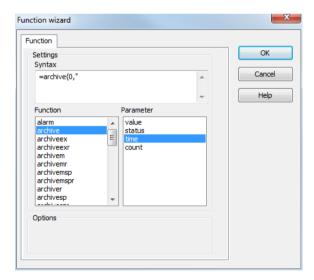
11. Click on Save

12. Select the variable "H1_Tank1"
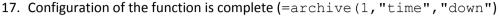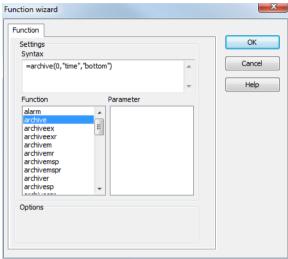
13. Confirm the selection by clicking on OK

14. The function assistant is displayed again

15. Select `time` as the `parameter` with a double-click.

16. Select the `parameter down` with a double-click.

17. Configuration of the function is complete (=`archive(1,"time","down")`

18. Close the assistant by clicking on the oκ button

The time of variable archiving is thus continued downwards from row 5 in column A.

💡 Info

The parameters always have to be selected with a double-click. All parameters have been set up once the function in the preview is closed with a bracket and no further parameters are offered. If the oκ button is clicked on before all parameters have been selected, the assistant is closed before all parameters for the function have been set up.

Values

You configure the values for the tanks in hall 1 and 2 in the same manner as the configuration for the time (on page 188). Pay attention to the following differences:

▸ The archive variables no longer need to be set up. They already exist and can be selected immediately. Assign each tank the corresponding variable.

▸ Instead of the `time` parameter, select the `value` parameter.
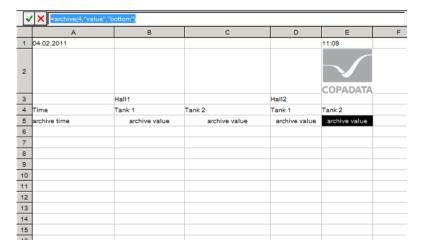
COPYING A FUNCTION

As soon as you have created a function, you can copy it to other cells and then edit it accordingly:

To do this:

1. Highlight the cell to be changed

2. Then click in the editing cell

3. In the function that is now displayed, the reference numbers refer to the corresponding archive variables as they are automatically issued during archive selection when the function is first created

4. Therefore replace the `1` as the first parameter in column `C` by a `2`
   in column `D` with a `3`
   and in column `E` with a `4`

Sum and average

For this report, you now configure the display of sum and average.

To do this, you must determine how many cells are needed for archive entries. You define these in the Report Generator. The number of entries in the list depends on the filter settings that are used to subsequently call up the report (**Execute report, Print report, Export report** functions).

In our example, we will select a time range of two minutes. As our archive records values every five seconds, this will make 24 entries for the report. Therefore:

1. Enter the text **sum** into cell A29.

2. Enter the text **Average:** into cell A30 on

3. Enter the function **=sum(B5:B25)** into cell B29.

4. Enter the function **=sum(B5:B25)** into cell B30.

5. Copy the cell B29 to C29, D29 and E29

6. Copy the cell B30 to C30, D30 and E30

7. Save the report under the name **Archiv.xrs**

> ### 💡 Info
>
> *When entering the functions, you can either type in cell names or click on the corresponding cell with the mouse.*
>
> *If you copy functions with cell addresses in the Report Generator, the cell addresses are automatically adapted.*

Displaying the report in Runtime

To be able to look at the report in Runtime, you need:

▶ A report screen

▶ A screen switching function

CREATE SCREEN OF TYPE REPORT

In order to create a Report screen:

1. Create a new screen named **Report**.

2. Select Report as screen type

3. Select the frame MAIN

4. Click in the new screen

5. Select the "Default" command from the "Control elements" menu

6. Add any other required control elements there may be
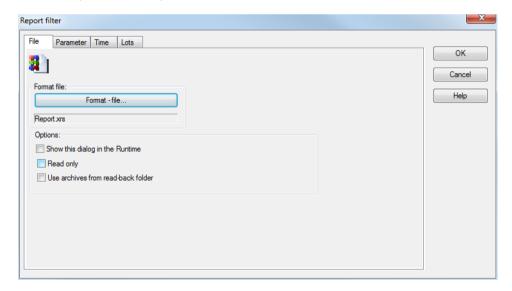
CREATE FUNCTION

Create a screen switch function to select the screen in Runtime. To do this:

1. Add a new button to the start screen

2. Label it with `Report`

3. Change to the `Functions` node

 create a new function

4. select `Screen switching`

5. Select the `Report` screen

6. The report filter is opened

7. Click on the `Format file` button

8. Select `Archiv.xrs`

9. Open the `Time` tab
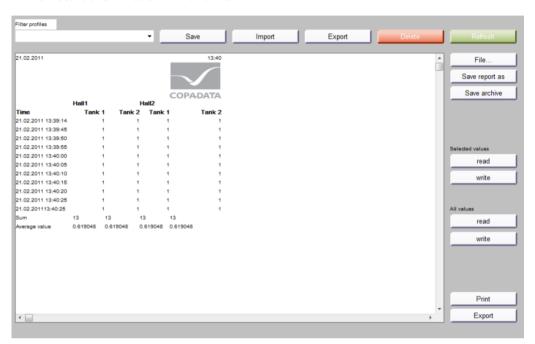
10. Set, for our example

    a) The time format to `Relative time period` and

    b) Set the time to `2 minutes`.

You can read more about the time filter in the Alarm Message List manual in the time filter chapter

11. Close the filter by clicking on OK

12. Allocate a the configured button to the function

13. Start the function in Runtime

2.2 Report Viewer

The Report Viewer can be used to display RDL reports of archive data, AML data, and CEL data as well as online values. Two historical time ranges can also be compared using two configurable time ranges.

RDL files that display the report template for Runtime are created and edited in the Editor. There are already 7 pre-defined datasets available. Further datasets can be created (on page 258) freely; only the table scheme (on page 271) for the four data categories is stipulated.

The source data that supplies the datasets created in the RDL file in Runtime is defined in the screen switching function (on page 201) on the Report Viewer (on page 199) screen. For each dataset used in the RDL file, a dataset with the same name and data origin must be created. The source variables and filter criteria can be prescribed for these depending on their type.

TYPICAL CONFIGURATION STEPS:

1. Create a new report definition file (on page 254) or import one

2. Create Report Viewer (on page 199) screen

3. Create screen switching function (on page 201) and configure datasets (on page 214)

4. Configure output (on page 247) as a PDF or output to the printer

5. Configuring (on page 258) an RDL file

6. Call up the function in Runtime using a button or menu

2.2.1 Engineering in the Editor

To be able to use the Report Viewer in Runtime, the following must be carried out in the Editor:

1. Create a Report Viewer (on page 199) screen

2. Create or import an RDL file, if none is present

3. Set up a screen switching function (on page 201) on the Report Viewer screen

4. Configure the datasets in the screen switching

5. Amend the report definition file (on page 254) (RDL) to suit the datasets (on page 271) defined in zenon.

6. Add a button to call up the Report Viewer in Runtime in the start screen or the menu screen

FOLDER FOR REPORT DEFINITION FILES (RDL)

RDL files are stored in the project tree in the Files/Report Viewer node. Existing RDL files can be imported and edited here or new files can be created.

REPORT TEMPLATE CONTEXT MENU

| Menu item	Action
`New report definition file`	Creates a new RDL file on the basis of the supplied `default.rdl` (on page 257).
`Open report definition file`	Opens the software linked to the RFL files to edit an existing definition file.
	zenon first looks for MS Report Builder, then for MS Report Designer (on page 258) by default.
`Jump back to starting element`	With linked elements, jumps back to the element from which the report template was jumped to.
	Only present in the context menu if linked elements are present.
`Copy`	Copies selected entries to the clipboard.
`Paste`	Pastes the contents of the clipboard. If an entry with the same name already exists, the content is pasted as "Copy of".
`Delete`	Deletes selected entries.
`Import existing report definition file`	Opens dialog to import existing report definition files. These must be of RDL 2.0 type.
`Remove all filters`	Removes all filter settings.
`Edit selected cell`	Opens the selected cell for editing. The binocular symbol in the header shows which cell has been selected in a highlighted line.
`Replace text in selected column`	Opens the dialog for searching and replacing texts.
`Properties`	Opens the **Properties** window for the selected entry.
`Help`	Opens the online help.
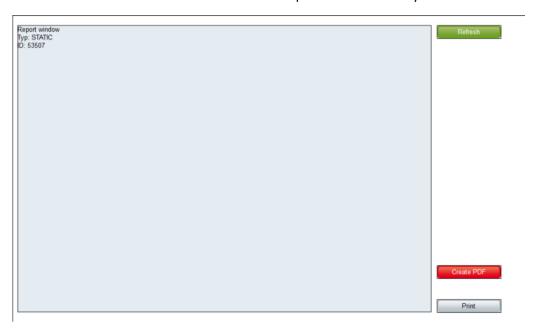
Create Report Viewer screen

In order to create a Report Viewer screen:

1. Create a new screen

2. Select the `Report Viewer` special screen type in the drop-down list

3. Name the screen

4. select **Add template** in the control element menu

5. The standard elements are created and placed automatically

Report window
Typ: STATIC
ID: 53507

Refresh

Create PDF

Print

Parameter	Description
`Add template`	Opens the dialog for selecting a template for the screen type. Templates are shipped together with zenon and can also be created by the user. Templates add pre-defined control elements to pre-defined locations in the screen. Elements that are not necessary can also be removed individually once they have been created. Additional elements are selected from the drop-down list and dragged onto the screen. Elements can be moved on the screen and arranged individually.
`Report window`	Functions and profiles for the report.
`Display: Set filter`	Displays the currently-configured time filter in Runtime.
`Update`	Reloads the report definition and the data and displays the report with the updated data.
`Print`	Prints the report on the printer defined for values and logs.
`Create PDF`	Saves the report currently being displayed as a PDF file in the folder defined for export.

Screen switching to a Report Viewer screen

To use the Report Viewer in Runtime, configure screen switching to a Report Viewer screen type:
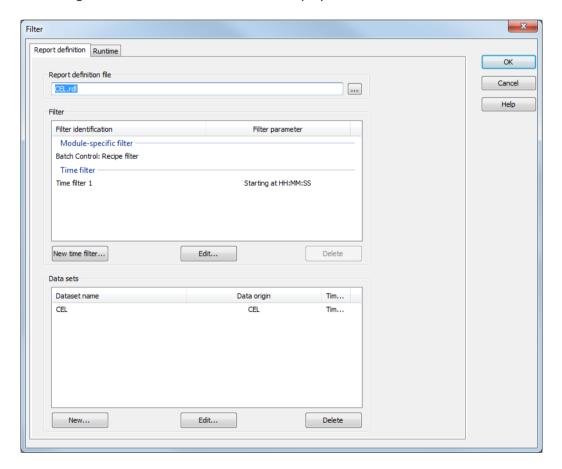
1. Create a new `Report Viewer` screen

2. Select the `New function` command in the `Functions` node

3. select the `Screen switching` function

4. Select the Report Viewer screen

5. The report filter (on page 214) is displayed

6. Configure, if you want, the replacement of the time filter in the Runtime (on page 204) tab

7. Switch to the `Report definition` (on page 202) tab

8. Select the RDL file

9. Define the module filter (on page 205) and/or time filter (on page 210)

10. Create the desired datasets (on page 214)

11. Link the functions with a button in the screen, in order to switch in Runtime

12. Edit the datasets in the RDL file (on page 254) according to their settings in screen switching

Report definition

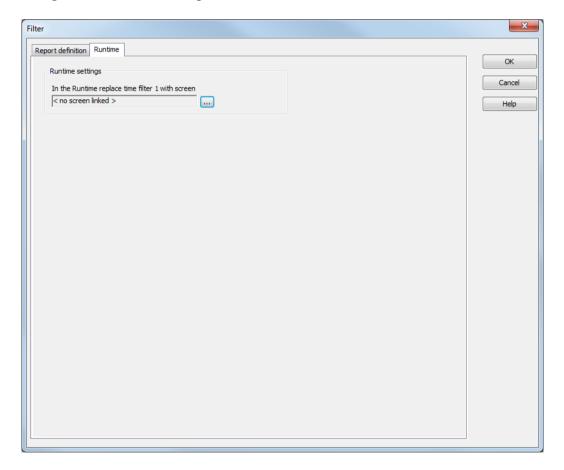You configure the data sets and filters to be displayed in this tab.

Parameter	Description
`Report definition file`	RDL file that is used for report definition. This file must be present in the `File/report templates` node. A click on the `...` button opens the file selection dialog.
Filter	List of the available filters with a display of names and filter type. The following is available: ▶ **Module-specific filter**: Batch Control recipe filter (on page 205) ▶ Time filter Configuration: ▸ Clicking on the **Module-specific filter** opens the dialog to configure a batch filter (on page 205). ▸ Clicking on the **New time filter ...** button opens the dialog to configure a time filter (on page 210). Double-clicking on the group overview opens the first configured filter of the group.
New time filter	Opens the dialog (on page 210) for configuring a new time filter.
Edit	Opens the list of the time filters highlighted in the list for editing.
Delete	Deletes the time filter highlighted in the list.
Data sets	Display of configured datasets (on page 214). The list contains: ▸ Dataset name ▸ Origin of the data ▸ Time filter
New	Opens the dialog (on page 211) to create a new report definition.
Edit	Opens the dialog (on page 214) to edit the definition of the selected element.
Delete	Deletes selected element and its definition from the list.

OK	Accepts changes in all tabs and closes dialog.
Cancel	Discards all changes and closes the dialog.
Help	Opens online help.

Runtime

In this tab, you configure the replacement of the first time filter by a `time filter` type screen. With this, you can, when switching a separate screen for example, use time filters instead of the filter configured in screen switching.

| Parameter	Description
`Runtime settings`	Settings for actions in Runtime.
`Replacing time filter 1 with a screen in Runtime`	Replaces the first-configured time filter with a `time filter` type screen in Runtime. The currently-linked screen is displayed. Click the `...` button and the dialog opens to select a screen. Only screens from the project calling them up and `time filter` screens are offered. When selecting a filter screen, the **BOX SCREEN** information is also shown in the detail view of the function in the **Parameter** column; without a linked screen, only **BOX** is displayed.
`OK`	Accepts changes in all tabs and closes dialog.
`Cancel`	Discards all changes and closes the dialog.
`Help`	Opens the online help.
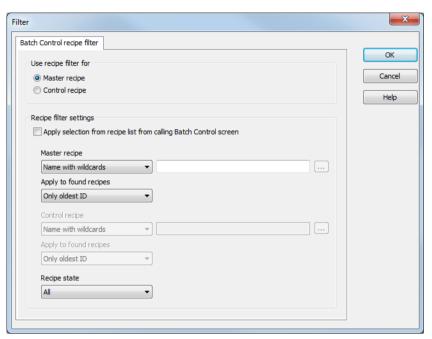
Batch Control recipe filter

When screen switching to a report viewer type screen, a filter can be set for recipes from the Batch Control module. To filter according to recipes:

1. Open the Report definition (on page 202) tab for screen switching

2. go to area **Filter**

3. In the **Module-specific filter** tab, click on `Batch Control: Recipe filter`

4. the dialog for configuring the filter is opened

Parameter	Description
Use recipe filter for	Selection of the recipe type that is applied to the filter: ▶ Master recipe ▶ Control recipe
Master recipe	`Active`: It is filtered on Master recipes.
Control recipe	`Active`: It is filtered on control recipes. Note: The attendant master recipes must also be selected. If no master recipe has been selected for the control recipe, th filer cannot find the recipe being searched for in Runtime. Hint: If the master recipe is not known, filtering of all master recipes with a placeholder is recommended.
Recipe filter settings	Options for the recipe filter
Apply selection from recipe list from calling Batch Control screen	`Active`: In Runtime, the first selected recipe of the batch screen from which the report viewer screen is called up is used. Individual settings in this dialog are then not available. `Inactive`: The filter settings are changed individually using this dialog.
Master recipe	Parameters for the selection of the master recipe. Select from drop-down list: ▶ Name with wildcards: A name with placeholder can be entered into the input field. Filtering according to this name is carried out. ▶ Name from variable: The name of the master recipe is defined by a variable in Runtime. A click on button `...` opens the dialog for selecting variables. ▶ ID from variable: The ID of the master recipe is defined by a variable in Runtime. A click on button `...` opens the dialog for selecting variables.
Filter result for	Define which ID is to be selected when filtering for names by selecting from the drop-down list:

	▸ Only oldest ID
	▸ Only newest ID
	Because a report can only be used for one recipe, it is not possible to filter for "all recipes".

| Control recipe	Parameters for the selection of the control recipe. Select from drop-down list:
	▸ Name with wildcards: A name with placeholder can be entered into the input field. Filtering according to this name is carried out.
	▸ Name from variable: The name of the control recipe is defined by a variable in Runtime. A click on button `...` opens the dialog for selecting variables.
	▸ ID from variable: The ID of the master recipe is defined by a variable in Runtime. A click on button `...` opens the dialog for selecting variables. Precisely one recipe can be found if the variable value at the time of execution is a valid ID of a control recipe.
	▸ Job ID from variable Finds control recipes that belong to the master recipes already found and which have the given job ID. Any type of variable can be linked. The value is automatically converted into STRING. Note: If the variable does not have a value, no recipe is sent to the Report Viewer.
Filter result for	Define which ID is to be selected when filtering for names or job ID by selecting from the drop-down list:
	▸ Only oldest ID
	▸ Only newest ID
Recipe state	Selection of the recipe status from the drop-down list:
	▸ All
	▸ Prepared
	▸ In execution
	▸ Executed
	▸ Terminated with error
	▸ Outdated
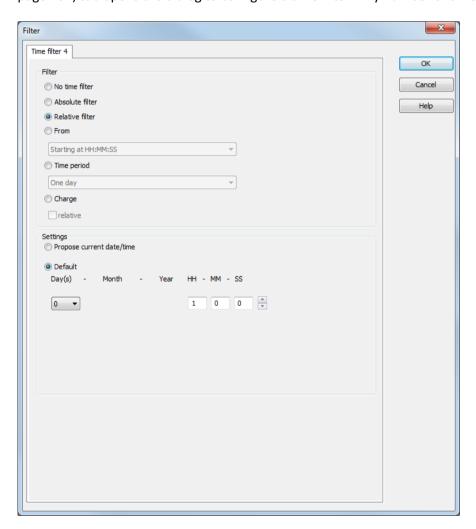
OK	Applies all changes, creates filter and closes the dialog.
Cancel	Discards all changes and closes the dialog.
Help	Opens the online help.

Note for variable selection using name or ID: For the selection of variables according to name or ID, numerical variables and string variables can be selected respectively. The data types are converted to the respective correct form.

Time filter

Clicking on the `New` button or `Edit` button in the `Time filter` section of the `Report definition` (on page 202) tab opens the dialog to configure a time filter. Any number of time filters can be used.

Time filtering can be carried out in two ways:

1. Define time period in the Editor

   Fixed time areas are used. A time period is given in the editor. It is only possible to filter according to this time period in Runtime. Other filters - such as filtering according to variable name, alarm/event groups and alarm/event classes etc. can no longer be amended in Runtime.

2. Time filter amendable in Runtime

   Pre-defined times are used. The time filter is defined in the Editor and can be changed in Runtime as desired.

You can read details on configuration in the Alarm administration manual in the Time chapter.

Note: If the `no time filter` option is set as time filter type, all Runtime entries since 1. 1. 2000 are displayed.

> ⚠ **Attention**
>
> *A Report Viewer type screen can have several time filters. If, in the Runtime (on page 204) tab, a filter is set to a Report Viewer type screen using a* `time filter` *screen, then only the first time filter is replaced with the settings of the filter screen. Other filter types of the time filter are automatically set to* `no time filter`*.*
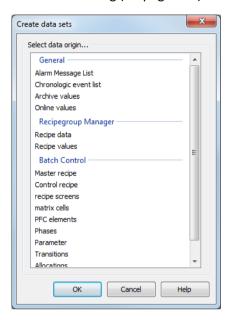
New dataset dialog

To define a new data set:

1. Define, in the screen switching (on page 202), the RDL file on which it is based on a `Report Viewer` screen

2. Click, in the **Datasets** section, on the **New** button

3. The dialog to create the datasets is opened

4. Select the data origin you want to assign

5. Confirm the selection by clicking **OK**

6. the dialog (on page 214) for configuring the defined type is opened

Create data sets

Select data origin...

General
Alarm Message List
Chronologic event list
Archive values
Online values

Recipegroup Manager
Recipe data
Recipe values

Batch Control
Master recipe
Control recipe
recipe screens
matrix cells
PFC elements
Phases
Parameter
Transitions
Allocations

OK   Cancel   Help

Parameter	Description

`Origin of the data`	Selection of the data origin:
	
	General
	
	  ▶ Alarm Message List (on page 215)
	
	  ▶ Chronological Event List (on page 217)
	
	  ▶ Archive values (on page 219)
	
	  ▶ Online values (on page 221)
	
	Recipegroup Manager
	
	  ▶ RGM recipe data (on page 223)
	
	  ▶ RGM recipe value (on page 230)
	
	Batch Control
	
	  ▶ Master recipe (on page 237)
	
	  ▶ Control recipe (on page 238)
	
	  ▶ Recipe screens (on page 238)
	
	  ▶ Matrix cells (on page 241)
	
	  ▶ PFC structure (on page 242)
	
	  ▶ Phases (on page 243)
	
	  ▶ Parameter (on page 244)
	
	  ▶ Transitions (on page 245)
	
	  ▶ Unit allocations (on page 246)
	
	  ▶ Operation instance (on page 247)

OK	Confirms selection and opens dialog (on page 214) for configuration of the type.
Cancel	Closes dialog without further configuration.
Help	Opens online help.

Datasets in the Report Viewer
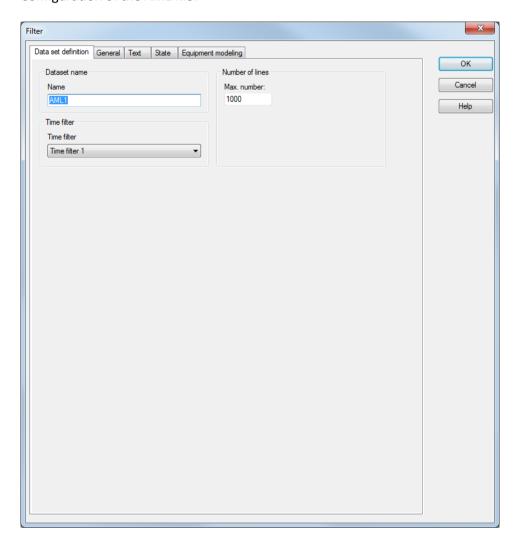
The settings for the different datasets depend on the type of data origin to be configured:

- ▶ AML data (on page 215)

- ▶ CEL data (on page 217)

- ▶ Archive data (on page 219)

- ▶ Online values (on page 221)

- ▶ RGM

  - • RGM recipe data (on page 223)
  - • RGM recipe value (on page 230)

- ▶ Batch Control (on page 236)
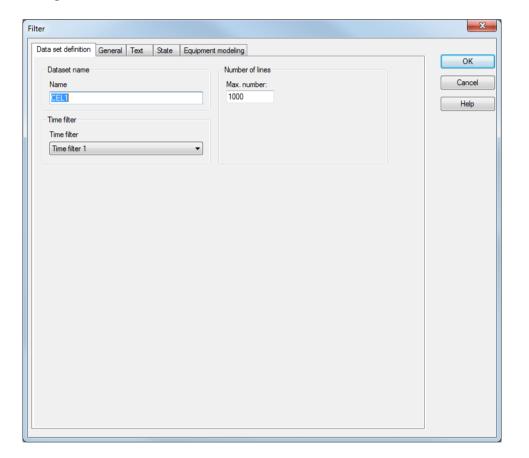
AML data

Configuration of the AML file:

Parameter	Description
`Dataset name`	Name of the data set: ▸ must not be empty ▸ Must correspond exactly to the name of the data set in the RDL file (on page 254) ▸ Must not contain any spaces or special characters
`Max. number:`	Maximum number of lines for the report. If the report has more lines than defined here due to its filter criteria, the first **n** lines are output. **n** corresponds to the value entered. Allowed values: `1 – 2000` Default: `1000`
`Time filter`	Selection of the time filter from the drop-down list. Contains the time filter defined in the Report definition (on page 202) tab.
`General` tab	Settings for: ▸ Variable filter ▸ Type of alarms ▸ Origin of the data ▸ Alarm/event groups, alarm/event classes and alarm areas For details see the Alarm administration manual, General chapter.
`Text` tab	Settings for the text filter. For details see the Alarm administration manual, Text chapter.
`Status` tab	Data for evaluation of status bits. For details see the Alarm administration manual, Status chapter.
`Equipment modeling` tab	Selection of the equipment model. For details see the Equipment modeling manual, Equipment modeling dialog chapter.

OK	Applies settings and closes the dialog.
Cancel	Discards changes and closes dialog.
Help	Opens online help.

CEL data

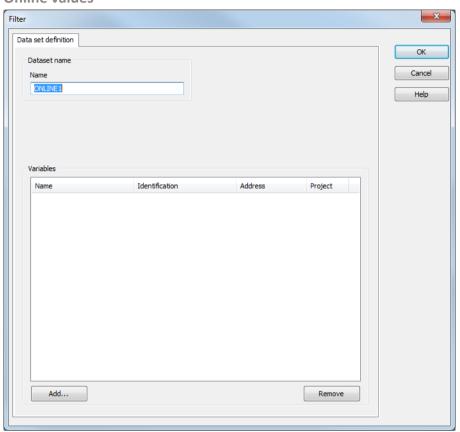Configuration of the CEL files:
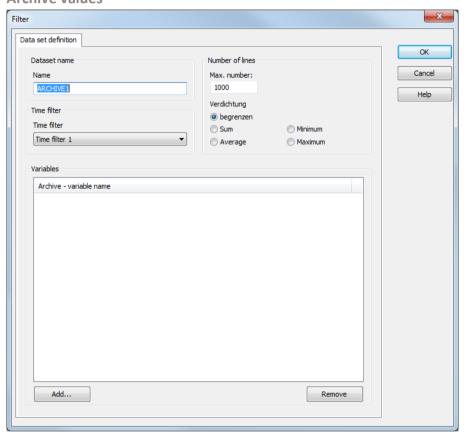
Parameter	Description
Dataset name	Name of the data set: ▸ must not be empty ▸ Must correspond exactly to the name of the data set in the RDL file (on page 254) ▸ Must not contain any spaces or special characters
Max. number:	Maximum number of lines for the report. If the report has more lines than defined here due to its filter criteria, the first **n** lines are output. **n** corresponds to the value entered. Allowed values: 1 – 2000 Default: 1000
Time filter	Selection of the time filter from the drop-down list. Contains the time filter defined in the Report definition (on page 202) tab.
General tab	Settings for: ▸ Variable filter ▸ Origin of the data ▸ Alarm/event groups, alarm/event classes and alarm areas For details see the Chronological Event List manual, General chapter.
Text tab	Settings for the text filter. For details see the Alarm administration manual, Text chapter.
Status tab	Data for evaluation of status bits. For details see the Alarm administration manual, Status chapter.
Equipment modeling tab	Selection of the equipment model. For details see the Equipment modeling manual, Equipment modeling dialog chapter.
OK	Applies settings and closes the dialog.

| Cancel	Discards changes and closes dialog.
Help	Opens online help.

Archive values

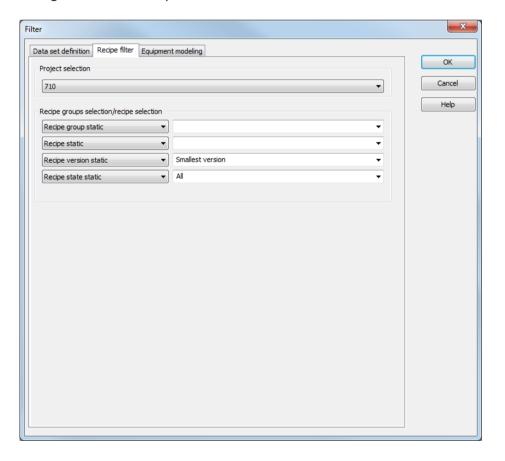| Parameter	Description
Dataset name	Name of the data set: ▶ must not be empty ▶ Must correspond exactly to the name of the data set in the RDL file (on page 254) ▶ Must not contain any spaces or special characters
Max. number:	Maximum number of lines for the report. If the report has more lines than defined here due to its filter criteria, the first **n** lines are output. **n** corresponds to the value entered. Allowed values: $1 - 2000$ Default: 1000
Compaction	type of aggregation: ▶ limit Outputs the first n values per archive if the filter criteria would provide more entries than this number. ▶ Sum: Time range is divided into **n** equidistant time intervals and the archive values in these are added up. Status values are ORed. ▶ Average: Time range is divided into **n** equidistant time intervals and an average of the archive values in these is calculated. Status values are ORed. ▶ Minimum: Time range is divided into **n** equidistant time intervals and the archive values in these are minimized. ▶ Maximum: Time range is divided into **n** equidistant time intervals and the archive values in these are maximized. The time stamp corresponds to the respective start time of the interval.
Time filter	Selection of the time filter from the drop-down list. Contains the time filter defined in the Report definition (on page 202) tab.
Archive - variable names	List of selected variables.
Add	Opens the dialog for selecting several variables.
Remove	Deletes selected variable from the list.
OK	Applies settings and closes the dialog.
Cancel	Discards changes and closes dialog.
Help	Opens online help.

Online values
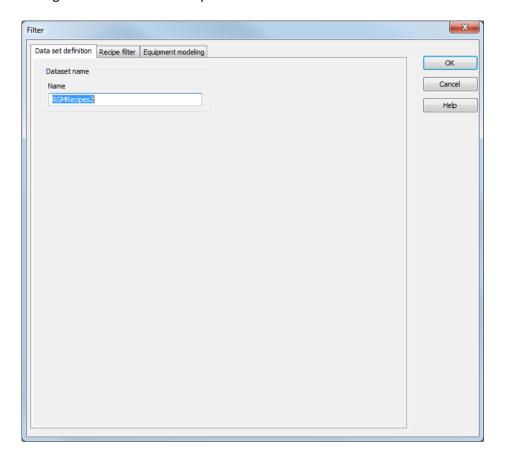
Parameter	Description
Dataset name	Name of the data set: ▶ must not be empty ▶ Must correspond exactly to the name of the data set in the RDL file (on page 254) ▶ Must not contain any spaces or special characters
Variables	List of selected variables with: ▶ Name ▶ Identification ▶ Address ▶ Project
Add	Opens the dialog for selecting several variables.
Remove	Deletes selected variable from the list.
OK	Applies settings and closes the dialog.
Cancel	Discards changes and closes dialog.
Help	Opens online help.

RGM recipe data

Configuration of the RGM recipe data:

Parameter	Description
Dataset name	Name of the data set: ▸ must not be empty ▸ Must correspond exactly to the name of the data set in the RDL file (on page 254) ▸ Must not contain any spaces or special characters
OK	Applies settings and closes the dialog.
Cancel	Discards changes and closes dialog.
Help	Opens online help.

RECIPE FILTER

Configuration of the recipe filter

Parameter	Description
`Project selection`	Selection of the project for the recipe data from the drop-down list.
`Recipegroup`	Selection of the recipe group. The selection can take place: ▸ Static from pre-defined entries (`Static recipe group`) ▸ Dynamic via variable (`Recipegroup name from variable`) Clicking on Property opens a drop-down list to select the method. Selection of the group: ▸ from drop-down list ▸ Input of a filter text with wildcards (**?** for exactly one character and ***** for 0 to as many characters as desired) ▸ Via a string variable: The text value of the variable is then interpreted as a group name or filter text with wildcards
`Recipe`	Selection of the recipe from the selected recipegroup. The selection can take place: ▸ Static from pre-defined entries (`Static recipe`) ▸ Dynamic via variable (`Recipe name from variable`) Clicking on Property opens a drop-down list to select the method. Selection of the name: ▸ from drop-down list ▸ Input of a filter text with wildcards (**?** for exactly one character and ***** for 0 to as many characters as desired) ▸ Via a string variable: The text value of the variable is then interpreted as a recipe name or filter text with wildcards
`Recipe version static`	Selection of the recipe version from the selected recipegroup. The selection can take place: ▸ Static from pre-defined entries (`Static recipe version`)

	▸ Dynamic via variable (`Recipe version from variable`)
	
	Clicking on Property opens a drop-down list to select the method.
	
	Select version:
	
	▸ Entry from the drop-down list:
	  - `Largest version`
	  - `Smallest version`
	  - `All`
	  - recipe versions (only available if the group and recipe selection results in an individual recipe; selection without wildcards)
	
	▸ Entering of a list of recipe versions, with the versions separated by commas (`,`), for example "`1, 3, 5`"
	
	▸ A string variable that provides a list of version numbers as a text separated by a comma (`,`)
	
	▸ A numerical variable with a version number (`1 - 89999`) or the values:
	  - `90000` for "`Smallest version`"
	  - `90001` for "`Largest version`"
	  - `90002` for "`all versions`"

`Recipe state static`	Selection of the recipe status from the selected recipegroup. The selection can take place: ▸ Static from pre-defined entries (`Static recipe status`) ▸ Dynamic via variable (`Recipe status from variable`) Clicking on Property opens a drop-down list to select the method. Selection of the status: ▸ Entry from drop-down list: `- All` - individual recipe status ▸ List of recipe status separated by comma (**,**) for example "**1, 2, 7**" ▸ A string variable that provides a list of status numbers as a text separated by a comma (**,**) ▸ A numeric value with a status number (`1 - 4294967294; 0xFFFFFFFE`) or the value ▸ `0` for `All`
OK	Applies settings and closes the dialog.
Cancel	Discards changes and closes dialog.
Help	Opens online help.

Possible combinations of recipe version and recipe status:

Version filter	Status filter	Result
`Smallest version`	`All`	Only the smallest respective versions of all recipes are shown.
`Smallest version`	`1.2`	Only the recipe versions that have the status 1 or 2 have the smallest versions shown.
`Largest version`	`All`	Only the largest respective versions of all recipes are shown.
`Largest version`	`1.2`	Only the recipe versions that have the status 1 or 2 have the largest versions shown.
`All`	`All`	All recipe versions are displayed.
`All`	`1.2`	All recipe versions with the status 1 or 2 are displayed.
`2.3`	`All`	All recipe versions 2 and 3 are displayed regardless of status.
`2.3`	`1.2`	The respective recipe version 2 and 3 is displayed if these have either the status 1 or 2.

You can read more about the RGM in the Recipegroup Manager manual.

EQUIPMENT MODELING

Selection of an Equipment model.

Select the desired equipment model from the list or create a new equipment model.

Note: Equipment models are managed in the global project. A global project must be present here in order to be able to create a new equipment model.

RGM recipe value

Configuration of the RGM recipe values. The visibility variables are evaluated for the output of recipe values. Only recipe values with the status `visible` are displayed in the Report Viewer:
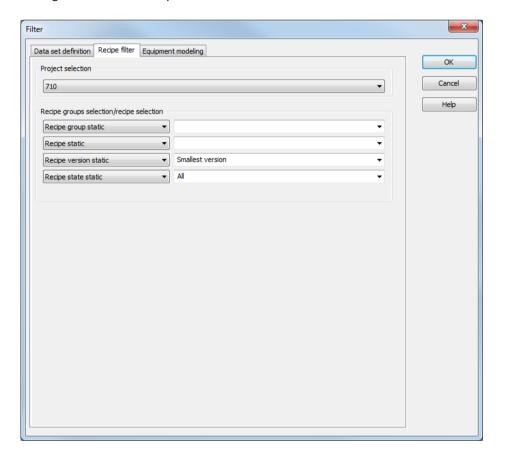
Parameter	Description
Dataset name	Name of the data set: ▸ must not be empty ▸ Must correspond exactly to the name of the data set in the RDL file (on page 254) ▸ Must not contain any spaces or special characters
OK	Applies settings and closes the dialog.
Cancel	Discards changes and closes dialog.
Help	Opens online help.

RECIPE FILTER

Configuration of the recipe filter

Parameter	Description
`Project selection`	Selection of the project for the recipe data from the drop-down list.
`Recipegroup`	Selection of the recipe group. The selection can take place: ‣ Static from pre-defined entries (`Static recipe group`) ‣ Dynamic via variable (`Recipegroup name from variable`) Clicking on Property opens a drop-down list to select the method. Selection of the group: ‣ from drop-down list ‣ Input of a filter text with wildcards (**?** for exactly one character and ***** for 0 to as many characters as desired) ‣ Via a string variable: The text value of the variable is then interpreted as a group name or filter text with wildcards
`Recipe`	Selection of the recipe from the selected recipegroup. The selection can take place: ‣ Static from pre-defined entries (`Static recipe`) ‣ Dynamic via variable (`Recipe name from variable`) Clicking on Property opens a drop-down list to select the method. Selection of the name: ‣ from drop-down list ‣ Input of a filter text with wildcards (**?** for exactly one character and ***** for 0 to as many characters as desired) ‣ Via a string variable: The text value of the variable is then interpreted as a recipe name or filter text with wildcards
`Recipe version static`	Selection of the recipe version from the selected recipegroup. The selection can take place: ‣ Static from pre-defined entries (`Static recipe version`)

	▸ Dynamic via variable (`Recipe version from variable`)
	
	Clicking on Property opens a drop-down list to select the method.
	
	Select version:
	
	▸ Entry from the drop-down list:
	  - `Largest version`
	  - `Smallest version`
	  - `All`
	  - recipe versions (only available if the group and recipe selection results in an individual recipe; selection without wildcards)
	
	▸ Entering of a list of recipe versions, with the versions separated by commas (`,`), for example "`1, 3, 5`"
	
	▸ A string variable that provides a list of version numbers as a text separated by a comma (`,`)
	
	▸ A numerical variable with a version number (`1 – 89999`) or the values:
	  - `90000` for "`Smallest version`"
	  - `90001` for "`Largest version`"
	  - `90002` for "`all versions`"

`Recipe state static`	Selection of the recipe status from the selected recipegroup. The selection can take place: ▸ Static from pre-defined entries (`Static recipe status`) ▸ Dynamic via variable (`Recipe status from variable`) Clicking on Property opens a drop-down list to select the method. Selection of the status: ▸ Entry from drop-down list: `- All` - individual recipe status ▸ List of recipe status separated by comma (**,**) for example "**1, 2, 7**" ▸ A string variable that provides a list of status numbers as a text separated by a comma (**,**) ▸ A numeric value with a status number (`1 - 4294967294; 0xFFFFFFFE`) or the value ▸ `0` for `All`
OK	Applies settings and closes the dialog.
Cancel	Discards changes and closes dialog.
Help	Opens online help.

Possible combinations of recipe version and recipe status:

Version filter	Status filter	Result
Smallest version	All	Only the smallest respective versions of all recipes are shown.
Smallest version	1.2	Only the recipe versions that have the status 1 or 2 have the smallest versions shown.
Largest version	All	Only the largest respective versions of all recipes are shown.
Largest version	1.2	Only the recipe versions that have the status 1 or 2 have the largest versions shown.
All	All	All recipe versions are displayed.
All	1.2	All recipe versions with the status 1 or 2 are displayed.
2.3	All	All recipe versions 2 and 3 are displayed regardless of status.
2.3	1.2	The respective recipe version 2 and 3 is displayed if these have either the status 1 or 2.

You can read more about the RGM in the Recipegroup Manager manual.

EQUIPMENT MODELING

Selection of an Equipment model.

Select the desired equipment model from the list or create a new equipment model.

Note: Equipment models are managed in the global project. A global project must be present here in order to be able to create a new equipment model.

datasets for Batch Control:

The following datasets are available for the Batch Control module:
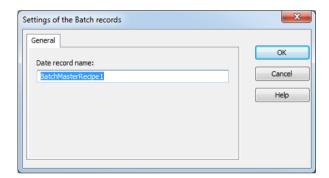
- ▸ Master recipe (on page 237)

- ▸ Control recipe (on page 238)

- ▸ Recipe screens (on page 238)

- ▸ Matrix cells (on page 241)

- ▶ PFC structure (on page 242)

- ▶ Phases (on page 243)

- ▶ Parameter (on page 244)

- ▶ Transitions (on page 245)

- ▶ Unit allocations (on page 246)

- ▶ Operation instance (on page 247)

Master recipe

Configuration of a data set for a master recipe in the Batch Control module.
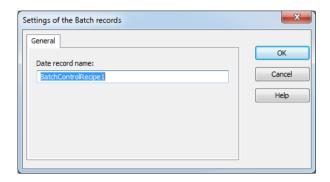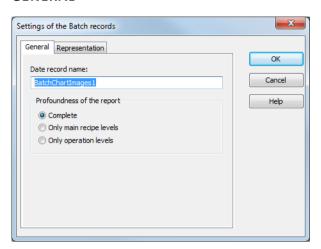

| Parameter	Description
Date set name	Name of the data set: ▸ must not be empty ▸ Must correspond exactly to the name of the data set in the RDL file (on page 254) ▸ Must not contain any spaces or special characters
OK	Applies settings and closes the dialog.
Cancel	Discards changes and closes dialog.
Help	Opens online help.

Control recipe

Configuration of a data set for a control recipe in the Batch Control module.

Parameter	Description
Date set name	Name of the data set: ▸ must not be empty ▸ Must correspond exactly to the name of the data set in the RDL file (on page 254) ▸ Must not contain any spaces or special characters
OK	Applies settings and closes the dialog.
Cancel	Discards changes and closes dialog.
Help	Opens online help.

Recipe screens

Configuration of a data set for recipe screens in the Batch Control module. For screens, in addition to the report detail, it is also possible to prescribe the display in Runtime.

GENERAL
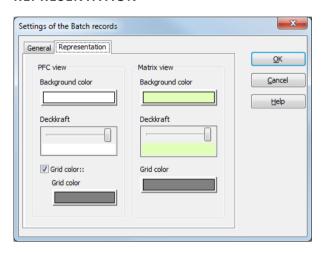
Parameter	Description
Date set name	Name of the data set:
	▸ must not be empty
	▸ Must correspond exactly to the name of the data set in the RDL file (on page 254)
	▸ Must not contain any spaces or special characters
Profoundness of the report	Configuration of the report detail:
	▸ Complete: Report contains data from the whole recipe, including its operations.
	▸ Only main recipe levels: Report only contains data from the main recipe.
	▸ Only partial recipe levels: Report only contains data from the operations contained in the recipe

OK	Applies settings and closes the dialog.
Cancel	Discards changes and closes dialog.
Help	Opens online help.

REPRESENTATION
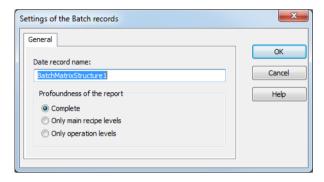
Parameter	Description
PFC view	Display for PFC recipes
Background color	Definition for the background color. Click on the color and a dialog opens to select a color or a palette.
Opacity	Setting of the opacity using a slider.
Show grid	`Active`: Grid is displayed.
Grid color	Definition for the grid color. Click on the color and a dialog opens to select a color or a palette.
Matrix view	Display for matrix recipes
Background color	Definition for the background color. Click on the color and a dialog opens to select a color or a palette.
Opacity	Setting of the opacity using a slider.
Grid color	Definition for the grid color. Click on the color and a dialog opens to select a color or a palette.

OK	Applies settings and closes the dialog.
Cancel	Discards changes and closes dialog.
Help	Opens online help.

Matrix cells

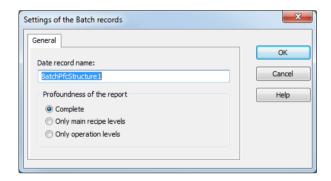Configuration of a data set for the structure of a matrix recipe in the Batch Control module.

Parameter	Description
Date set name	Name of the data set: ▸ must not be empty ▸ Must correspond exactly to the name of the data set in the RDL file (on page 254) ▸ Must not contain any spaces or special characters
Profoundness of the report	Configuration of the report detail: ▸ Complete: Report contains data from the whole recipe, including its operations. ▸ Only main recipe levels: Report only contains data from the main recipe. ▸ Only partial recipe levels: Report only contains data from the operations contained in the recipe

OK	Applies settings and closes the dialog.
Cancel	Discards changes and closes dialog.
Help	Opens online help.

PFC structure

Configuration of a data set for the structure of a PFC recipe in the Batch Control module.

Parameter	Description
Date set name	Name of the data set: ▸ must not be empty ▸ Must correspond exactly to the name of the data set in the RDL file (on page 254) ▸ Must not contain any spaces or special characters
Profoundness of the report	Configuration of the report detail: ▸ Complete: Report contains data from the whole recipe, including its operations. ▸ Only main recipe levels: Report only contains data from the main recipe. ▸ Only partial recipe levels: Report only contains data from the operations contained in the recipe

OK	Applies settings and closes the dialog.
Cancel	Discards changes and closes dialog.
Help	Opens online help.

Phases

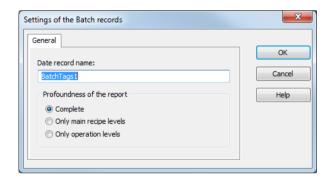Configuration of a data set for phases in the Batch Control module.

Parameter	Description
Date set name	Name of the data set: ▸ must not be empty ▸ Must correspond exactly to the name of the data set in the RDL file (on page 254) ▸ Must not contain any spaces or special characters
Profoundness of the report	Configuration of the report detail: ▸ Complete: Report contains data from the whole recipe, including its operations. ▸ Only main recipe levels: Report only contains data from the main recipe. ▸ Only partial recipe levels: Report only contains data from the operations contained in the recipe

OK	Applies settings and closes the dialog.
Cancel	Discards changes and closes dialog.
Help	Opens online help.

Parameter

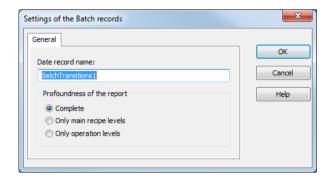Configuration of a data set for parameters in the Batch Control module.

Parameter	Description
Date set name	Name of the data set: ▸ must not be empty ▸ Must correspond exactly to the name of the data set in the RDL file (on page 254) ▸ Must not contain any spaces or special characters
Profoundness of the report	Configuration of the report detail: ▸ Complete: Report contains data from the whole recipe, including its operations. ▸ Only main recipe levels: Report only contains data from the main recipe. ▸ Only partial recipe levels: Report only contains data from the operations contained in the recipe

OK	Applies settings and closes the dialog.
Cancel	Discards changes and closes dialog.
Help	Opens online help.

Transitions

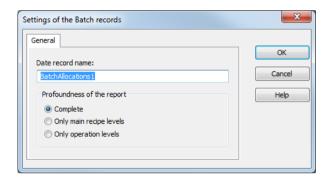Configuration of a data set for transitions in the Batch Control module.

Parameter	Description
Date set name	Name of the data set: ▸ must not be empty ▸ Must correspond exactly to the name of the data set in the RDL file (on page 254) ▸ Must not contain any spaces or special characters
Profoundness of the report	Configuration of the report detail: ▸ Complete: Report contains data from the whole recipe, including its operations. ▸ Only main recipe levels: Report only contains data from the main recipe. ▸ Only partial recipe levels: Report only contains data from the operations contained in the recipe

OK	Applies settings and closes the dialog.
Cancel	Discards changes and closes dialog.
Help	Opens online help.

Unit allocations

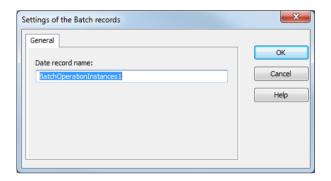Configuration of a data set for unit allocation in the Batch Control module.

Parameter	Description
Date set name	Name of the data set: ▸ must not be empty ▸ Must correspond exactly to the name of the data set in the RDL file (on page 254) ▸ Must not contain any spaces or special characters
	Configuration of the report detail: ▸ Complete: Report contains data from the whole recipe, including its operations. ▸ Only main recipe levels: Report only contains data from the main recipe. ▸ Only partial recipe levels: Report only contains data from the operations contained in the recipe

OK	Applies settings and closes the dialog.
Cancel	Discards changes and closes dialog.
Help	Opens online help.

Operation instance

Configuration of a data set for operation instances in the Batch Control module.

Parameter	Description
Date set name	Name of the data set: ▸ must not be empty ▸ Must correspond exactly to the name of the data set in the RDL file (on page 254) ▸ Must not contain any spaces or special characters
OK	Applies settings and closes the dialog.
Cancel	Discards changes and closes dialog.
Help	Opens online help.

Export or print function report
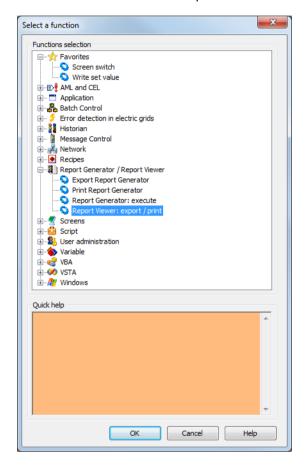
With the Report Viewer: export / print function, reports can be evacuated in Runtime, to their own thread as a PDF or as an online-printout. Because the rendering of the file is asynchronous, it ensures that Runtime can continue to be operated during rendering. The execution of functions of a low priority is however blocked until the Report Viewer has completed rendering. Output as a PDF or

printout can alternatively be carried out by means of control elements in the screen (on page 287), whereby the operability of Runtime can be restricted until the end of the output.

To create the function:

1.   Select the **New function...** command in the **Functions** node

2.   The function selection is opened.

3.   navigate to the node **Report Generator / Report Viewer**

4.   select the function **Report Viewer: export / print**

5.   The dialog for configuration is opened

6.   Details on the configuration of the dialog:

- As a PDF: see Configure output as a PDF (on page 249) section
- To the printer: see Configure output to the printer (on page 252) section

- Report definition: see Report definition (on page 202) chapter, Screen switching to Report Viewer type screen (on page 201)
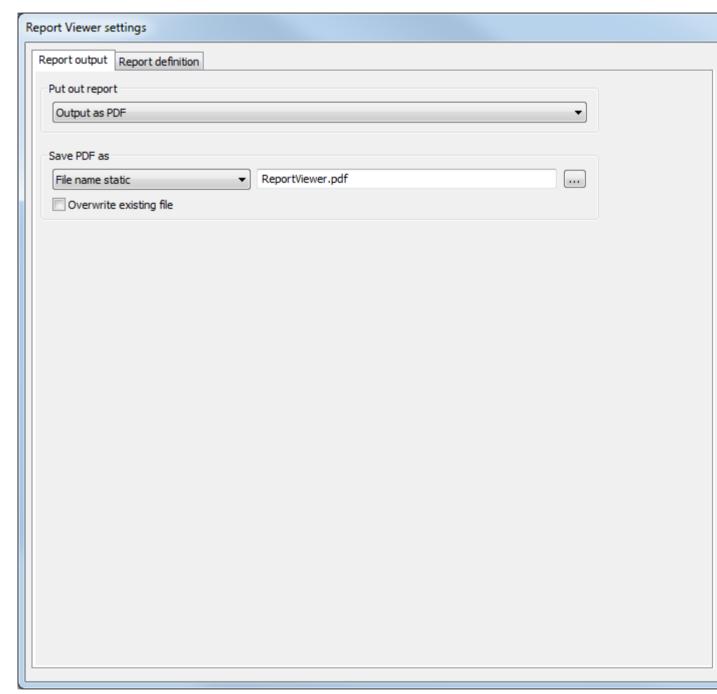
Configure output as PDF

To output the report as a PDF:

1. For the `Report output` option, select `Output as PDF`

2. Define a fixed filename or select a variable that defines the filename in Runtime.

3. Decide whether the existing data is to be overwritten

4. Confirm your configuration by clicking on the OK button

Report Viewer settings

Report output | Report definition

Put out report

Output as PDF

Save PDF as

File name static | ReportViewer.pdf | ...

☐ Overwrite existing file

| Parameter	Description
`Put out report`	Selection of how the report is to be output from a drop-down list: ▸ `Output as PDF`: Creates PDF according to the configuration ▸ `Output on printer`: Outputs the report via the defined printer.
`Save PDF as`	Selection of the issuing of the file name from a drop-down list: ▸ `File name static`: Name is entered in the text field and selected from the file Explorer after clicking on the `...` button. ▸ `File name from variable`: Clicking on the `...` button opens the dialog to select a variable that sends the filename for the report in Runtime. **Composition of the static filename:** For static file names, the components of the name are compiled using placeholders: ▸ **<ProjectName>**: Name of the project ▸ **<DateTime>**: Date and time in accordance with system settings ▸ **<Date>**: Date in accordance with system settings ▸ **<Time>**: Time in accordance with system settings The placeholders can also be inserted into the text field by right-clicking on the text field above the context menu at the current cursor position. The file name can be given as either an absolute path or as a relative path (relative to the export folder of the project). **Save location for PDFs:** As in the folder defined for export. Standard path: `%PUBLIC%\Documents\zenon_Projects\[PROJECTNAME]>\Export`
`Overwrite existing file`	`Active`: Existing data in the save location is overwritten.
OK	Accepts changes in all tabs and closes the dialog.
Cancel	Discards all changes and closes the dialog.

| Help | Opens the online help. |
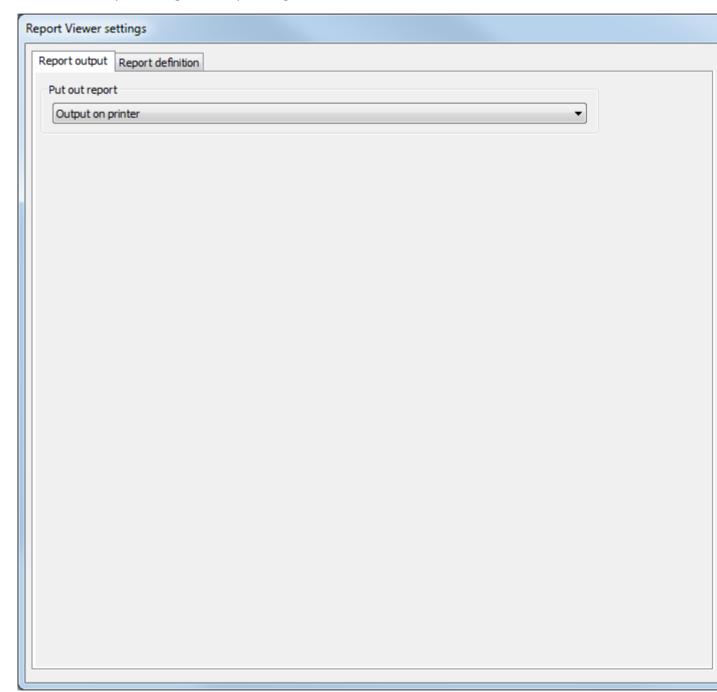
Configure output to the printer

To configure the report for a printer:

1. For the `Report output` option, select `Output to printer`

2. Confirm your configuration by clicking on the ок button

Parameter	Description
`Put out report`	Selection of how the report is to be output from a drop-down list: ▸ `Output as PDF`: Creates PDF according to the configuration ▸ `Output on printer`: Outputs the report via the defined printer.
OK	Accepts changes in all tabs and closes the dialog.
Cancel	Discards all changes and closes the dialog.
Help	Opens the online help.

2.2.2 Report definition files

Report definition files (RDL) must be type 2 files in order to be able to be used in the zenon Report Viewer. Type 3 RDL files can be displayed, however features that are not present in version 2 are displayed as an empty space.

To create your own datasets or edit existing ones, you need appropriate software to render the RDL files. You have the following options for this:

▸ SQL Server Data Tool (SSDT):

Is supplied by COPA-DATA from zenon 7.10 and also installed when installing zenon.

SSDT consists of Microsoft Visual Studio 2010 with other project types that are only available for SQL Server Business Intelligence and is the primary environment for business services projects such as reporting. Templates for creating objects necessary for the Business Intelligence solution are provided as well as designer, tools and assistants for processing these objects.

When calling up an RDL file to edit with SSDT, a selection window to set the environment setting for Visual Studio 2010 may be shown:

Select the `Business Intelligence Settings` options and click on the Start `Visual Studio` button.

▶   Microsoft Report Builder 2.0:

Is available on the Microsoft download page
(http://www.microsoft.com/download/en/details.aspx?id=24085) as a download. It provides a wizard for the configuration of datasets and is recommended for users without special knowledge for the configuration of RDL files.
Attention: Note the licensing conditions before downloading.
Users of the zenon Analyzer already have Report Builder 3 and the SQL license required for the Report Builder. However, Report Builder 2 must be installed and used for editing the datasets of the Report Viewer.

▶   Software from other suppliers:

Software from other manufacturers can also be used to edit report definition files.

If both the MS Report Builder and the Microsoft SQL Server Data Tools are installed, you can select the desired tool in a dialog.
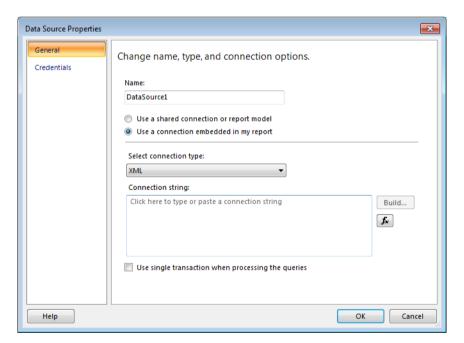
> ⚠ **Attention**
>
> *Note:*
>
> ▶ RDL files should always be created and edited with the same tool.
>
> ▶ RDL versions 2 and 3 must not be mixed.
>
> ▶ If an RDL file was saved with SSDT, it can no longer be edited with the MS Report Builder 2.0.

DATA SOURCE

The properties for the data source (superordinate to the datasets) must not be changed.

The standard settings are:

Parameters	Description
Name	DataSource1
Connection	Use a connection embedded in my report
Select connection type	XML

EDITING OF RDL FILES IN PROJECTS WITH DISTRIBUTED ENGINEERING

Note, when editing RDL files with distributed engineering, that files are not automatically set to "Check out" when opened.

▸ It is possible to open RDL files with the SSDT or the MS Report Builder (on page 258), to edit these and save changes locally, without the file being set to "Check out". These changes can however be accepted on the server and are thus lost again when synchronizing or with Check out. The older version of the server is automatically adopted.

▸ It is also possible to open an RDL file that has been set to "Check out" on another computer, to edit this and to save changes locally. The file is not write protected when setting "Check out". These local changes are also overwritten again when synchronizing.

To save changes on a lasting basis, the RDL file must be set to "Check out" before opening.

> ⚠ **Attention**
>
> *RDL files may only be renamed in the zenon Editor.*
>
> *If RDL files are renamed in SSDT or MS Report Builder, the new name is not taken on in projects with distributed engineering and changes are lost.*

Default.rdl

Default.rdl is called up if a new RDL file is created in the Editor. There are already pre-defined datasets available in the supplied default.rdl, which can be provided with data in Runtime:

▸ General
  - Alarm Message List
  - Chronological Event List
  - Archive values

- Online values

▶ <u>Recipegroup Manager</u>

- Recipe data

- Recipe values

▶ <u>Batch Control</u>

- Master recipe

- Control recipe

- Recipe screens

- Phases

- Parameter

- Transitions

- Matrix cells

- Unit allocations

- PFC elements

- Operation instance

If more datasets are required, these can be created (on page 264) with any desired name. If fewer datasets are required, these must be deleted from the RDL file. The names of the datasets must correspond to the dataset names in the screen switching (on page 201).

Note: The table schematic (on page 271) for the data categories is stipulated and must not be modified.

Creating and editing RDL files with MS Report Builder 2.0

MS Report Builder is available on the Microsoft download page (http://www.microsoft.com/download/en/details.aspx?id=24085) as a free download.
Attention: Note the licensing conditions before downloading.
Users of the zenon Analyzer already have Report Builder 3 and the SQL license required for the Report Builder. However, Report Builder 2 must be installed and used for editing the datasets of the Report Viewer.
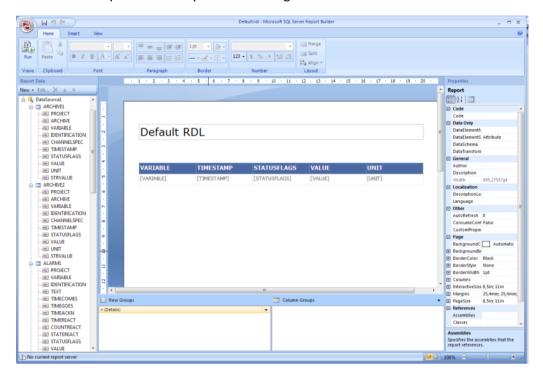
Install the Report Builder and check if it is correctly linked to zenon.
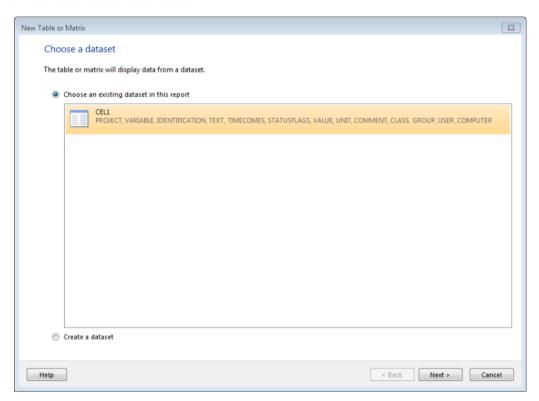
CREATING AN RDL FILE

To create an RDL file:

1. Select, in the zenon Editor, in the **File/Report Viewer** node in the context menu or the **New report definition file** command in the toolbar. A new report definition file is created, based on a default file (on page 254).

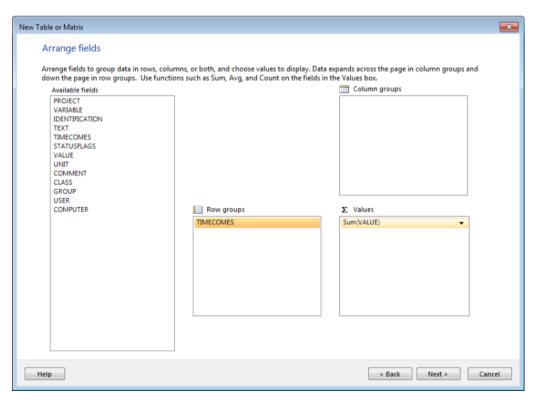2. Start the Report Builder by double clicking on the file.

3. Adapt the datasets (on page 264) to the configuration in the screen switching function by:

   - Rename: Datasets must be identical with the dataset names in screen switching

   - Delete: Datasets that do not have any corresponding linking in the screen switching and are thus not supplied with data lead to an error message in Runtime

   - Create new: Datasets that are configured in the screen switching but do not have any corresponding entries in the RDL file lead to an error message in Runtime.

4. Select, in the **Insert** menu, the command *Table -> Table Wizard.*
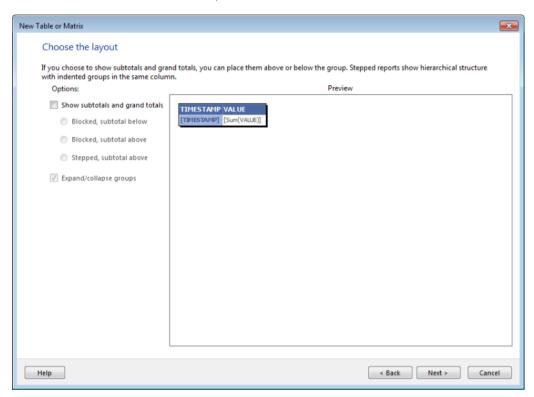
5. The wizard is started.

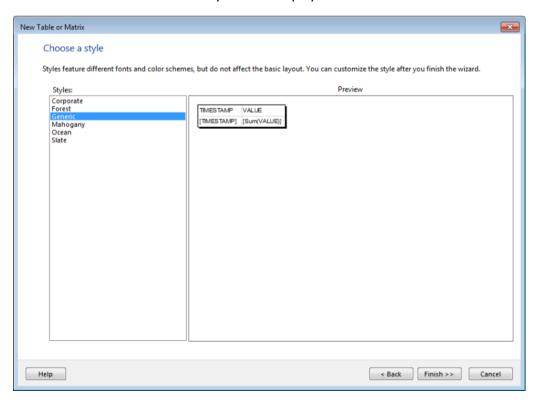6. Select the desired dataset.

7. Locate the fields as desired.

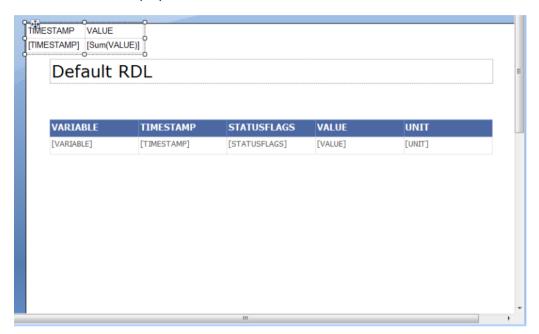8. Click on **Next** and select the layout.

9. Click on **Next** and select the style of the display.

10. Click on `Finish`.
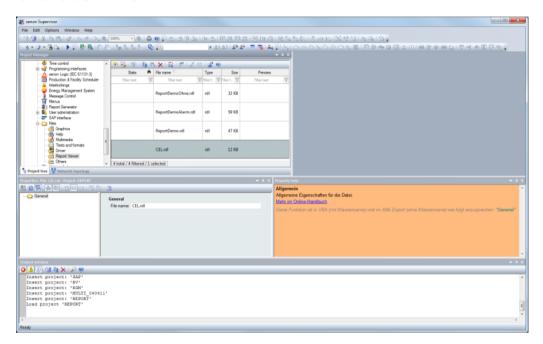
    The dataset is displayed with the finished table

11. Delete the sample table.

12. Place the newly created table at the desired location.

13. Adapt the heading as desired.
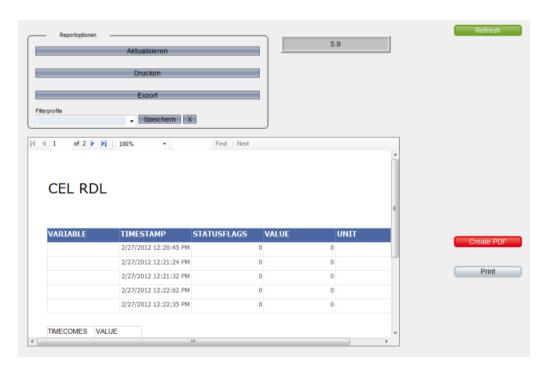
14. Repeat the process for all datasets to be configured.

15. Save the RDL file.
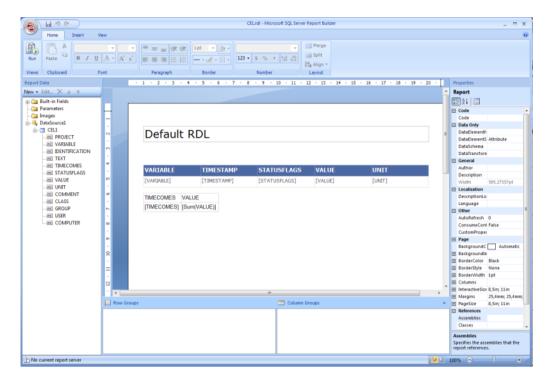
Result in Runtime:

EDITING AN RDL FILE

To edit an RDL file, double click the file or select the `Open report definition file` command. Use the same process as creation for editing.
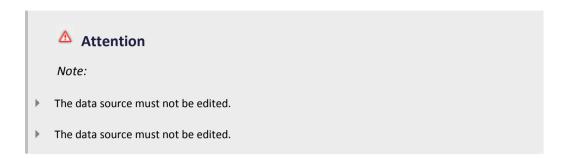
Administering datasets

You can do the following with datasets in the Report Builder:

▶ Rename

▶ delete

▶ create new

Number and name of the dataset of an RDL file must always correspond to that of the dataset defined in screen switching.
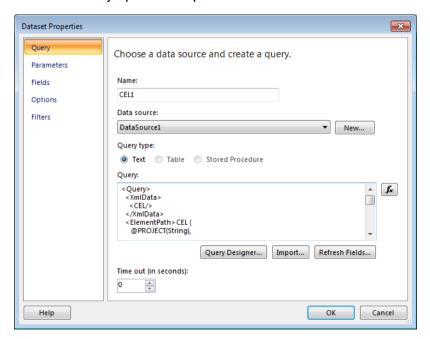
RENAMING A DATASET

To rename a data set:

1. Right-click on the data set

2. Select the **Properties** command in the context menu

3. The dialog for setting the properties of the data set is opened

4. Give it the desired name in the `Name` property

Note: the name must correspond to the name in the screen switching (on page 201). It must not contain any spaces or special characters.
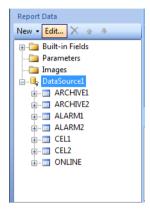
5. Leave the other settings unchanged

6. Close the dialog by clicking on oĸ

DELETING THE DATA SET

To delete a data set:

1. Highlight the data set to be deleted in the report data window below the data source
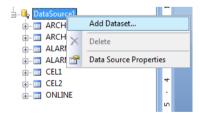
2. Select the `Delete` command in the context menu

3. the data set is deleted after confirmation is requested
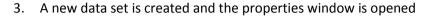
CREATING A NEW DATA SET
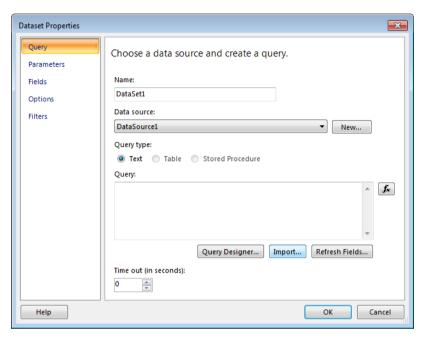
To create a new data set:

1. Right-click on the data source

2. Select `Add dataset` in the context menu

3. A new data set is created and the properties window is opened

4. name it as you wish

Note: the name must correspond to the name in the screen switching (on page 201). It must not contain any spaces or special characters.

5. Insert the appropriate query; you have three options for doing this:

a) `Query Designer`: Opens the dialog for manual input of the query

b) `Import`: Import from an existing RDL file

c) Copying manually from an existing dataset: First open the corresponding source dataset, copy the content of the query and paste the content of the query in the `Query` field in the new dataset

6. Close the dialog by clicking on OK

IMPORTING A QUERY

To import a query from an existing RDL file:

1. Click on the `Import` bin the dataset properties dialog

2. The file selection assistant is opened

3. Search for and open the desired RDL file

4. The import dialog is displayed

```
Import Query                                                    [×]

Choose the query you want to import from 'Default.rdl'.

Datasets:                      Query:

  ARCHIVE1                     <Query>
  ARCHIVE2                       <XmlData>
  ALARM1                            <ARCHIVE/>
  ALARM2                         </XmlData>
  CEL1                           <ElementPath>ARCHIVE {
  CEL2                                @PROJECT(String),
  ONLINE                              @ARCHIVE(String),
                                      @VARIABLE(String),
                                      @IDENTIFICATION(String),
                                      @CHANNELSPEC(String),
                                      @TIMESTAMP(Date),
                                      @STATUSFLAGS(Integer),
                                      @VALUE(Float),
                                      @UNIT(String),
                                      @STRVALUE(String)}
                                   </ElementPath>
                               </Query>

                                          [ Import ]   [ Cancel ]
```
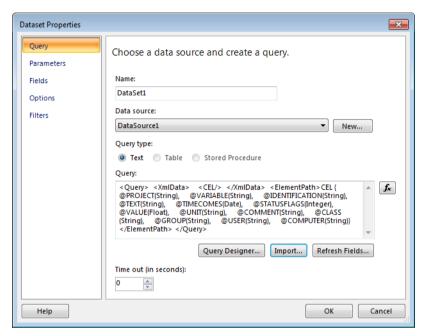
5. Highlight the desired dataset

```
Import Query                                                    [×]

Choose the query you want to import from 'Default.rdl'.

Datasets:                      Query:

  ARCHIVE1                     <Query>
  ARCHIVE2                       <XmlData>
  ALARM1                            <CEL/>
  ALARM2                         </XmlData>
  CEL1                           <ElementPath>CEL {
  CEL2                                @PROJECT(String),
  ONLINE                              @VARIABLE(String),
                                      @IDENTIFICATION(String),
                                      @TEXT(String),
                                      @TIMECOMES(Date),
                                      @STATUSFLAGS(Integer),
                                      @VALUE(Float),
                                      @UNIT(String),
                                      @COMMENT(String),
                                      @CLASS(String),
                                      @GROUP(String),
                                      @USER(String),
                                      @COMPUTER(String)}
                                   </ElementPath>
                               </Query>

                                          [ Import ]   [ Cancel ]
```

6. click on **Import**

7. The query of the source data set is pasted to the dataset that is currently open

Create and edit RDL files with the help of the Microsoft Business Intelligence Development Studio

The Microsoft Business Intelligence Development Studio is shipped and installed together with zenon.

To create and edit RDL files with the help of the Microsoft Business Intelligence Development Studio, you basically act the same way as with the Report Builder 2.0 (on page 258).

The differences are primarily:

▶ For the report to be edited, you must first create a new project of type `Report Server Project` - Solution - in the Microsoft Business Intelligence Development Studio.
They are created automatically when you open the RDL file via double click. When closing the Business Intelligence Development Studios you can save the project and the solution in the temporary folder. If you do this, you are no longer requested to do so.
The datasets and the report can now be edited in Microsoft Business Intelligence Development Studio.

▶ There is no table wizard in the Microsoft Business Intelligence Development Studio; the tables of the datasets must be configured manually.

▶ The version is not checked:

This way, features from version 3 can be added to RDL files from version 2. These are not displayed in Runtime however.

▶ In the Microsoft Business Intelligence Development Studio the datasets are separated from the data sources. The data source must not be changed.

INSTRUCTION

For a comprehensive guide about the use of the Microsoft Business Intelligence Development Studios see the Microsoft website in several languages:

Language	Link
English:	http://msdn.microsoft.com/en-en/library/ms173767.aspx (http://msdn.microsoft.com/en-en/library/ms173767.aspx)
German:	http://msdn.microsoft.com/de-de/library/ms173767.aspx (http://msdn.microsoft.com/de-de/library/ms173767.aspx)
French:	http://msdn.microsoft.com/fr-fr/library/ms173767.aspx (http://msdn.microsoft.com/fr-fr/library/ms173767.aspx)
Italian:	http://msdn.microsoft.com/it-it/library/ms173767.aspx (http://msdn.microsoft.com/it-it/library/ms173767.aspx)

DELETE DATASETS

Unused datasets must be deleted in the Microsoft Business Intelligence Development Studio as they may cause errors in the Runtime.

To delete datasets:

1. click on `View`

2. select `Report Data`

3. navigate to `Datasets`

4. delete all unused datasets

DISPLAY DATASET WINDOW
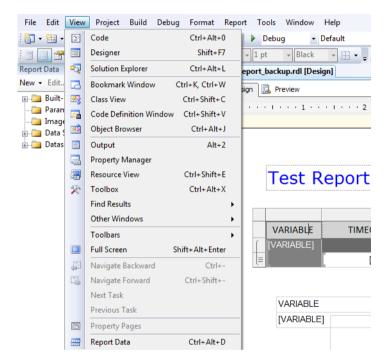
If the `Dataset` window is not displayed, it can be called up with the following steps:

▶ Select report

▶ Click on the `View` menu in the main menu

▶ Scroll to the end of the list

▶ Select `Report Data`

Note: If the menu item is not displayed, then no report was selected.

Alternatively, the key combination `Ctrl+ALT+D` can be pressed after selecting the report.

Definition of datasets

The columns of the dataset types are stipulated.

Definition of datasets for:

▶ AML (on page 273)

▶ CEL (on page 274)

▶ ARCHIVE (on page 275)

▶ ONLINE (on page 276)

▶ RGM

- RGM recipe data (on page 277)
- RGM recipe value (on page 278)

▶ Batch Control

- Master recipe (on page 279)
- Control recipe (on page 280)
- Phases (on page 284)
- Parameter (on page 285)

Note: If zenon Analyzer reports are to be reused in the Report Viewer, the following applies:

▶ Language-switchable texts have already had the language switched

▶ Unit-switchable values are given in conversion units

▶ Time stamps correspond to the local time of the executing computer

AML

Name of the column	Data type	Meaning
PROJECT	STRING	Name of the project.
RESSOURCESLABEL	STRING	Resources label.
VARIABLE	STRING	Name of the variable.
IDENTIFICATION	STRING	Identification of the variable.
TEXT	STRING	Entry text.
TIMECOMES	DATE	Time when the alarm occurs.
TIMEGOES	DATE	Time.
TIMEACKN	DATE	Time of acknowledgement.
TIMEREACT	DATE	Time of reactivation.
COUNTREACT	INTEGER	Number of reactivations.
STATEREACT	INTEGER	Status of reactivation.
STATUSFLAGS	INTEGER	Status bits.
VALUE	FLOAT	Value of the variables.
UNIT	STRING	Unit.
COMMENT	STRING	User comments.
CLASS	STRING	Alarm/Event class
GROUP	STRING	Alarm/Event group...
USER	STRING	name of the user.
COMPUTER	STRING	Name of the computer.

CEL

Name of the column	Data type	Meaning
PROJECT	STRING	Name of the project.
RESSOURCESLABEL	STRING	Resources label.
VARIABLE	STRING	Name of the variable.
IDENTIFICATION	STRING	Identification of the variable.
TEXT	STRING	Entry text.
TIMECOMES	STRING	Time when the event occurs.
STATUSFLAGS	INTEGER	Status bits.
VALUE	FLOAT	Value of the variables.
UNIT	STRING	Unit.
COMMENT	STRING	User comments.
CLASS	STRING	Alarm/Event class
GROUP	STRING	Alarm/Event group...
USER	STRING	name of the user.
COMPUTER	STRING	Name of the computer.

Archive

Name of the column	Data type	Meaning
PROJECT	STRING	Name of the project.
ARCHIVE	STRING	Name of the archive.
RESSOURCESLABEL	STRING	Resources label.
VARIABLE	STRING	Name of the variable.
IDENTIFICATION	STRING	Identification of the variable.
CHANNELSPEC	STRING	Variable including type of data reduction.
TIMESTAMP	STRING	Time stamp.
STATUSFLAGS	INTEGER	Status bits.
VALUE	FLOAT	Value of the variables.
UNIT	STRING	Unit.
STRVALUE	STRING	Value as string.

Online

Name of the column	Data type	Meaning
PROJECT	STRING	Name of the project.
RESSOURCESLABEL	STRING	Resources label.
VARIABLE	STRING	Name of the variable.
IDENTIFICATION	STRING	Identification of the variable.
TIMESTAMP	STRING	Time stamp.
STATUSFLAGS	INTEGER	Status bits.
VALUE	FLOAT	Value of the variables.
UNIT	STRING	Unit.
STRVALUE	STRING	Value as string.

RGM recipe data

Name of the column	Data type	Meaning
PROJECTNAME	STRING	Name of the project.
GROUPNAME	STRING	Name of the recipe group.
NAME	STRING	Recipe name.
NUMBER	UINT	Recipe number.
STATUSNUMBER	UINT	Status number
STATUSTEXT	STRING	Status text.
VERSION	UINT	Version.
COMMENT1	STRING	Comment 1
COMMENT2	STRING	Comment 2
COMMENT3	STRING	Comment 3
COMMENT4	STRING	Comment 4
COMMENT5	STRING	Comment 5
COMMENT6	STRING	Comment 6
COMMENT7	STRING	Comment 7
COMMENT8	STRING	Comment 8
USERLEVEL	UINT	User authorization.
CHANGETIME	DATETIME	Time of the change.
CHANGEUSER	STRING	User that was logged in when the change was made.

RGM recipe value

Name of the column	Data type	Meaning
PROJECTNAME	STRING	Name of the project.
GROUPNAME	STRING	Name of the recipe group.
NAME	STRING	Recipe name.
VERSION	UINT	Version.
RESSOURCESLABEL	STRING	Resources label.
VARIABLENAME	STRING	Variable name.
VARIABLETAG	STRING	Variable identification.
SYMBOLVALUE	STRING	Symbolic address for variable.
VALUE	STRING	value.
UNIT	STRING	Unit.
MIN	DOUBLEWORD	Minimum value.
MAX	DOUBELWORD	Maximum value.
ACTION	STRING	RGM recipe action to be executed.
FILTER	STRING	Recipe variable filter.

Master recipe

Name of the column	Data type	Meaning
MrId	INTEGER	ID of the recipe.
MrName	STRING	Name of the recipe.
MrDescription	STRING	Description of the recipe.
MrStatus	STRING	Status of the recipe
RecipeType	STRING	Type of the recipe.
ReeStatus	STRING	Status of the REE.
ReeModus	STRING	Mode of the REE.
ApprovalTime	DATE	Time when the recipe was approved.
ApprovalUserID	STRING	ID of the user who approved the master recipe.
ApprovalUserName	STRING	Name of the user who approved the master recipe.
OutdatedTime	DATE	Time when the recipe was set to "obsolete".
OutdatedUserID	STRING	ID of the user who set the recipe to "obsolete".
OutdatedUserName	STRING	Name of the user who set the recipe to "obsolete".

Control recipe

Name of the column	Data type	Meaning
CrId	INTEGER	ID of the control recipe.
CrName	STRING	Name of the control recipe.
CrDescription	STRING	Description of the control recipe.
CrStatus	STRING	Status of the control recipe.
MrId	INTEGER	ID of the master recipe.
MrName	STRING	Name of the master recipe.
MrDescription	STRING	Description of the master recipe.
MrStatus	STRING	Status of the master recipe.
RecipeType	STRING	Type of the recipe.
ReeStatus	STRING	Status of the REE.
ReeModus	STRING	Mode of the REE.
CrJobID	STRING	Job ID of the control recipe
ApprovalTime	DATE	Time when the master recipe was approved
ApprovalUserID	STRING	ID of the user who approved the master recipe.
ApprovalUserName	STRING	Name of the user who approved the master recipe.
CreationTime	DATE	Time of creation of the control recipe.
CreationUserID	STRING	ID of the user who created the master recipe.
CreationUserName	STRING	Name of the user who created the master recipe.
StartingTime	DATE	Time when the recipe was started
StartingUserID	STRING	ID of the user who started the master recipe.
StartingUserName	STRING	Name of the user who started the recipe.
OutdatedTime	DATE	Time when the master recipe was set to "obsolete".
OutdatedUserID	STRING	ID of the user who set the master recipe to "obsolete".
OutdatedUserName	STRING	Name of the user who set the master recipe to "obsolete".

Recipe screens

Name of the column	Data type	Meaning
`Name`	STRING	Name of the recipe.
`Description`	STRING	Description of the recipe.
`RecipePart`	STRING	Master recipe, control recipe or operation instance
`OperationTemplateID`	INTEGER	ID of the Operation template.
`Type`	STRING	Type of the recipe: PFC recipe or matrix recipe.
`ImagePNG`	STRING	Screenshot of the recipe Must be created especially.

CREATING A SCREEN

1. Insert a new screen in the Report Builder.

2. Set the source to `Database`.

3. Enter, for the `Use this field`: property, `ImagePNG`.

4. Enter, for the MIME type property: `image/png`.

5. Insert the screen using the Insert option or by dragging & dropping in the report.

6. In the `size` property, adjust the size to `Original Size`.

7. If necessary, amend the `Padding` property for the distance from the screen to the cell borders.

Matrix cells

Name of the column	Data type	Meaning
StepNr	INTEGER	Number of the Step.
StepDescr	STRING	Description of the Step.
ObjectID	INTEGER	ID of the object in the recipe.
ObjectName	STRING	Name of the object in the recipe.
ObjectDescr	STRING	Description of the object in the recipe.
ObjectType	STRING	Type of object in the recipe. (phase, transition, ...)
ObjectUnit	STRING	Unit of the object.
OperationTemplateID	INTEGER	ID of the Operation template. 0 = main recipe.
OperationTemplateName	STRING	Name of the Operation template. empty = main recipe.

PFC structure

Name of the column	Data type	Meaning
ChartRow	INTEGER	Rung in the recipe.
ChartCol	INTEGER	Column in the recipe.
ObjectID	INTEGER	ID of the object in the recipe.
ObjectName	STRING	Name of the object in the recipe.
ObjectDescr	STRING	Description of the object in the recipe.
ObjectType	STRING	Type of object in the recipe. (phase, transition, ...)
ObjectUnit	STRING	Unit of the object.
PrevIds	STRING	IDs of the previous objects.
NextIds	STRING	IDs of the following objects.
PrevNames	STRING	Names of the previous objects.
NextNames	STRING	Names of the following objects.
PrevTypes	STRING	Types of the previous objects.
NextTypes	STRING	Types of the following objects.
CellsSpan	STRING	Number of columns used (width of the object).
OperationTemplateID	INTEGER	ID of the Operation template. 0 = main recipe.
OperationTemplateName	STRING	Name of the Operation template. empty = main recipe.

Phases

Name of the column	Data type	Meaning
ObjectID	INTEGER	ID of the object in the recipe.
ChartRow	INTEGER	Lines in which the phase is located in the recipe.
ChartCol	INTEGER	Column in which the phase is located in the recipe.
PhaseName	STRING	Name of the phase.
PhaseDescr	STRING	Description of the phase.
UnitID	INTEGER	ID of the unit.
UnitName	STRING	Name of the unit.
UnitDescr	STRING	Description of the unit.
OperationInstID	INTEGER	ID of the operation instance in which the phase is located in the recipe.
OperationInstName	STRING	Name of the operation instance in which the phase is located in the recipe.
TOAllocation	STRING	Waiting period unit allocation.
TOInterlocking	STRING	Waiting period of the interlocking.
MinExecTime	STRING	Minimum execution time.
ExplanationMinExecTimeNeeded	BOOL	Reason for changing the minimum execution time necessary.
MaxExecTime	STRING	Maximum execution time.
TOFollowingCond	STRING	Waiting period for Following condition.
CondInterlocking	STRING	Interlocking condition.
CondDone	STRING	Condition for ended.
CondFailure	STRING	Condition for loss of communication.
CondConnReconnect	STRING	Condition for communication reestablished.
CondPausing	STRING	Condition for pause.
CondHolding	STRING	Condition for holding.
CondStopping	STRING	Condition for stop.

CondAborting	STRING	Condition for abort.
CondRestarting	STRING	Condition for restart.
CondEscaping	STRING	Escape condition.

Parameter

Name of the column	Data type	Meaning
TagName	STRING	Name of the tag.
TagDescr	STRING	Description of the tag.
TagType	STRING	Type of the tag.
TagValue	STRING	Name of the tag.
MeasUnit	STRING	Measuring unit
ValueMin	STRING	Minimum value.
ValueMax	STRING	Maximum value.
EditableInRecipe	BOOL	States if editable in the recipe.
EditableInCr	BOOL	States if editable in the control recipe.
ExplanationNeeded	BOOL	States if a reason for change is necessary.
TagModified	BOOL	States if a parameter was changed.
Variable	STRING	Name of the tag.
DataType	STRING	Data type of the variable.
PhaseID	INTEGER	ID of the phase.
PhaseName	STRING	Name of the phase.
OperationInstID	INTEGER	ID of the operation instance.
OperationInstName	STRING	Name of the operation instance.

Transitions

Name of the column	Data type	Meaning
ObjectID	INTEGER	ID of the transition in the recipe.
ChartRow	INTEGER	Row of the recipe.
ChartCol	INTEGER	Column of the recipe.
Condition	STRING	Condition for the transition.
OperationInstID	INTEGER	ID of the operation instance in which the transition is located.
OperationInstName	STRING	name of the operation instance in which the transition is located.

Unit allocations

Name of the column	Data type	Meaning
ObjectID	INTEGER	ID of the unit allocation in the recipe.
ChartRow	INTEGER	Row of the recipe.
ChartCol	INTEGER	Column of the recipe.
AllocationData	STRING	Unit allocation and number of the units allocated.
OperationInstID	INTEGER	ID of the operation instance in which the unit allocation is located.
OperationInstName	STRING	Name of the operation instance in which the unit allocation is located.

Operation instance

Name of the column	Data type	Meaning
ObjectID	INTEGER	ID of the operation instance in the recipe.
ChartRow	INTEGER	Row of the recipe.
ChartCol	INTEGER	Column of the recipe.
OperationName	STRING	Name of the operation.
OperationDescr	STRING	Description of the operation.
OperationType	STRING	Recipe type: PFC recipe or matrix recipe.
OperationTemplateID	INTEGER	ID of the Operation template.

2.2.3 Operating during Runtime

the Report Viewer screen is switched to in Runtime with a screen switching function. Depending on the configured report definition, reports for AML, CEL, archives and online values can be displayed. You can also do the following with these reports, if the corresponding functions are configured:

▶ Update: Reload report definition and data and display report
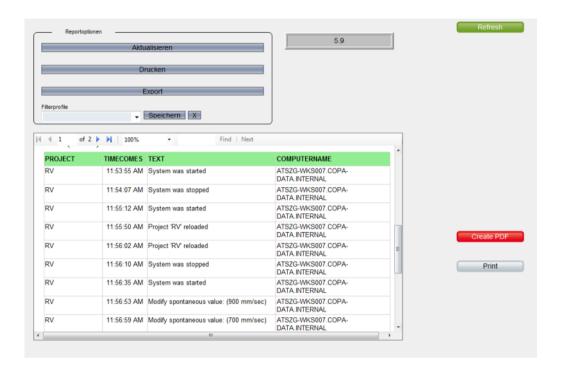
▶ print: on the printer defined for values and logs

▶ export: as a PDF file in the defined export folder

Parameter	Description
Update	Reloads the report definition and the data and displays the report with the updated data. After the report file in the Editor has been changed, the screen must be called up again; a reload is not sufficient.
Print	Prints the report on the printer defined for values and logs. **Note:** The printout can also be evacuated to its own thread (on page 247). This ensures that Runtime can be operated during printing.
Create PDF	Saves the report currently being displayed as a PDF file in the folder defined for export. Standard path: `%PUBLIC%\Documents\zenon_Projects\[PROJECTNAME]>\Export` **Note:** The PDF creation can also be evacuated to its own thread (on page 247). This ensures that Runtime can be operated during the creation and export.
Report list	Displays the reports defined in the screen switching function (on page 201). The list can be searched. Long lists are broken up into several pages.

NAVIGATION, ZOOM AND SEARCH

The report header offers elements for:

- ▶ Navigation

- ▶ Zoom

- ▶ Search

Parameter	Description
Page navigation	Displays number of pages and current page. Enables navigation in the document by means of ▸ Arrow keys ▸ Selection of a particular page
Zoom	Selection of a zoom level from a drop-down list: ▸ Page Width ▸ Whole Page ▸ 500% ▸ 200% ▸ 150% ▸ 100% (default) ▸ 75 % ▸ 50 % ▸ 25%
Search	Input of a search term in the search field. ▸ **Find**: Displays the first place of finding ▸ **Next**: Navigates to further places of finding Wildcards cannot be used.

2.2.4   Error treatment

ERROR MESSAGES IN POP-UPS

Error message	Meaning

`DataTable missing for DataSet:`
 	
`<DatasetName>`
	A dataset is used in the report definition, but there is no table with this name in the configuration. The report cannot be displayed.

`DataSet not used for DataTable:`
 	
`<TableName>`
	A table was defined, but the report definition does not use this name for a dataset. This message is only given after a missing table has been established.