zenon manual

Import - Export

v.7.11

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. Import - Export

Parts of a project (e.g. functions,variables, screens, etc.) can be imported to zenon or exported from zenon. Thus you can use, save or edit externally elements in other projects.

License information

Import and export are part of the standard license for the most formats.

Exception: *SQL export must be additionally licensed.*

3. General

The exported data can then be edited externally and be imported into the same or another project.

Attention

Limitations:

- Formulas:
  import and export via CSV or dBase does not support driver-specific variable settings, for example no formulas. Export and import these via XML.

- XML and backward compatibility:
  there is no backward compatibility with XML import/export. Data from older zenon versions cannot be taken over. The handover of data from newer to older versions is not supported.

The following functionalities of a project are available for import/export:

- Variable definition
  - Data types
  - Reaction matrices
  - Allocations
- Screens
  - Frames
  - Fonts
- Functions
  - Script - Functions
- Historian

- Standard recipes
- Recipegroup Manager
- Time control
- Interlockings
- Menu administration
- User administration

Here either all objects of one type or only selected objects can be exported.

- `All objects:` *Select the folder of the project manager -> Context menu XML Export all ...*
- `Selected objects:` *Select one or more objects in the detail view ->Context menu XML Export selected...*

On exporting/importing the objects are identified by their names not by their internal IDs, i.e.if objects are imported into another project, they do not have the same internal IDs there!

During the import a progress bar displays the status of the import/export.

> ⚠ Attention
>
> *If an element in the project and in the import file have the same name, then the element in the project is overwritten by the element from the import file. (The only exception are the data types.)*

The properties of the objects are listed in the export file with their internal name, that is also used for the access from VBA.

The properties are described in detail in the properties help of the Editor.

Properties that do not exist in the import file (e.g. because they were deleted), are filled with default settings for elements that do not exist in the project and are newly created. If the elements already exist in the project, the missing properties stay unchanged.

> Example
>
> *A variable first is imported without the offset. So the offset is initialised with the default value 0. With the S7 import the offset is changed to 100. Later the variable is imported again, the offset in the import file still missing. So the offset 100 stays unchanged.*

⚠ **Attention**

The XML file to be imported has to be consistent. There is no plausibility check on importing the file. If there are errors in the import file, this can lead to undesired effects, even crashes.

You particularly have to care for this, if not all properties exist in the XML file and thus are filled with default values. For example: A binary variable has a limit of 300.

4. Archives

The export file for the archives has the following sections:

Archive list

On exporting/importing archives the linked variables or functions are not automatically exported/imported. You have to care, that the needed variables or functions are exported/imported before. Either all archives at the same time or selected archives can be exported/ imported.

5. Screens

EXPORTING SCREENS

The included variables (on page 17) and functions (on page 15) are exported with the screens. The export file for the screens has the following sections:

- Screens
- Frames
- Symbols
- Variables
- Functions

To export screens:

1. highlight the desired screens
2. select **Export selected XML...** in the context menu of the detail view
   Alternate: select, in the context menu of the **Screen** node, the **XML export all...** command
3. the selection dialog for the saving location is opened
4. select the desired saving location
5. assign a name
6. pay attention to file type XML
7. confirm by clicking on the **Save** button

The selected screens are exported to an XML file and can be imported in this or in other projects at any time.

IMPORTING SCREENS

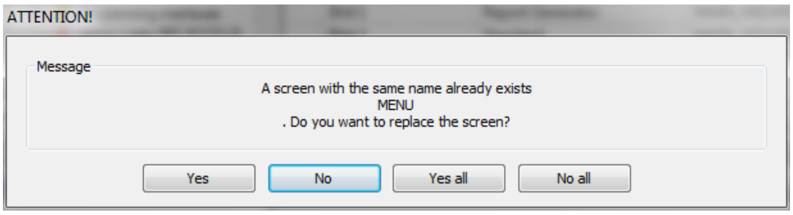
To import screens:

1. in the context menu of node **Screens** or in the detail view **Screens** select the **Import XML** command
2. the selection dialog for the saving location is opened
3. navigate to the saving location of the desired XML file
4. Select the desired file
5. confirm the import by clicking on the **Open** button

The screens are imported.

NAME CONFLICTS DURING THE IMPORT

At the import of a screen the name (attribute **ShortName**) which is stored in the XML file is used for the name of the screen which is created. If a screen with the same name already exist, the import is halted and an error message is displayed:



A screen with the same name already exists
MENU
. Do you want to replace the screen?

Parameters	Description
Yes	The screen described in the error message is replaced by the screen from the XML file at the import. If a new conflict occurs, the error message is displayed again.
No	The screen described in the error message is not imported. The existing screen is maintained. If a new conflict occurs, the error message is displayed again.
Yes, all	The screen described in the error message is replaced by the screen from the XML file at the import. This setting is used automatically when errors due to using the same name occur. All affected screen are replaced.
No, all	The screen described in the error message is not imported. The existing screen is maintained. This setting is used automatically when errors due to using the same name occur. All affected screen are maintained.

Note: Only the **ShortName** is used. The file name does not matter for the name of the screens which should be imported. Thus it cannot be used to solve the conflict.

VARIABLES AND FUNCTIONS IN SCREENS

Dynamic elements and Frames are automatically imported together with the screens. Variables and functions have to be imported manually before from the same file.

Only variables and functions of the first level are treated, i.e. variables and functions that are directly linked to the screen. As on the one hand variables can be linked to functions (e.g. setting values) but on the other hand functions can be linked to variables (e.g. limit functions), it can be necessary to first import the variables, then the functions, and then the variables again. Then all links should work correctly.

Example: A screen contains a button with a function **Send value to hardwrae** to a variable. The function is imported with the screen but not the variable.

Information

Here we recommend using the XML Import Wizards.

5.1 Frames

Frames can be imported/exported independently from the screens.

The frames are exported with the original coordinates (depending on the resolution of the source computer) When importing the frames, you have to be aware, that they are not automatically adapted to the resolution of the target computer.

5.2 Fonts

When exporting fonts, the following is exported:

- Font list
- Fonts

All these data are automatically imported with the fonts.

5.3 Symbols

Import/export of symbols is possible from the project and the global symbol library. Single or selected symbols or the entire library can be imported or exported. The path for the export can be defined freely.

EXPORT SYMBOLS

To export symbols:

1. highlight the desired symbols

2. select `Export selected XML...` in the context menu of the detail view
   Alternate: select, in the context menu of the `project symbol library` node, the `XML export all...` command
3. the selection dialog for the saving location is opened
4. select the desired saving location
5. assign a name
6. pay attention to file type XML
7. confirm by clicking on the **Save** button

The selected symbols are exported to an XML file and can be imported in this or in other projects at any time.

IMPORT SYMBOLS

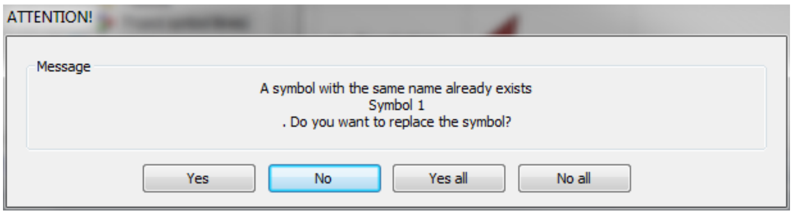
To import symbols:

1. in the context menu of the node or the detail view of the `Project symbol library` or in the detail view of the `Global symbol library` select `Import XML...` command
2. the selection dialog for the saving location is opened
3. navigate to the saving location of the desired XML file
4. Select the desired file
5. confirm the import by clicking on the `Open` button

The symbols are imported.

NAME CONFLICTS DURING THE IMPORT

At the import of a symbols the name (attribute `ShortName`) which is stored in the XML file is used for the name of the symbol which is created. If a symbol with the same name already exist, the import is halted and an error message is displayed:



A symbol with the same name already exists
Symbol 1
. Do you want to replace the symbol?

Parameters	Description
`Yes`	The symbol described in the error message is replaced by the symbol from the XML file at the import. If a new conflict occurs, the error message is displayed again.
`No`	The symbol described in the error message is not imported. The existing symbol is maintained. If a new conflict occurs, the error message is displayed again.
`Yes, all`	The symbol described in the error message is replaced by the symbol from the XML file at the import. This setting is used automatically when errors due to using the same name occur. All affected symbols are replaced.
`No, all`	The symbol described in the error message is not imported. The existing symbol is maintained. This setting is used automatically when errors due to using the same name occur. All affected symbols are maintained.

Note: Only the `ShortName` is used. The file name does not matter for the name of the symbol which should be imported. Thus it cannot be used to solve the conflict.

6. User

The export file for the users has the following sections:

User list

Locally users, that have been created in the project, can be exported/ imported.

Attention

The passwords of the users are exported in an encrypted form! So they cannot be changed directly in the export file.

7. Data types

The export file for the data types has the following sections:

Data type list

See also chapter: Import of variables and datatypes (on page 19).

7.1 Creating data types manually

Attention

Only for experts

If a data type should be entered in the XML file manually, the following definition should be used:

Information

```
<Apartment ShortName='zenon(R) type list' Version='0x00000000' >
<Type TypeID='3' IsComplex='FALSE' >
<Name>UINT</Name>
</Type>
</Apartment>
```

The other properties of the data type definition should be left out.

8. Functions

On exporting/importing functions the linked variables, etc. are not automatically exported/imported. You have to care, that the needed objects are exported/imported before.

Information

From zenon version 6.50 and above, time information is saved in an XML based function in UTC (seconds from 01.01.1970, 00:00 am). This affects, for example, functions such as screen switching on the following screen types:

- Alarm Message List
- Alarm Message List Filter
- Archive revision
- Chronological Event List
- Chronological Event List Filter
- Extended Trend
- Production & Facility Scheduler
- Report
- Time filter

For versions of zenon older than 6.50, this information is saved on a string based format (for example 02.01.1970 01:00:00).

8.1 Scripts

The export file for the scripts has the following sections:

Script list

On exporting/importing scripts the linked functions are not automatically exported/imported. You have to care, that the needed functions are exported/imported before.

Each script has its own section in the export file containing the script's name and the names of the linked functions.

9. Menus

The export file for the menus has the following sections:

Menu list

On exporting/importing menus the linked variables, functions, help pages and macros are not automatically exported/imported. You have to take care that the needed variables or functions are exported/imported before.

10. Reaction matrices

The export file for the reaction matrices has the following sections:

Reaction matrix list

11. Standard recipes

The export file for the standard recipes has the following sections:

Recipe list
Variable list
Driver list
Data type list

The included variables are exported with the standard recipes. The recipes are imported automatically. The variables have to be imported manually before from the same file.

In addition to the XML import/export, standard recipes can also be imported in ASCII format.

Attention: With the ASCII import recipes can only be imported singly.

The non-linearized value of the recipe is saved. This value is used for VBA and for the export of recipes via XML.

12. Recipegroups

The export file for the recipes of the Recipegroup Manager has the following sections:

- Recipegroup list

On exporting/importing the recipe groups and their recipes, the linked variables are not automatically exported/imported. Take care that the needed variables are exported/imported before.

The XML file is based on the recipegroups, i.e. single recipegroups are are imported (with their variables and recipes and recipe values). After an import, existing recipe values, which are not in the import file, will still exist. The same is true for recipes and linked variables.

When importing a single recipe from the recipegroup, the following applies: The associated recipegroup must be selected. An according message will be shown in the output window.

> Information
>
> *In addition to XML, the RGM also supports ASCII import of recipes.*
>
> ***Attention:*** *With the ASCII import recipes can only be imported singly.*

13. Variables

Variables can be exported and imported with zenon :

- XML (on page 20): Export and Import; during import the files are analyzed and in case of conflicts, solutions are offered
- CSV: Export (on page 38) and Import (on page 29); during import the present data is overwritten
- DBF (on page 50): Export and Import; during import the present data is overwritten
- S7 (on page 40): Import (on page 40)

- TwinCAT Projekt (on page 50): Import

> **Information**
>
> *When importing/exporting via CSV, no complex variables (structure variables, arrays) can be imported or exported.*

EXPORT

The export file for the variables has the following sections:

- Variable list
- Driver list
- Data type list

Along with the variables the drivers and the datatypes are also exported. See also manual Variables, chapter Export data.

> Attention
>
> *A driver is absolutely necessary in the XML file, too; otherwise no variables are imported! Also be aware that the XML file driver settings are project-dependent.*

IMPORT

During Import of the variables (on page 19), zenon opens a dialogbox in which the source drivers of the XML file can be allocated to target drivers in the project. All variables of the source driver are then imported as variables of the selected target driver. The data types are automatically imported with the variables. The drivers have to be created in the project before the import. See chapter Import of variables and datatypes (on page 19).

> **Attention**
>
> Changes for variables are collected in a unsorted list. Therefore only one change per variable can be taken over per import. If several changes should be taken over, it must be done via several import processes.
>
> Example: If the same variable is renamed and deleted in the same import process, the import could lead to a not deleted but renamed variable in the project.

IMPORT STRUCTURE VARIABLES:

STRUCTURES

Structures which differ from existing ones can be imported in already existing structures. Variables based on this are automatically adapted.

- The structure elements are identified by their name.
- At already existing structure elements the type is adapted if necessary.
- Non-existing elements are added.
- Elements which do not exist in the structure data type are removed.

INACTIVE VARIABLES

At the import of structure variables, active and inactive variables are imported. Existing imports are not overwritten at the import. If an inactive variable is imported to a project and then activated, it stays active even after a new XML import.

13.1 Import and export of variables and datatypes

For the import/export of variables the following is true:

- The import/export must not be started from the global project.
- The start takes place via:
  - Context menu of variables or data typ in the project tree
  - or context menu of a variable or a data type
  - or symbol in the symbol bar variables

⚠ **Attention**

When importing/overwriting an existing data type, all variables based on the existing data type are changed.

Example:

There is a data type XYZ derived from the type `INT`with variables based on this data type. The XML file to be imported also contains a data type with the name XYZ but derived from type `STRING`. If this data type is imported, the existing data type is overwritten and the type of all variables based on it is adjusted. I.e. the variables are now no longer `INT` variables, but `STRING` variables.

13.2 XML import

When importing variables or datatypes they are analyzed by zenon according to the Driver allocation (on page 19). The user can subsequently decide what to do with the according variables / datatypes:

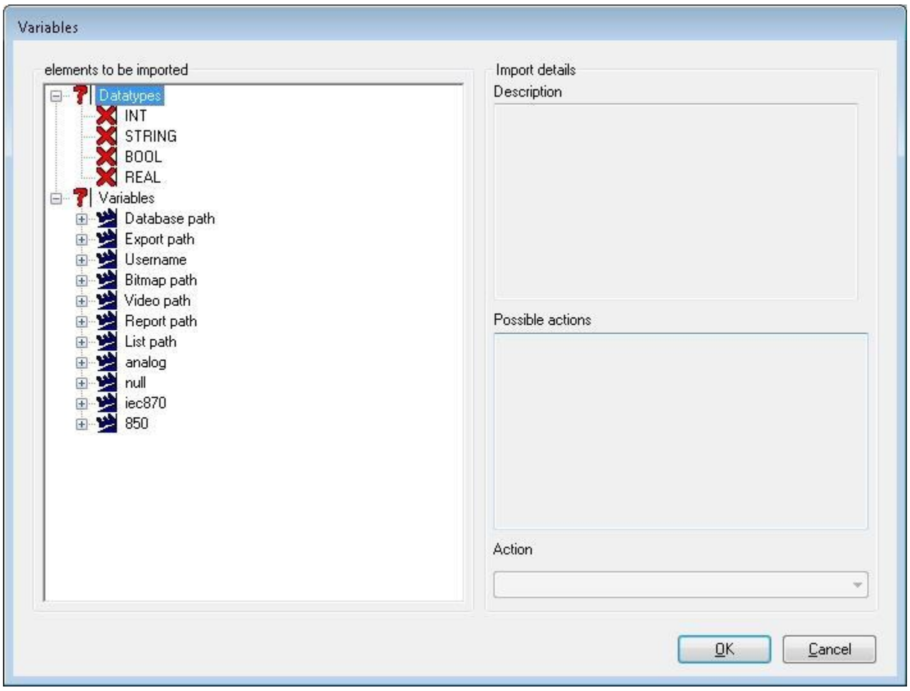
- import
- overwrite
- do not import

Only reasonable actions are offered.

The following conflicts prevent the import:

- The data type already exists and the linked variable cannot be converted to the data type you want to import.
- The data type is complex and the elements of the existing one differ from the elements of the one to be imported in number or name.
- The data type is complex and contains the same elements, but the elements differ in their data types, so that a conversion of the linked variables is not possible.
- The variable type cannot imported (see above).
- The variable type differs from the existing one (simple/complex).

- The driver assignment for the variable is incorrect (the assigned driver does not support the according datatype and/ot the respective area).



Parameters	Description
Elements to be imported	Lists all variables and data types that are to be imported. For complex variables, the according simple variables are shown. Structured types can be expanded, the elements of structure datatypes are displayed for information purposes. Symbols show if the element can be imported (green check mark), cannot be imported (red X) or if a conflict must be solved (blue symbol).
Import details:	Describes import status and offers solutions for conflict resolution.
Description	Describes the element status.
Available actions	Describes the available actions.
Action	Drop-down list with the available actions and the currently selected action. If only one action is possible, it is preselected and cannot be changed.
Import new data type / variable	The variable / data type does not exist yet. It is created by the import. (Default) Cannot be changed by the user.
Do not import data type/variable	The variable/data type already exists, there are conflicts. Cannot be changed by the user.
Import data type/variable	The variable/data type already exists, there are no conflicts. The user can decide whether the variable/data type should be overwritten or not imported. Per default, the variable/data type is overwritten.

LIMITATIONS

- Consistency

The XML file to be imported has to be consistent. There is no plausibility check on importing the file. If there are errors in the import file, this can lead to undesirable effects in the project, and even crashes.

You particularly have to care for this, if not all properties exist in the XML file and thus are filled with default values. For example: A binary variable has a limit of 300.

- Backward compatibility

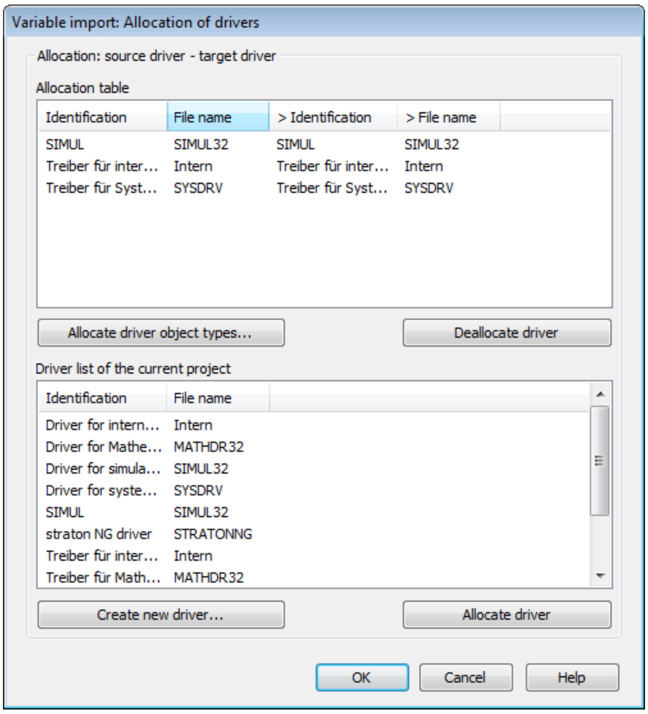
At the XML import/export there is no backward compatibility. Data from older zenon versions cannot be taken over. The handover of data from newer to older versions is not supported.

- Structure data types

Structure data types must have the same number of structure elements. If, for example, a structure data type in the project has three structure elements, a data type with the same name in the XML file but four structure elements, none of the variables based on this data type from the export file are imported into the project.

13.2.1 Allocate driver

The command `Import` opens the dialog for driver assignment:



Variable import: Allocation of drivers

Allocation: source driver - target driver

Allocation table

Identification	File name	> Identification	> File name
SIMUL	SIMUL32	SIMUL	SIMUL32
Treiber für inter...	Intern	Treiber für inter...	Intern
Treiber für Syst...	SYSDRV	Treiber für Syst...	SYSDRV

Driver list of the current project

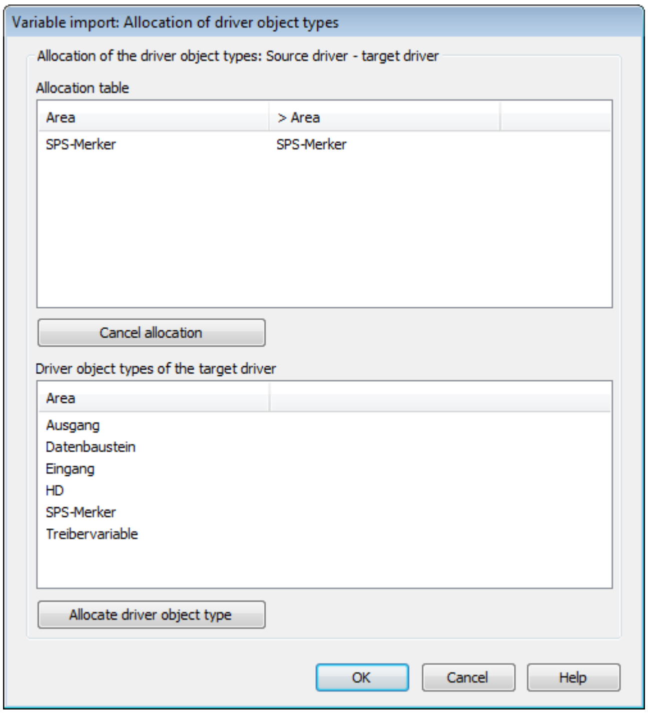
Identification	File name
Driver for intern...	Intern
Driver for Mathe...	MATHDR32
Driver for simula...	SIMUL32
Driver for syste...	SYSDRV
SIMUL	SIMUL32
straton NG driver	STRATONNG
Treiber für inter...	Intern
Treiber für Math...	MATHDR32

Parameters	Description
Allocation table	Displays the current allocation.
Allocate driver object type	Opens the dialog for allocating the driver object types (on page 25).
Cancel driver allocation	Resolves existing allocations.
Driver list	Lists all existing dirvers of the project.
Create new driver	Open dialog in order to create a new driver.
Allocate driver	Allocates the selected driver from the driver list to the driver selected from the allocation table.
OK	Applies settings, closes the dialog and starts the import.
Cancel	Discards settings and closes the dialog without import.
Help	Opens online help.

CREATE NEW DRIVER

During import, you can create and allocate new drivers by using the button **New driver** In the process, the dialog for Creating and Configuring new drivers is opened.

13.2.2 Assign area


Variable import: Allocation of driver object types
Allocation of the driver object types: Source driver - target driver
Allocation table
Area
> Area
Cancel allocation
Driver object types of the target driver
Area
Allocate driver object type
OK
Cancel
Help

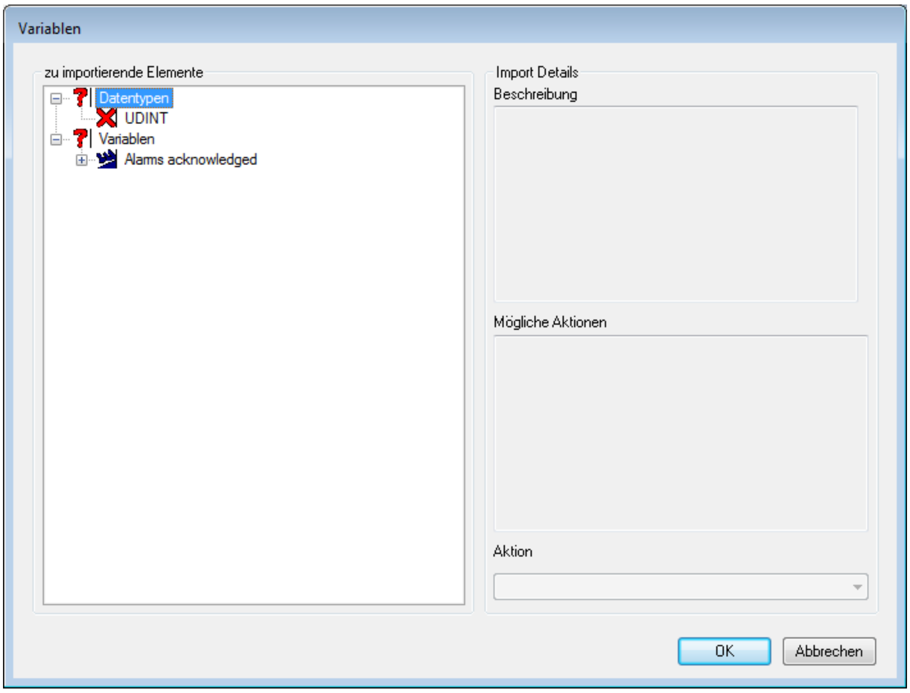
Area	> Area
SPS-Merker	SPS-Merker

Area
Ausgang
Datenbaustein
Eingang
HD
SPS-Merker
Treibervariable

Parameters	Description
`Allocation table`	Displays the current allocation.
`Cancel allocation`	Resolves existing allocations.
`Driver object type of the target driver`	List of the available drive object types.
`Allocate driver object type`	Allocates the selected driver object type of the target driver to the selected driver object type in the allocation table.
`OK`	Applies settings and closes the dialog.
`Cancel`	Discards settings and closes the dialog.
`Help`	Opens online help.

13.2.3 Troubleshooting import

If during the variable import incompatibilities are noticed, the dialog for error treatment is opened.



Parameters	Description
Elements to be imported	List of elements which should be imported. Faulty elements are highlighted.
Import details	Details about the import.
Description	Error description.
Available actions	Description of the actions which can be set for this error.
Actions	Drop-down list for the selection of the desired action.
OK	Take over actions, closes the dialog and imports variables.
Cancel	Discards the entry and cancels the import of the variables.

13.2.4 Manual editing of the variables XML file

XML-files can be edited after export and/or before the import with an external program.

⚠ **Attention**

External editing of XML files is recommended `only for experts`. *Incorrect imports can damage or destroy your project. Therefore always save your project before you edit it.*

The header definitions must contain the following attributes:

Parameters	Description
ShortName	Name of the variable
DriverID	Driver ID Please consider that the identification of every driver is project-dependent. You can identify the driver exactly by creating a driver and an associated variable in your project and exporting it via XML.
TypeID	ID for the data type CAUTION: the TYPEID also depends on the project. Details can be found in Import for data types.
HWObjectType	Identification of the driver object type The area on the PLC is defined here, e.g. data block area, marker area, input, output, etc. Please consider that not every driver supports all object types. Please look at the documentation of the according driver to find out which driver supports which object types. The correct identifications are listed in the category tpKanaltypes in the VBA object catalogue.
HWObjectName	Name for the driver object type Redundant information for the HWObjectType. This property only serves for information purposes.

IsComplex	TRUE = Structure variable
	FALSE = simple variable

Matrix	Name of the linked reaction matrix This property is empty if no reaction matrix is linked.

SYNTAX EXAMPLES FOR EDITING THE XML FILE:

No.	Example	Expression	Comment
1	Creating a new variable during import	<Variable ShortName='WIZ_VAR_10' DriverID='4' TypeID='2' HWObjectType='8' HWObjectName='PLC marker' IsComplex='FALSE' Matrix=''>	
2.	Deleting an existing variable during import	<Variable ShortName='WIZ_VAR_10' Delete='TRUE' DriverID='4' TypeID='2' HWObjectType='8'	If the variable existed in the zenon project, you can delete it by using the attribute Delete. Please make sure that the attribute is not empty. This means it must contain e.g. TRUE

		HWObjectName='PLC marker' IsComplex='FALSE' Matrix='>>	or `FALSE`. The content of the entry is not evaluated in this case.
`3.`	`Renaming an existing variable during import`	<Variable ShortName='WIZ_VAR_10' Delete='TRUE' DriverID='4' TypeID='2' HWObjectType='8' HWObjectName='SPS-Merker' IsComplex='FALSE' Matrix='' `NewName='Standard_1'>`	When renaming, the variable is first imported and then renamed. This means you can change the variable´s properties and rename it at the same time. If there is already a variable with the new name, renaming fails. The output window provides you with further information.

13.3 CSV import

Variables can be exported (on page 38)with zenon to CSV files and imported from CSV files, for example for further editing in Microsoft Excel. Variables exported to CSV are always saved as TXT files in unicode format.

If the compatibility limit of Microsoft Excel was surpassed during the export and the export was executed in spite of a warning, the TXT file to be imported doesn´t contain all exported data.

FORMAT SPECIFICATION

- Separator: `Tab`
- File ending: `.txt`
- Encoding: The exported file is encoded in unicode. The file to be imported has to be encoded in unicode, too.

> **Information**
>
> *Import and Export via CSV or dBase supported; no driver specific variable settings, such as formulas. Use export/import via XML for this.*

COMPATIBILITY WITH MICROSOFT EXCEL

EXPORT TO EXCEL

For proper import of exported variables in CSV files in Microsoft Excel, please mind:

- Format:

  In order for Microsoft Excel to use the proper decimal seperator, you have to set the format in your operating system´s `Regional and language options` to `English`. **Attention:** This setting is mandatory for the usage of decimal seperators, otherwise the values are interpreted/saved incorrectly.

- Unicode:

  The files must be saved in Microsoft Excel as `Unicode text (.txt)`. They are saved as TXT files. In older versions of Microsoft Excel, the description can differ slightly.

  Attention: When using a different format, the data cannot be imported properly!

- @ sign:

  Microsoft Excel interprets @ signs in a cell not as text and therefore might not allow editing. You have to change the cell type (category) from `Standard` to `Text` for editing cells with @ signs. For that

  - mark the cell or the column
  - right-click on the cell and choose the command **`Format cells`** in the context menu.
  - Switch the category in the dialog to `Text`

- Maximum values:

  - Microsoft Excel 97-2003: 65536 lines, 256 columns
  - Microsoft Excel 2007: 1048576 lines, 16384 columns

> **Attention**
>
> *If the number of 16384 or 256 columns and 1048576 or 65536 lines is exceeded, a dialog pops up and you can select to export correspondingly less dynamic properties (if possible). If you choose the limitation, as many dynamical properties as possible are exported.*
>
> *Statistical properties are not limited.*
>
> *If removing the dynamical properties is not sufficient, an additional message pops up.*
>
> *For the maximum number of variables you have to take into account that the header needs its own line, too.*

IMPORT FROM EXCEL

At the import from Excel you must not delete the last (empty) line. Otherwise the CSV file cannot be imported to zenon.

13.3.1 Minimum requirements

DEFINITION OF FORMATS

Format	Contents
VariableName	Name the variable shall bear
DriverName	Description of available driver
DriverType	Type of available driver
HWObjectType	Number of driver `Object type`, the name depends on the language. Is also described as `driver object type` and can be found out either through export of an according variable or with the help of the driver documentation.
TypeName	Name of the available `Data type`

CONTENTS

- Driver of imported DRIVERTYPE and imported DRIVERNAME must exist:
  If no driver with the DRIVERTYPE and the DRIVERNAME can be found, the next step is to try to link the variable with a similar driver. First, only the DRIVERTYPE, then only the DRIVERNAME is looked for. If no driver for an allocation can be found, the next step is to try to link the variable with the `Internal driver`, then with an existing driver. If no driver is available or the variable can´t be allocated to a driver, the variable is not imported. The dialog driver allocation (see chapter Import of variables and data types (on page 19)) allows to create and/or change the relevant link before the variable is imported. This dialog can be opened after the file you want to import has been selected.
- Simple data type with the imported TYPENAME and the imported HWOBJECTTYPE:
  If the data type with the relevant TYPENAME is not available, the variable is not imported. After

all further variables have been executed/imported, the output window displays how many variables could not be imported because of missing data types.

HEADER

A header must exist to describe the property values:
The Header describes the properties based on columns. The first entry in the header is always valid for the first values entry, the second entry in the header is always valid for the second values entry and so on. The number of fields in the header must correspond with the maximum number of fields of the whole file.

- The header
  need not be located in the first line of the file, but it must be the first/top line of all lines.
- The case
  - is not important for the header.
  - is considered for the property values.
- The header must be concise and complete.
  If an empty field (area between separators) is located in the header, the import is cancelled.
  Definition `empty field`: Two successive separators without content in between.
- Leading and terminating spaces
  in a field (no matter if with or without quotation marks) are interpreted as such. No spaces are removed. Therefore, the value/field must be precise.
- The separator
  should not be used at the end of a line, with one exception: An existing field with an empty value.
- All fields (area between separators) where the separator appears must be in quotation marks. That affects only the according field. The remaining fields that do not meet these criteria do not have to be in quotation marks. Tabulator-separators are automatically removed in a field in quotation marks.
- If quotation marks are to be used,
  the whole field (area between separators) must be within quotation marks and each intentional quotation must also be directly doubled. Thus, the number of quotation marks must always be even.
  Example:

  - T"e"st becomes "T""e""st",
  - "AB"C" becomes """AB""C""",
  - AB"C becomes "AB""C"

- Only two quotation marks in a field without text in between them are not valid for empty values.
- Empty fields (2 successive separators)
  are interpreted as a non-existing entry for this column. If the field of a property that exists for the variable is empty and therefore not set the standard value of this property is used.
- If less fields exist for values than determined in the header,
  the remaining non-existing fields are interpreted for values as empty fields.
- If the number of values/fields
  in a line is greater than the number of fields determined in the header, the last/affected values/fields are ignored.
- Semicolon and comma
  can be freely used in fields (without quotation marks).
- Points
  are always considered decimal separators for numeric values.
- Empty lines are allowed.
  They are ignored.

NAMES, LINKS AND REPLACEMENTS

- Every variable must have its own name in the file.
  If multiple variables with the same name exist, then the first one is imported and the others are ignored.
- Links must exist.
  Reaction matrix, functions, variables, alarm areas, alarm/event groups, alarm/event classes, interlocking and so on are only linked during import if they exist in the project. If the element does not exist under the name, the element is not linked. If a property cannot be linked, the remaining properties are nevertheless imported, regardless of that fact.
- Dynamic properties (like for example limits) have a common identification and a sequence number.

The sequence number begins with 0. The description is similar to the XML export format. (for example: Limits_8_Active, Limits_8_Text, Limits_8_LimitValue, ...)

- array variables and variables of complex data types are neither exported nor imported.
- Available complex variables are not overwritten during the installation.
- The import of a variable is allowed if the same name exists in a child element of an existing complex variable. If e.g. the name of the variable you want to import is ABC.DEF and the variable DEF in the zenon Editor is a child of variable ABC in the Editor, the variable is inserted nevertheless.
- All properties that are imported or exported have similar column/header names as in the XML import or export. The column/header names are found out the easiest way by exporting an according variable.
- Some drivers have driver-specific properties. To find the according column/header names, exporting a variable with the corresponding driver is the most suitable solution.

LIMITATIONS

Properties that are not exported or imported:

- ID_ComplexVariable
- ID_Complex
- HWObjectName
- ID_DriverTyp
- DriverID
- TypeID
- IsComplex
- All IsLocal or Flag properties
- IsSWProtokol, IsSW_Akt and IsSW_VBA

13.3.2 Delete or rename variable during import

You can delete or renamed variables during the import. The deleting is carried out via an entry in field `KANAL_D`. The renaming is carried out via an entry in field `KANAL_R`. The variable which should be deleted or renamed is defined in filed `VariableName`.

DELETE VARIABLES

Variables are deleted if valid entries are detected in filed `KANAL_D` and in field `VariableName` of the CSV file.

Field	Function	valif if
KANAL_D	Sets delete command.	1
VariableName	Name of the variable which should be deleted	Entry which does not consists solely of unprintable characters

For deleting you can use CSV files which only consist of these two fields (Short-Format).

RENAME VARIABLE

Variables are renamed if valid entries are detected in filed `KANAL_R` and in field `VariableName` of the CSV file.

Field	Function	valif if
KANAL_R	Original name of the variable which should be replaced.	Entry which does not consists solely of unprintable characters
VariableName	New variable name.	Entry which does not consists solely of unprintable characters

For renaming you can use CSV files which only consist of these two fields (Short-Format).

⚠ **Attention**

If the renaming and the deleting is commanded in the same line, the variable is deleted.

MESSAGES IN THE OUTPUT WINDOW

If variables are deleted or renamed during the CSV import, messages are displayed in the output window of the Editor stating the success or the failure of the actions:

Message	Level	Description
`Rename variable '%s' does not exist.`	Warning	The variable which should be renamed and is referred to by the variable name stated in field `KANAL_R` does not exist.
`Variable '%s' which should be deleted does not exist.`	Warning	The variable which should be deleted and is referred to by the variable name stated in field `VariableName` does not exist.
`Variable '%s' was renamed in '%s'.`	Normal	The renaming of the variable was successful.
`Variable '%s' was deleted.`	Normal	The deleting of the variable was successful.
`%i of %i variable(s) renamed.`	Normal or Warning	This message is only displayed when renaming is carried out with valid fields, i.e. fields which are not empty `KANAL_R`. The first placeholder (`%i`) equals the number of the successful renamings. The second placeholder equals the number of the commanded renamings. If not all renamings could be carried out successfully, the output is formatted as Warning.
`%i of %i variable(s) deleted.`	Normal or Warning	This message is only displayed when deleting is carried out via valid `KANAL_D` fields (numeric value `1`). The first placeholder (`%i`) equals the number of the successful deletions. The second placeholder equals the number of the commanded deletions. If not all deletions could be carried out successfully, the output is formatted as Warning.

13.3.3 Compatibility with Microsoft Excel

EXPORT TO EXCEL

For proper import of exported variables in CSV files in Microsoft Excel, please mind:

- Format:
  In order for Microsoft Excel to use the proper decimal seperator, you have to set the format in your operating system´s `Regional and language options` to `English`. **Attention:** This setting is mandatory for the usage of decimal seperators, otherwise the values are interpreted/saved incorrectly.

- Unicode:

  The files must be saved in Microsoft Excel as `Unicode text (.txt)`. They are saved as TXT files. In older versions of Microsoft Excel, the description can differ slightly.

  Attention: When using a different format, the data cannot be imported properly!

- @ sign:

  Microsoft Excel interprets @ signs in a cell not as text and therefore might not allow editing. You have to change the cell type (category) from `Standard` to `Text` for editing cells with @ signs. For that

  - mark the cell or the column
  - right-click on the cell and choose the command **`Format cells`** in the context menu.
  - Switch the category in the dialog to `Text`

- Maximum values:

  - Microsoft Excel 97-2003: 65536 lines, 256 columns
  - Microsoft Excel 2007: 1048576 lines, 16384 columns

⚠ **Attention**

If the number of 16384 or 256 columns and 1048576 or 65536 lines is exceeded, a dialog pops up and you can select to export correspondingly less dynamic properties (if possible). If you choose the limitation, as many dynamical properties as possible are exported.

Statistical properties are not limited.

If removing the dynamical properties is not sufficient, an additional message pops up.

For the maximum number of variables you have to take into account that the header needs its own line, too.

IMPORT FROM EXCEL

At the import from Excel you must not delete the last (empty) line. Otherwise the CSV file cannot be imported to zenon.

13.3.4 Example for import

For the import of a variable, the following information must be provided in the file:

VariableName	DriverName	DriverType	HWObjectType	TypeName
Test	MyDriver	Intern	33	Mydata type

Before you can import the file, you have to ensure that a driver called "`MyDriver`" type `Internal` has been created in the project. This driver must support the channel type / HWObjectType. In addition, the data type "`MyData type`" must exist. After you selected the file for import, a dialog for driver assignment pops up. In this dialog, all variables to be imported of a driver created in the project can be linked with another driver. If you want to use the same driver, you can confirm the dialog directly with OK. Then the variables are imported.

13.4 CSV export

Variables can be exported with zenon to CSV files and imported (on page 29) from CSV files, for example for further editing in Microsoft Excel.

Exception: Complex data types and array variables must not be exported.

⚠ **Attention**

The data is exported to a tabulator-seperated unicode text file with the ending `.txt` *instead of* `.csv`.

Background: *Microsoft Excel saves CSV-files as ANSI text files with* `;` *as seperator. If saved as CSV in Excel, the unicode is removed. That would irrepabably damage the export data for the project.*

EXPORT OPTIONS

1. Export of all exportable variables in the project
2. Export of selected exportable variables
   - only available if at least one exportable variable has been selected

During export of the variables, a dialog pops up where you can choose the save location and the file name. The file ending is predefined `.txt`.

If there are too many properties in the file that you want to export to and edit in Microsoft Excel, you are offered several possibilities:

- Cut off properties or lines:
  Only part of, for example, the limits or variables is being exported (that depends on the limitation). Primarily information that cannot be handled by older Microsoft Excel versions is not exported. The exported file can be edited with an external program.
- Export all properties
- Cancel export

COMPATIBILITY WITH MICROSOFT EXCEL

EXPORT TO EXCEL

For proper import of exported variables in CSV files in Microsoft Excel, please mind:

- Format:
  In order for Microsoft Excel to use the proper decimal seperator, you have to set the format in your operating system´s `Regional and language options` to `English`. **Attention:** This setting is mandatory for the usage of decimal seperators, otherwise the values are interpreted/saved incorrectly.
- Unicode:
  The files must be saved in Microsoft Excel as `Unicode text (.txt)`. They are saved as TXT files. In older versions of Microsoft Excel, the description can differ slightly.
  Attention: When using a different format, the data cannot be imported properly!
- @ sign:
  Microsoft Excel interprets @ signs in a cell not as text and therefore might not allow editing. You have to change the cell type (category) from `Standard` to `Text` for editing cells with @ signs. For that
  - mark the cell or the column
  - right-click on the cell and choose the command **`Format cells`** in the context menu.
  - Switch the category in the dialog to `Text`
- Maximum values:
  - Microsoft Excel 97-2003: 65536 lines, 256 columns
  - Microsoft Excel 2007: 1048576 lines, 16384 columns

⚠ **Attention**

If the number of 16384 or 256 columns and 1048576 or 65536 lines is exceeded, a dialog pops up and you can select to export correspondingly less dynamic properties (if possible). If you choose the limitation, as many dynamical properties as possible are exported.

Statistical properties are not limited.

If removing the dynamical properties is not sufficient, an additional message pops up.

For the maximum number of variables you have to take into account that the header needs its own line, too.

IMPORT FROM EXCEL

At the import from Excel you must not delete the last (empty) line. Otherwise the CSV file cannot be imported to zenon.

13.5 S7-project

With zenon, S7 projects can be imported for Step 7 and TIA. In this documentation, the term Step 7 is used synonymously with TIA for the sake of simplicity.

REQUIREMENTS

To use S7 projects in zenon, you can:

- Use automatic import for versions up to **`Step 7 Version 11`**
- Import variables with **`Step 7 Version 11`** or later via the **`Variable Export Manager`** tool from Step 7 and then import them to zenon via automatic import

The variables you need are created in zenon during the import. It is possible to use the WinCC name. zenon or WinCC- or Combined names can be selected. For WinCC names, it is possible to select technological steps and specify a separator.

> **Information**
>
> *The S7 import does not support structures. Structure information from the S7 project is resolved and imported as a flat list. Other drivers also offer the option to read in variables from the control program (such as OPC, Codesys, Bernecker and Rainer, ...).*

13.5.1 General

To be able to import an S7 project:

- it must be available locally on the hard disk or in the network
- it must not be opened by a Step7-program at the same time.
- everything must be translated in Step 7 or PC S7 so that all objects can be taken over in zenon
- From Step 7 Version 11, the variables must be exported with the `Variable Export Manager` and the export file must be present on the import computer

> **Information**
>
> *For the import of PCS7-projects, the subdirectory CSV_export is needed. In this subdirectory you need a CSV-file for each station of the project with the same name as the station. For the import, the hierarchical names from this file are used.*

13.5.2 Import S7 project

To import S7 projects:

1. In the zenon Editor, navigate to the 'Variables' node
2. Right click on the 'Variables' node
3. Select, in the context menu, the *Extended Import/Export -> Import S7 Project* command
4. Select the desired driver from the selection dialog

5. Click on the 'OK' button.

6. Follow the steps in the import assistant

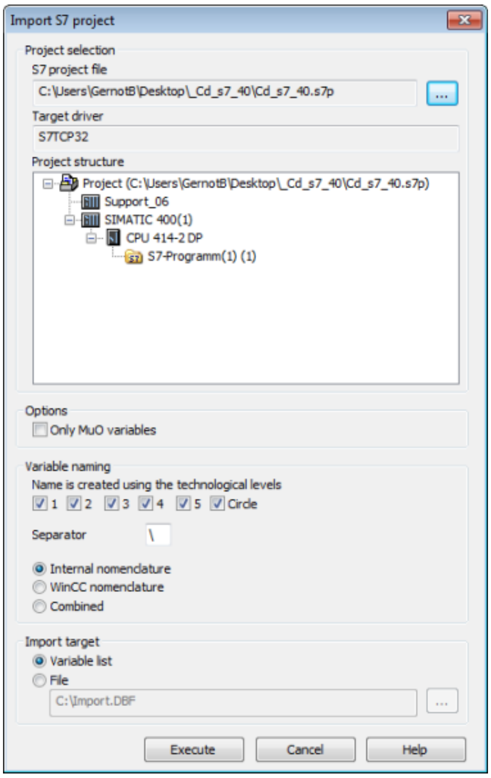
   a) Select the desired project

   b) Select the desired stations (on page 44)

   c) Configure (on page 46) the stations

   d) Select the variables (on page 48)

IMPORT DIALOG



Import S7 project

Project selection

S7 project file: C:\Users\GernotB\Desktop_Cd_s7_40\Cd_s7_40.s7p

Target driver: S7TCP32

Project structure:

```
Project (C:\Users\GernotB\Desktop\_Cd_s7_40\Cd_s7_40.s7p)
    Support_06
    SIMATIC 400(1)
        CPU 414-2 DP
            S7-Programm(1) (1)
```

Options

☐ Only MuO variables

Variable naming

Name is created using the technological levels

☑ 1 ☑ 2 ☑ 3 ☑ 4 ☑ 5 ☑ Circle

Separator: \

● Internal nomenclature
○ WinCC nomenclature
○ Combined

Import target

● Variable list
○ File: C:\Import.DBF

The following settings are available.

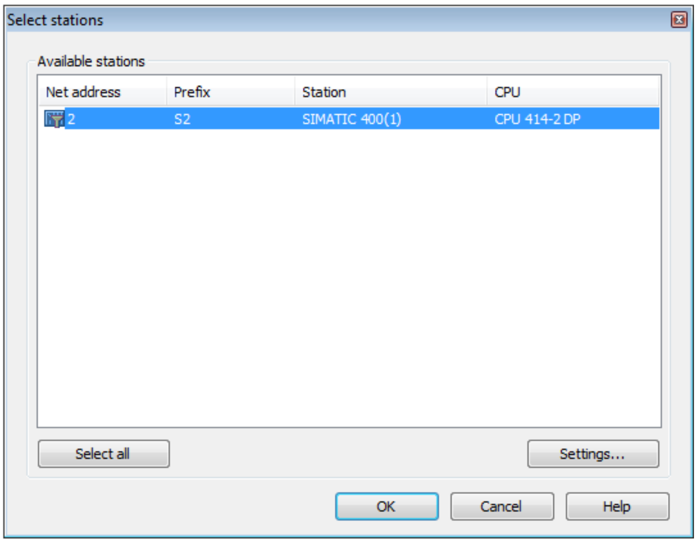
Parameters	Description
Project selection	
S7 project file	Select the desired project file (*.s7p).
Target driver	Here, you can see the driver that you selected before once again. Changes are no longer possible here.
Project structure	As soon as you have selected a project file, the structure of the selected project is shown.
Options	
Only MuO variables	Activate these options if you want to import only variables that are marked Operation and Monitoring (OaM).
Variable naming	
Name is created using technological levels	Select the technological levels (1-5 and circle) from CFC programming that are to be used for naming the variables.
Separator	Selection of the character that separates the selected technological levels from one another.
Internal nomenclature	The internal nomenclature is used for the names of the created objects. Example: **S0** (for net address = 0)
WinCC nomenclature	The WinCC nomenclature is used for the names of the created objects. Example: **S7 program**
Combined	Both the internal nomenclature and the WinCC nomenclature is used for the names of the created objects. A dot (.) is used as a separator.
Import target	
Variable list	The variables are imported in the zenon variable list. The program checks during import if a variable with the same name already exists. If that is the case, zenon-specific attributes are retained, that means the imported variables and the existing ones are merged.
File	The variables are imported into the specified file (***.dbf**). You can edit this file and import it into zenon. **Note:** Note the limited field sizes of the DBF file. If the field sizes are exceeded, information can get lost during the import.

Parameters	Description
`Execute`	Starts the import. The dialog for selecting the stations (on page 44) is opened.
`Cancel`	Discards all changes and closes the dialog.
`Help`	Opens online help.

> **Information**
>
> *The settings you make in this dialog are saved. When you open the dialog again, all settings you made are displayed again.*

13.5.3 Select stations

Select the desired stations.



Select stations

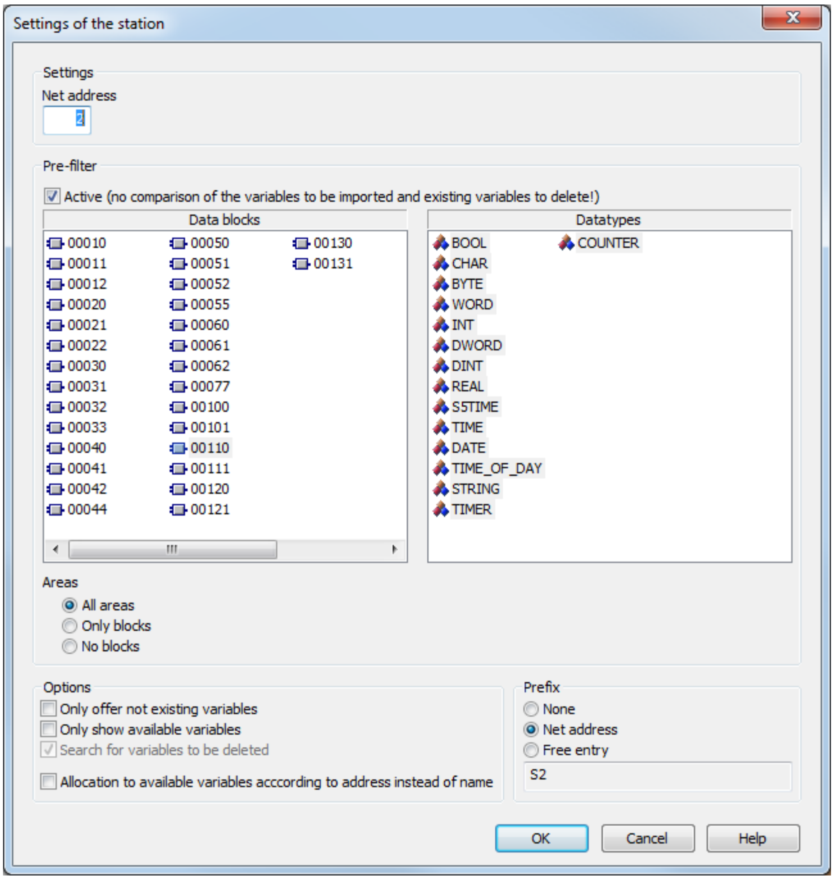
Available stations

Net address	Prefix	Station	CPU
2	S2	SIMATIC 400(1)	CPU 414-2 DP

Parameters	Description
List of stations	Lists all available stations of the selected project file. Selection via mouse click. Multiple selection is possible.
Select all	Selects all projects.
Options	Opens the dialog to configure the stations (on page 46).
OK	Accepts the selection and opens the dialog to select variables (on page 48). Depending on the size of the project, this procedure can take some time.
Cancel	Discards all changes and closes this dialog.
Help	Opens online help.

Settings of the stations

In this dialog you can set filters and make various adjustments.



Settings

Net address: 2

Pre-filter

☑ Active (no comparison of the variables to be imported and existing variables to delete!)

Data blocks

00010, 00011, 00012, 00020, 00021, 00022, 00030, 00031, 00032, 00033, 00040, 00041, 00042, 00044

00050, 00051, 00052, 00055, 00060, 00061, 00062, 00077, 00100, 00101, 00110, 00111, 00120, 00121

00130, 00131

Datatypes

BOOL, CHAR, BYTE, WORD, INT, DWORD, DINT, REAL, S5TIME, TIME, DATE, TIME_OF_DAY, STRING, TIMER

COUNTER

Areas

● All areas

○ Only blocks

○ No blocks

Options

□ Only offer not existing variables

□ Only show available variables

☑ Search for variables to be deleted

□ Allocation to available variables acccording to address instead of name

Prefix

○ None

● Net address

○ Free entry

S2

The following settings are available.

Parameters	Description
Options	
Net address	States the Net address and affects the station name. **Note**: Always starts with 0.
Pre-filter	
Active	Active: uses the variables and data types displayed in the list as a prefilter. Differences between Step 7 and zenon are not displayed. Inactive: Variables and data types of Step 7 and zenon are compared during the import. Differences are displayed. You are asked if you want to delete or merge elements that are not present.
Data blocks	Select the data blocks from which you want to import the variables. You can select multiple data blocks using the ctrl and Shift keys.
Data types	Select the data types that are the basis of the variables you want to import. You can select multiple data types with the ctrl and Shift keys.
Areas	Select the desired area. The choices are: ▶ All areas ▶ Only blocks ▶ No blocks
Options	
Offer only not available variables	Only those variables are offered for import that do not exist yet in zenon.
Only show available variables	Variables that already exist are searched for. These can be merged during import.
Search for variables to be deleted	After the variable selection, all variables that are not available in the S7-project are displayed. They can be deleted from the zenon Project. Inactive, if active is activated (no comparison of variables to be imported and existing variables for deletion!).

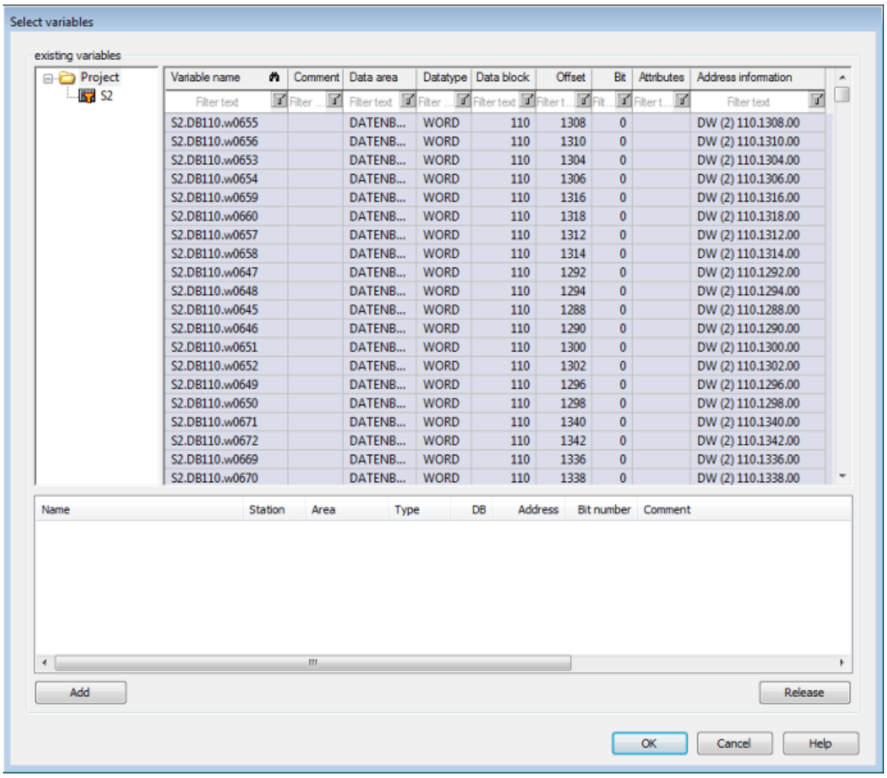
Assignment to existing variables according to address instead of name	Active: Pre-existing driver variables are merged using the variable address instead of the name. It is thus possible to accept amended names from a Step7 project.
Prefix	
None	The imported variables are not assigned prefixes in front of the name.
Net address	The imported names are assigned the net address followed by a point as prefix in front of the name.
Free entry	Enter the desired prefix in the input field. When importing a variable, the prefix entered is put in front of the variable name, with a dot as a separator. **Note**: For another import you have to use the same prefix again. If you don´t, variables are created twice.
OK	Accepts selection and closes the dialog.
Cancel	Discards all changes and closes this dialog.
Help	Opens online help.

13.5.4 Select variable

All variables available for import are shown. The following information is available for each variable:

- Variable name
- Comment
- Data area
- Data type
- Data block
- Offset
- Bit
- Attributes
- Address information.

You can sort the list for each column in ascending or descending order and filter using a filter text. The column address information contains all the the cumulated information of the respective variable.



Variable name	Comment	Data area	Datatype	Data block	Offset	Bit	Attributes	Address information
S2.DB110.w0655		DATENB...	WORD	110	1308	0		DW (2) 110.1308.00
S2.DB110.w0656		DATENB...	WORD	110	1310	0		DW (2) 110.1310.00
S2.DB110.w0653		DATENB...	WORD	110	1304	0		DW (2) 110.1304.00
S2.DB110.w0654		DATENB...	WORD	110	1306	0		DW (2) 110.1306.00
S2.DB110.w0659		DATENB...	WORD	110	1316	0		DW (2) 110.1316.00
S2.DB110.w0660		DATENB...	WORD	110	1318	0		DW (2) 110.1318.00
S2.DB110.w0657		DATENB...	WORD	110	1312	0		DW (2) 110.1312.00
S2.DB110.w0658		DATENB...	WORD	110	1314	0		DW (2) 110.1314.00
S2.DB110.w0647		DATENB...	WORD	110	1292	0		DW (2) 110.1292.00
S2.DB110.w0648		DATENB...	WORD	110	1294	0		DW (2) 110.1294.00
S2.DB110.w0645		DATENB...	WORD	110	1288	0		DW (2) 110.1288.00
S2.DB110.w0646		DATENB...	WORD	110	1290	0		DW (2) 110.1290.00
S2.DB110.w0651		DATENB...	WORD	110	1300	0		DW (2) 110.1300.00
S2.DB110.w0652		DATENB...	WORD	110	1302	0		DW (2) 110.1302.00
S2.DB110.w0649		DATENB...	WORD	110	1296	0		DW (2) 110.1296.00
S2.DB110.w0650		DATENB...	WORD	110	1298	0		DW (2) 110.1298.00
S2.DB110.w0671		DATENB...	WORD	110	1340	0		DW (2) 110.1340.00
S2.DB110.w0672		DATENB...	WORD	110	1342	0		DW (2) 110.1342.00
S2.DB110.w0669		DATENB...	WORD	110	1336	0		DW (2) 110.1336.00
S2.DB110.w0670		DATENB...	WORD	110	1338	0		DW (2) 110.1338.00

Name	Station	Area	Type	DB	Address	Bit number	Comment

You can select either the whole project or unique single stations in the tree view. The respective variables are shown depending on the selection. Select the desired variables and click the button **Add** to add the variable to the import. The selected variables are shown in the lower part of the dialog. To remove variables from there, select the desired variables and click the button **Remove**.

Info

With the help of `Ctrl` and/or `Shift` you can select more than one entry at a time.

- By pressing and holding `Ctrl` you can select a number of entries.
- By pressing and holding `Shift` and select two entriey, you select all entries which lie between the two selected entries.
- By pressing and holding both `Ctrl` and `Shift` and selecting two entries, all entries which lie between the selected entries are selected. The entries which were selected beforehand remain selected.

After finishing, click the button **OK** to close the dialog and confirm the modifications you made. The variables are created in zenon.

13.6 dBase import and export

Data can be exported to and imported from dBase.

> **Information**
>
> *Import and Export via CSV or dBase supported; no driver specific variable settings, such as formulas. Use export/import via XML for this.*

IMPORT DBF FILE

To start the import:

1. right-click on the variable list
2. in the drop-down list of **`Extended export/import...`** select the **`Import dBase`** command
3. follow the import assistant

The format of the file is described in the chapter File structure (on page 51).

> **Information**
>
> *Note:*
>
> - Driver object type and data type must be amended to the target driver in the DBF file in order for variables to be imported.
> - dBase does not support structures or arrays (complex variables) at import.

EXPORT DBF FILE

To start the export:

1. right-click on the variable list
2. in the drop-down list of **`Extended export/import...`** select the **`Export dBase...`** command
3. follow the export assistant

⚠ **Attention**

DBF files:

- must correspond to the 8.3 DOS format for filenames (8 alphanumeric characters for name, 3 character suffix, no spaces)
- must not have dots (.) in the path name.
  e.g. the path `C:\users\John.Smith\test.dbf` is invalid.
  Valid: `C:\users\JohnSmith\test.dbf`
- must be stored close to the root directory in order to fulfill the limit for file name length including path: maximum 255 characters

The format of the file is described in the chapter File structure (on page 51).

Information

dBase does not support structures or arrays (complex variables) at export.

13.6.1 dBaseIV variable file

The dBaseIV file must have the following structure and contents for variable import and export:

⚠ **Attention**

dBase does not support structures or arrays (complex variables) at export.

DBF files must:

- conform with there name to the 8.3 DOS format (8 alphanumeric characters for name, 3 characters for extension, no space)
- Be stored close to the root directory (Root)

STRUCTURE

Description	Type	Field size	Comment
KANALNAME	Char	128	Variable name. The length can be limited using the MAX_LAENGE entry in **project.ini**.
KANAL_R	C	128	The original name of a variable that is to be replaced by the new name entered under "VARIABLENNAME" (field/column must be entered manually). The length can be limited using the MAX_LAENGE entry in **project.ini**.
KANAL_D	Log	1	The variable is deleted with the 1 entry (field/column has to be created by hand).
TAGNR	C	128	Identification. The length can be limited using the MAX_LAENGE entry in **project.ini**.
Unit	C	11	Technical unit
DATENART	C	3	Data type (e.g. bit, byte, word, ...) corresponds to the data type.
KANALTYP	C	3	Memory area in the PLC (e.g. marker area, data area, ...) corresponds to the driver object type.
HWKANAL	Num	3	Bus address
BAUSTEIN	N	3	Datablock address (only for variables from the data area of the PLC)
ADDRESS	N	5	Offset

BITADR	N	2	For bit variables: bit address For byte variables: 0=lower, 8=higher byte For string variables: Length of string (max. 63 characters)
ARRAYSIZE	N	16	Number of variables in the array for index variables ATTENTION: Only the first variable is fully available. All others are only available for VBA or the Recipe Group Manager
LES_SCHR	R	1	Write-Read-Authorization 0: Not allowed to set value. 1: Allowed to set value.
MIT_ZEIT	R	1	time stamp in zenon zenon (only if supported by the driver)
OBJEKT	N	2	Driver-specific ID number of the primitive object comprises TREIBER-OBJEKTTYP and DATENTYP
SIGMIN	Float	16	Non-linearized signal - minimum (signal resolution)
SIGMAX	F	16	Non-linearized signal - maximum (signal resolution)
ANZMIN	F	16	Technical value - minimum (measuring range)
ANZMAX	F	16	Technical value - maximum (measuring range)
ANZKOMMA	N	1	Number of decimal places for the display of the values (measuring range)
UPDATERATE	F	19	Update rate for mathematics variables (in sec, one decimal possible) not used for all other variables
MEMTIEFE	N	7	Only for compatibility reasons
HDRATE	F	19	HD update rate for historical values (in sec, one decimal possible)
HDTIEFE	N	7	HD entry depth for historical values (number)
NACHSORT	R	1	HD data as postsorted values
DRRATE	F	19	Updating to the output (for zenon DDE server, in [s], one decimal possible)
HYST_PLUS	F	16	Positive hysteresis, from measuring range
HYST_MINUS	F	16	Negative hysteresis, from measuring range
PRIOR	N	16	Priority of the variable
REAMATRIZE	C	32	Allocated reaction matrix

ERSATZWERT	F	16	Substitute value, from measuring range
SOLLMIN	F	16	Minimum for set value actions, from measuring range
SOLLMAX	F	16	Maximum for set value actions, from measuring range
VOMSTANDBY	R	1	Get value from standby server; the value of the variable is not requested from the server but from the Standby Server in redundant networks
RESOURCE	C	128	Resources label. Free string for export and display in lists. The length can be limited using the MAX_LAENGE entry in **project.ini**.
ADJWVBA	R	1	Non-linear value adaption: 0: Non-linear value adaption is used 1: Non-linear value adaption is not used
ADJZENON	C	128	Linked VBA macro for reading the variable value for non-linear value adjustment.
ADJWVBA	C	128	ed VBA macro for writing the variable value for non-linear value adjustment.
ZWREMA	N	16	Linked counter REMA.
MAXGRAD	N	16	Gradient overflow for counter REMA.

Attention

When importing, the driver object type and data type must be amended to the target driver in the DBF file in order for variables to be imported.

LIMIT DEFINITION

Limit definition for limit values 1 to 4,   and status 1 bis 4:

Description	Type	Field size	Comment
AKTIV1	R	1	Limit value active (per limit value available)
GRENZWERT1	F	20	hnical value or ID number of a linked variable for a dynamic limit (see VARIABLEx) (if VARIABLEx is 1 and here it is -1, the existing variable linkage is not overwritten)
SCHWWERT1	F	16	Threshold value for limit
HYSTERESE1	F	14	Is not used
BLINKEN1	R	1	Set blink attribute
BTB1	R	1	Logging in CEL
ALARM1	R	1	Alarm
DRUCKEN1	R	1	Printer output (for CEL or Alarm)
QUITTIER1	R	1	Must be acknowledged
LOESCHE1	R	1	Must be deleted
VARIABLE1	R	1	Dyn. limit value linking the limit is defined by an absolute value (see field GRENZWERTx).
FUNC1	R	1	Functions linking
ASK_FUNC1	R	1	Execution via Alarm Message List
FUNC_NR1	N	10	ID number of the linked function (if “-1” is entered here, the existing function is not overwritten during import)
A_GRUPPE1	N	10	Alarm/event group
A_KLASSE1	N	10	Alarm/event class
MIN_MAX1	C	3	Minimum, Maximum
FARBE1	N	10	Color as Windows coding
GRENZTXT1	C	66	Limit value text
A_DELAY1	N	10	Time delay
INVISIBLE1	R	1	Invisible

Expressions in the column "Comment" refer to the expressions used in the dialog boxes for the definition of variables. For more information, see chapter Variable definition.

14. Interlockings

The export file for the interlockings has the following sections:

- Interlocking list

During export/import of the interlockings the linked variables are not automatically exported/imported. You have to watch out that the needed variables are exported/imported beforehand.

14.1 Command group

EXPORT

Command groups can be exported one by one or all together.

> **Information**
>
> *Command groups and general interlockings are not exported together!*

IMPORT

For the import of command groups the following is true:

- The relation between existing command groups and command groups to be imported is created using the name. If a command group with this name exists, it is updated.
- If there is already a general interlocking with that name, a new command group with a new name is created.
- Non-existing command groups are created.
- General interlockings existing in the import file are not imported.
- Absolute variable references are established using the name.
- New command interlockings get a unique ID that was never used before.

15. Time control

The export file for the time control has the following sections:

Time function list
Functions list

The included functions are exported with the time functions. The time functions are imported automatically. The functions have to be imported manually before from the same file.

16. Allocations

The export file for the allocations has the following sections:

Allocation list

On exporting/importing allocations the linked variables are not automatically exported/imported. Pay attention that the needed variables are exported/imported before.

17. Error messages

Error during import or export are displayed in the output window.

Error message	Possible cause and solution
Error: "The file cannot be opened as it has already been opened by another user."	File is locked for editing. User must release file before the import/export can be carried out again.
Error: "Database cannot be created or opened"	The rights for creating the field may be missing or the file is corrupt.
Error: "Variable is invalid and cannot be saved"	Variable is invalid. Check all parameters of the variable.
Error: "Variable could not be added to the database."	Error during the recording in the database.
Error: "Variable contains an invalid hardware address."	A hardware address of the variable is not correct. Correct the address before starting the import again.
Error: "Driver does not provide an object for import/export. Import/export is not carried out."	The selected driver does not have any objects which can be imported/exported.
Error: "Variable cannot be deleted as it is does not exist"	A variable which is selected for deleting does no longer exist.
Warning: "Variable with the new name already exists. Import is carried out with the old name.	A variable which should be renamed during the import is not renamed as a variable with the same name already exist. It is imported with its original name.
Warning: "Variable cannot be renamed and is newly created."	A variable which should be renamed cannot be renamed. It is

Errors during the XML import are treated and solved in an own dialog (on page 26).