zenon manual

Wizards

v.7.10

COPA·DATA
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. Wizards

In order to be able to handle recurring tasks in the engineering phase easily and expeditiously, zenon offers wizards for different fields of engineering.

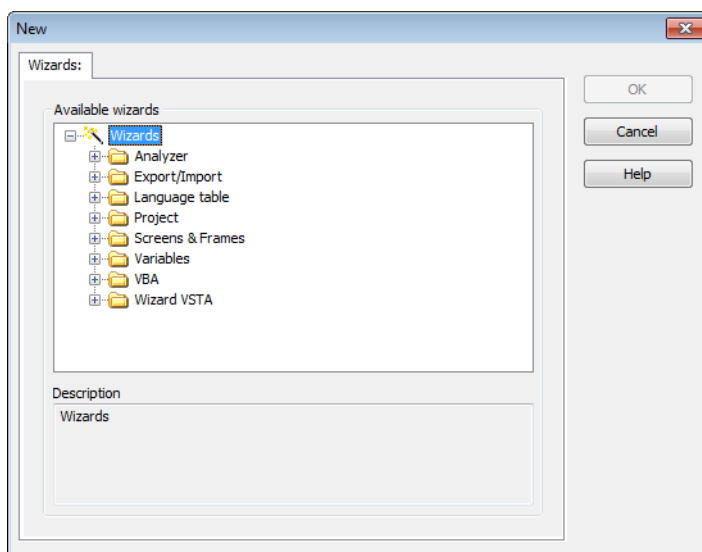Users can also create their own wizards.

License information

Part of the standard license of the Editor and Runtime.

START WIZARDS

To start a wizard:

▶ Select, in the `File` drop-down list, `Wizards ...`

or

▶ press the short cut `Alt+F12`

The wizard for project creation is automatically offered when a new project is created.

SETTINGS ZENON6.INI

For wizards to be displayed, the settings for VBA and/or VSTA must be set correctly in file `zenon6.ini`*:*

`[VBA]`

`EIN=1`

`[VSTA]`

`ON=1`

If VSTA wizards are not displayed although the settings are correct, set entry `LOADED=` *to* `1` *in area* `[VSTA].`

2.1   Topics

The following wizards are available in zenon:

- ▶ Import - Export (on page 33)
  - FactoryLink import wizard (on page 33)
  - Import Wizard (on page 67)
  - PDiag import wizard (on page 36)
  - XML export wizard (on page 62)
- ▶ Language Table (on page 67)
  - Language Translation Wizard (on page 75)
  - System Text Wizard VSTA
  - Wizard for System Texts
- ▶ Project (on page 127)
  - VBA project wizard
  - VSTA project wizard (on page 141)
  - Documentation wizard (on page 164)
- ▶ Variables and functions (on page 164)
  - Wizard for Functions
  - System Variable Creation Wizard
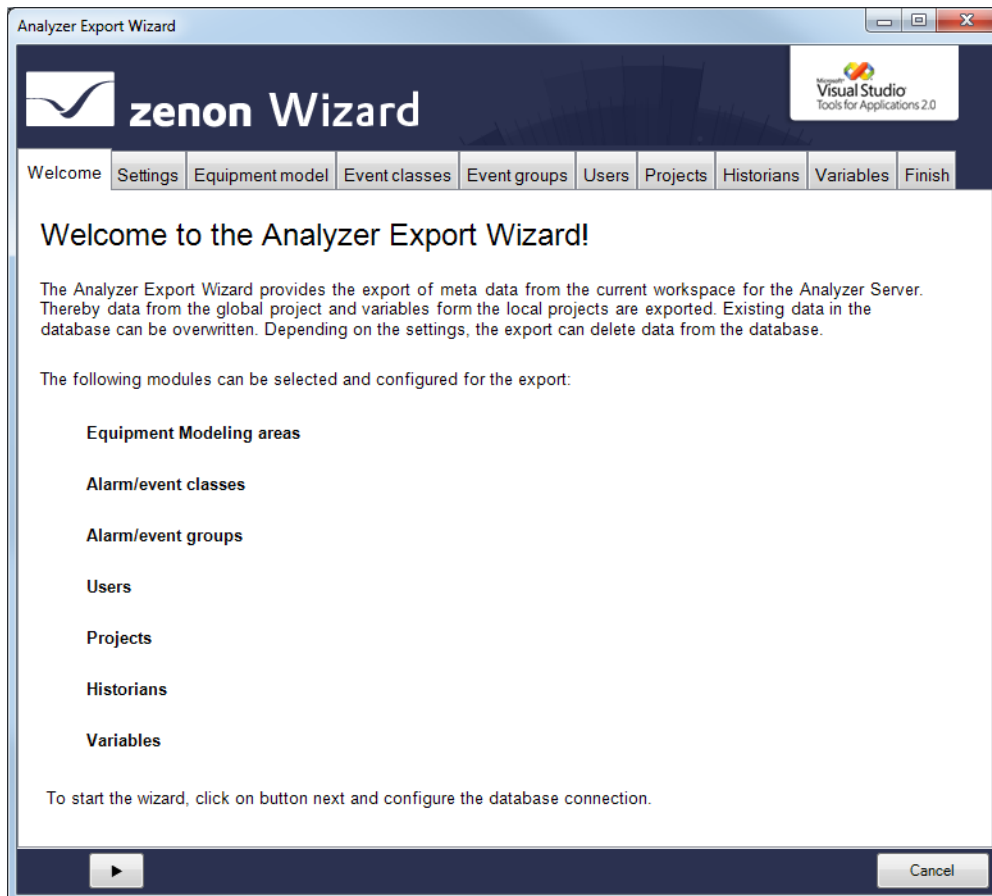 - Wizard for creating variables (on page 164)

2.1.1   Analyzer Export Wizard

The zenon Analyzer Export Wizard supports the export of metadata from zenon from version 7.0 SP0 for the zenon Analyzer.

The following can be exported:

- ▶ Data from the global project
  - equipment models
  - Alarm/event classes
  - alarm/event groups

- User

▶ Data from selected projects:

- Archives
- Variables

Note: The wizard is only available in English.

COMPATIBILITY:

The Analyzer Export Wizard works with zenon from Version 7 and with zenon 6.51 with some limitations.

Install and call up wizard

The wizard is automatically installed together with zenon.

If a manual installation is necessary:

1. Open the workspace in zenon

2. Select, in the **File** menu, the **Update wizards** command

3. In the **Additional folder for VSTA Wizards** folder, click on the `...` button

4. In the file browser, navigate to the installation medium

5. There, select the folder `[Installation medium]\Setup\Wizard\zenon7.10\WizardsVSTA`

6. Select **wizards.ini**

7. Start the update

8. The Wizard was entered into the folder of the wizard and can be started

Note: If the wizard is not displayed, add the following reference to the workspace add-in: **system.core**

CALLING UP THE WIZARD

For wizards to be displayed, the settings for VBA and/or VSTA must be set correctly in file **zenon6.ini**:
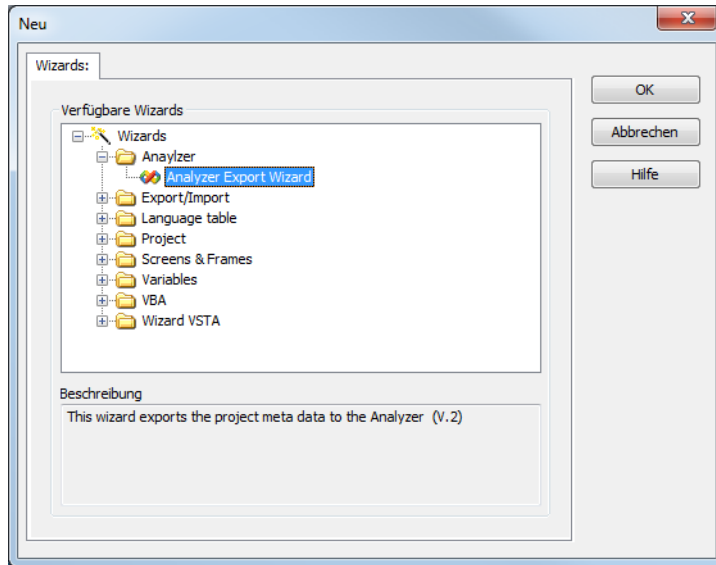
[VBA]
EIN=1
[VSTA]
ON=1

If VSTA wizards are not displayed although the settings are correct, set entry **LOADED=** to 1 in area `[VSTA]`.

To call up the wizard:

1. In zenon, open the **File** menu or press the key combination `Alt+F12`

2. Select **Wizards...**

3. The selection dialog is opened

4. Navigate to the **Analyzer** node
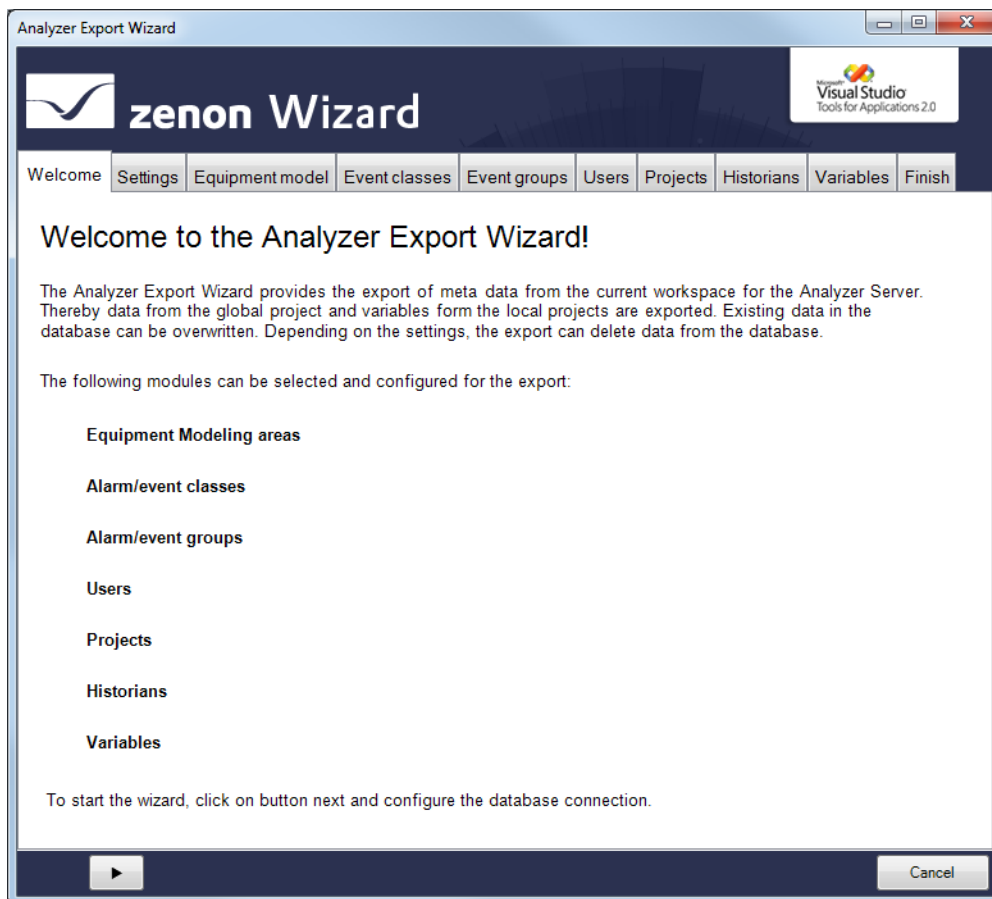
5. Select the `Analyzer Export Wizard`

6. Start the wizard by clicking on oĸ

Start window

When the wizard is opened, you receive an overview page that lists all exportable objects.

The individual objects are configured for export in their own respective tabs.

Click on the button with the `arrow` to navigate through the configuration (on page 10) of the export.

Configuration

When exporting with the Analyzer Export Wizard, all modules selected in the Settings (on page 12) tab are offered in sequence for detailed configuration. You get to the next level by clicking on the button with the `right arrow`. You can also select tabs directly by clicking on the title of the tab.

The following tabs are available for configuration of the export:

▸ Settings (on page 12): Options to collect the metadata

▸ Equipment model: (on page 16) Export of the model groups from the global project

▸ Alarm_event classes (on page 19): Alarm/Event classes from global project

▸ Alarm_event groups (on page 21): Alarm/event groups from global project

- ▶ Users (on page 23): User from global project

- ▶ Projects (on page 24): Projects from workspace

- ▶ Historian (on page 27): Archives of the selected projects

- ▶ Variables (on page 29): Variables of the selected projects

- ▶ Finish (on page 30): Start of the export and output of the result
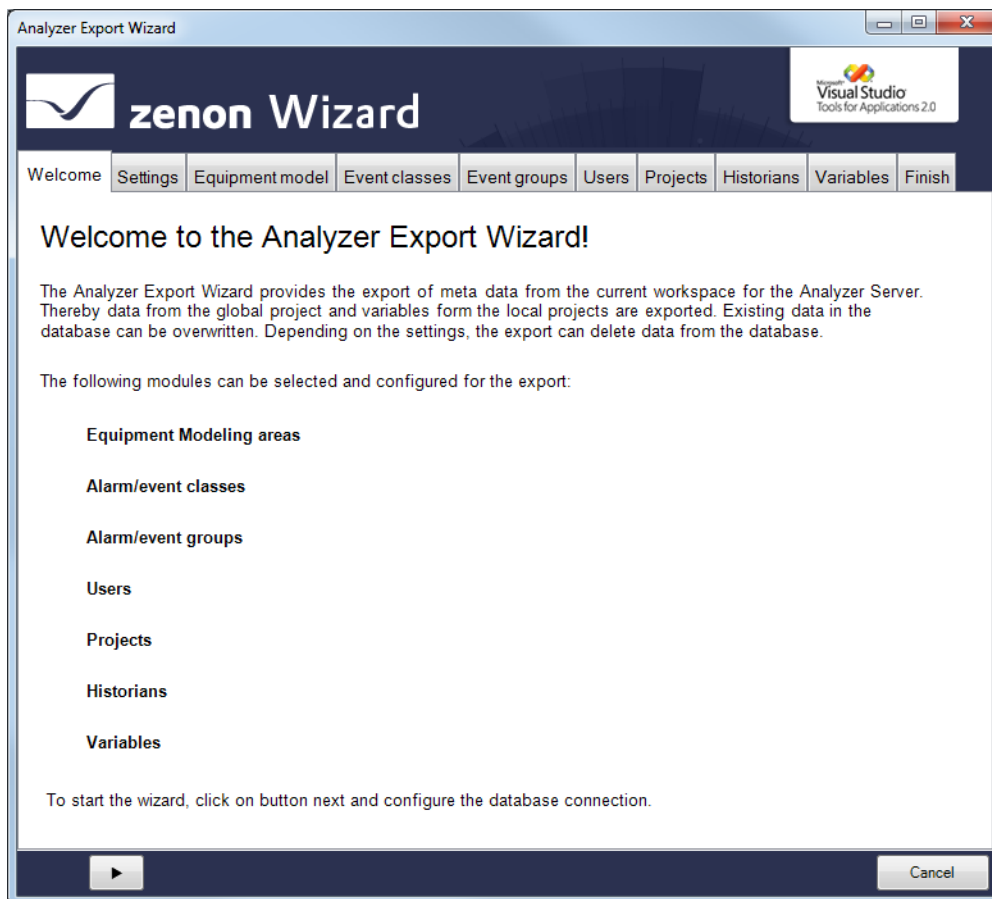
Attention: Only one global project can be exported to the database! Workspaces with projects that are to be exported to the database must include this global project.

Navigation

Navigation through the tabs is carried out by means of the navigation bar in the lower area of the wizard window:

Button	Description
`Column left`	Goes back one tab in the wizard process.
`Column right`	Goes forward one tab in the wizard process.
`Export`	Exports the data to the Analyzer database. Is only active if the `Finish` tab is opened.
`Cancel`	Closes the wizard without exporting. When closing, a dialog asks if the configuration is to be saved

▸ `Yes:` writes the settings configured in the Settings (on page 12) tab to the registry and closes the wizard; the wizard is opened with these settings the next time it is restarted

▸ `No:` closes the wizard without saving the configuration
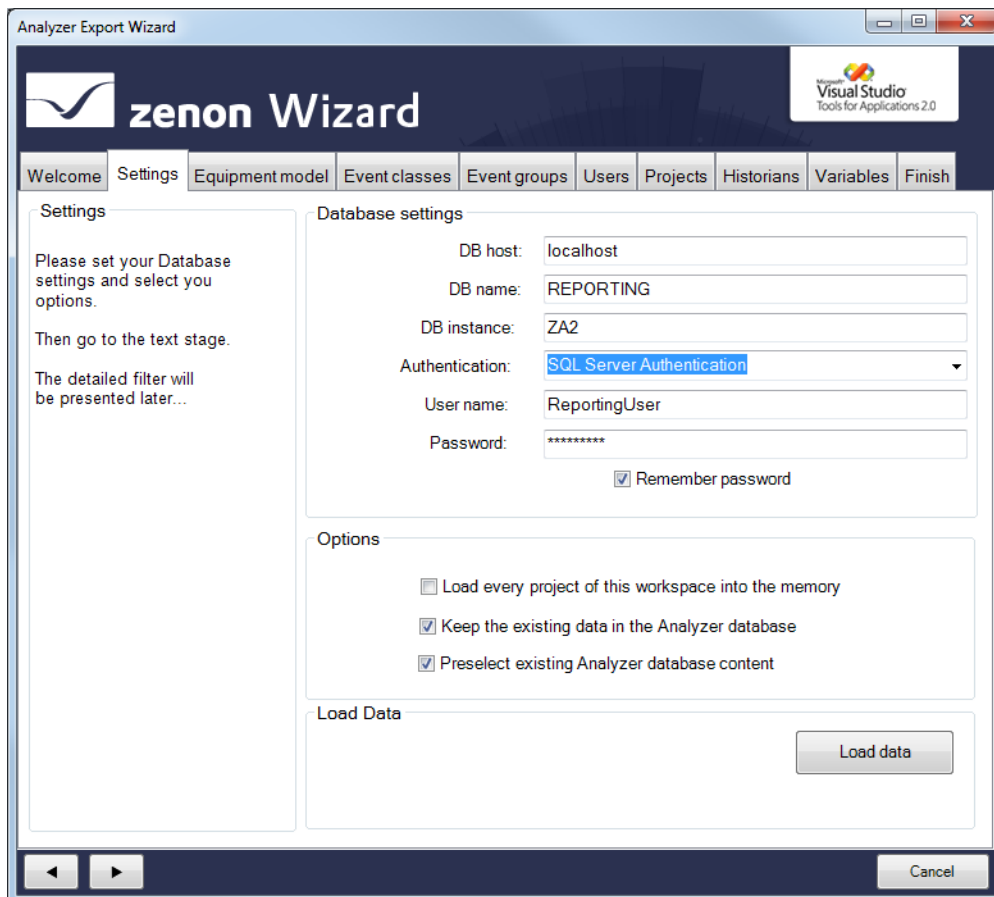
The configuration is saved for each specific user. |

Individual tabs can also be selected by clicking directly on the title of the tab.

Settings

In this tab:

1. You define the database to which the wizard connects

2. You define general options for exporting

3. You start the data readout

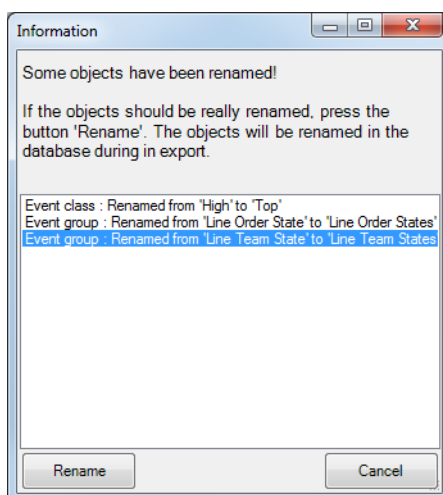| Parameters	Description
Settings	Information and notes on current export processes.
Database settings	Connection settings to the Analyzer server.
DB host	Computer on which the database is located.
DB name	Name of the database.
DB instance	Database instance
Authentication	Type of authentication: ▸ Windows Authentication: Windows login information is used. ▸ SQL Server Authentication: Login with data from an SQL server user.
User name	Entry of the user name. Only for login with SQL Server Authentication. Display only with Windows Authentication.
Password	Entry of the password. Only for login with SQL Server Authentication. No input possible with Windows Authentication.
Remember password	Password is saved for next connection. Only for login with SQL Server Authentication. Inactive for Windows Authentication.
Options	General settings for the export:
Load every project of this workspace into the memory	Active: Loads all projects present in the workspace, even if they are not active and not set to **Keep project in memory**.
Keep the existing data in the Analyzer database	Active: Only entries from the workspace are written to the database. Inactive: Entries in the database are also updated or deleted. Exception: Projects are not deleted
Preselect existing	Active: Entries already present in the database are preselected in

Analyzer database content	the individual areas.
Load Data	
Load Data	Clicking on the button loads, depending on the `Load every project of this workspace into the memory` parameter - the data from the currently loaded project into the wizard.

In doing so, a check is made to see if data is present in the Analyzer database. Pre-existing data is combined with the data from the workspace and loaded into the wizard. In the event of naming conflicts, a dialog to rectify the error is called up.

If the loading of data has been successfully concluded, the export can be configured in the following tabs. |

RENAMING OBJECTS

Objects must always be named the same in the Analyzer database and in zenon. If objects that are already present in the database are renamed in zenon, these changes can be accepted or rejected when the data is combined. Rejection of the changes leads to the wizard being closed, because only objects with identical names can be handled correctly.

DIALOG FOR RENAMING

In the event of conflicts in the naming of objects, a dialog for dealing with the error is opened:

Parameters	Description
`List of amended objects`	Contains all objects that were changed. Previous name and new name are displayed. The following renamed objects are displayed in the list: ▸ Equipment model group name ▸ Names of the alarm/event classes ▸ Names of the alarm/event groups ▸ Project name ▸ Variable name Exceptions: ▸ Users are always recreated ▸ Archive names are only created once in the database as a `visual name` and can be overwritten in the Analyzer
`Rename`	Renames all objects listed in the database, closes the dialog and stops reading in data.
`Cancel`	Leaves the previous name in the database, finishes reading in data and closes the wizard.

Equipment model
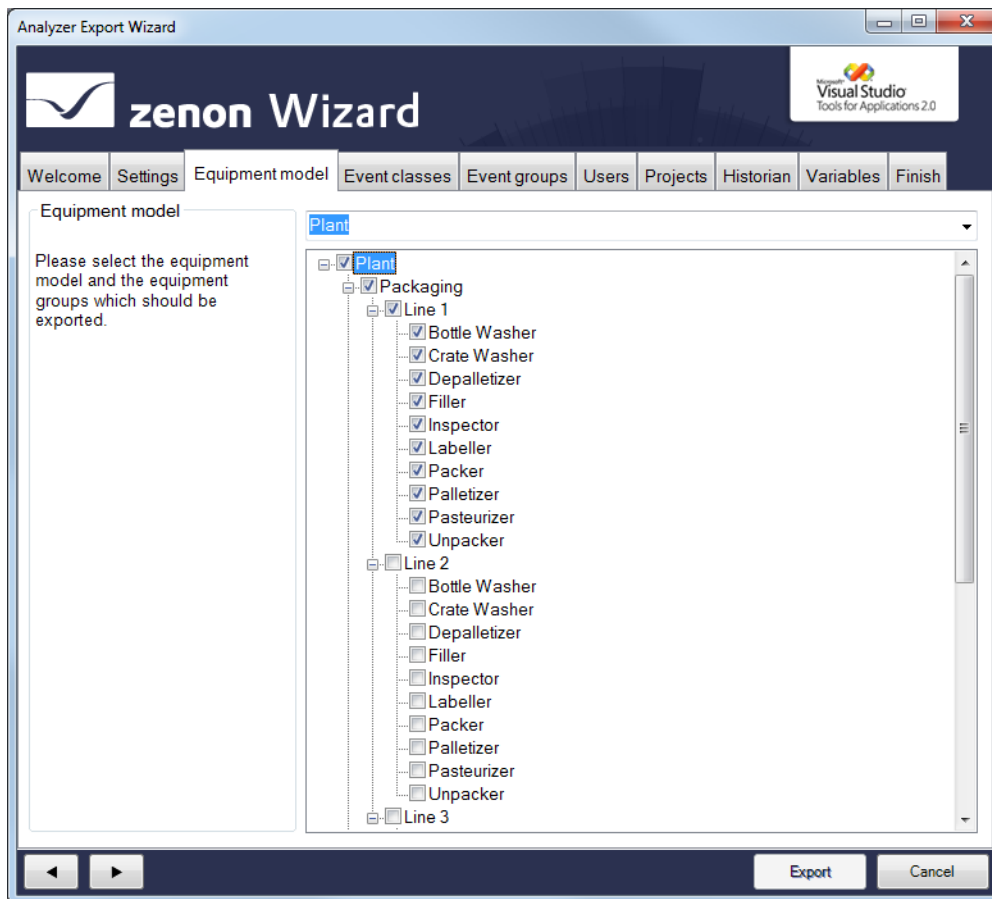
Configuration of the model groups to be exported from the global project.

> ⚠ **Attention**
>
> *Each equipment group in zenon may only be assigned to one individual time model.*
>
> *If several time model groups are assigned, the Analyzer Wizard Export uses the first that it finds and exports this to the metadata of the Analyzer. Other time model groups are ignored.*

Parameters	Description
`Equipment modeling`	Information and notes on exporting.
`Selection of equipment/medium`	Drop-down list to select what is offered in **List of equipment models/media** for configuration: ▸ `Plant`: displays equipment models ▸ `Media`: displays media
`List of equipment models/media`	List field with the possibility to select equipment models and model groups or media. To select an entry, activate the check box in fornt of the entry. In the list field, the name is always displayed in the individual nodes as it is stored in the database. If the name is changed, the original name from the zenon project is displayed in brackets. Equipment groups that were deleted in the global project are no longer displayed. If, in the **Settings** tab, the option `Keep the existing data in the Analyzer database` was selected, amended objects in the database are deleted or updated.

Alarm/event classes

Configuration of the alarm/event classes to be exported from the global project.

Parameters	Description
`Alarm/event classes`	Information and notes on exporting.
`List of the alarm/event classes`	List field with the possibility to select the alarm/event classes. To select an entry, activate the checkbox in front of the entry. `Sorting`: Clicking on the column identifier sorts the entries after this column upwards or downwards. `Multiple selection`: If several rows are highlighted, a click in the check box sets the options for all selected rows. Alarm/event classes that were deleted in the global project are no longer displayed here. If, in the `Settings` tab, the option `Keep the existing data in the Analyzer database` was selected, amended objects in the database are deleted or updated.
`Select all`	Selects all entries in the list and activates the checkboxes.
`Deselect all`	Selects all entries in the in the list and deactivates the check boxes.

Event groups

Configuration of the alarm/event groups from the global project to be configured.

Parameters	Description
`Alarm/event groups`	Information and notes on exporting.
`List of the alarm/event groups`	List field with the possibility to select alarm/event groups. To select an entry, activate the check box in front of the entry. `Sorting`: Clicking on the column identifier sorts the entries after this column upwards or downwards. `Multiple selection`: If several rows are highlighted, a click in the check box sets the options for all selected rows. Alarm/event classes that were deleted in the global project are no longer displayed here. If, in the `Settings` tab, the option `Keep the existing data in the Analyzer database` was selected, amended objects in the database are deleted or updated.
`Select all`	Selects all entries in the list and activates the checkboxes.
`Deselect all`	Selects all entries in the in the list and deactivates the check boxes.

Users

Configuration of the users to be exported from the global project.

Parameters	Description
`Users`	Information and notes on exporting.
`List of users`	List field with possibility to select users. To select an entry, activate the checkbox in front of the entry. `Sorting`: Clicking on the column identifier sorts the entries after this column upwards or downwards. `Multiple selection`: If several rows are highlighted, a click in the check box sets the options for all selected rows. If, in the `Settings` tab, the option `Keep the existing data in the Analyzer database` was selected, amended objects in the database are deleted or updated. If a user was renamed in zenon they are considered new and recreated in the project. The previous user is deleted.
`Select all`	Selects all entries in the list and activates the checkboxes.
`Deselect all`	Selects all entries in the in the list and deactivates the check boxes.

Projects

Configuration of the local projects to be exported. The names for the server and standby-server can be changed here. To do this:

1. Highlight the project in the list of projects

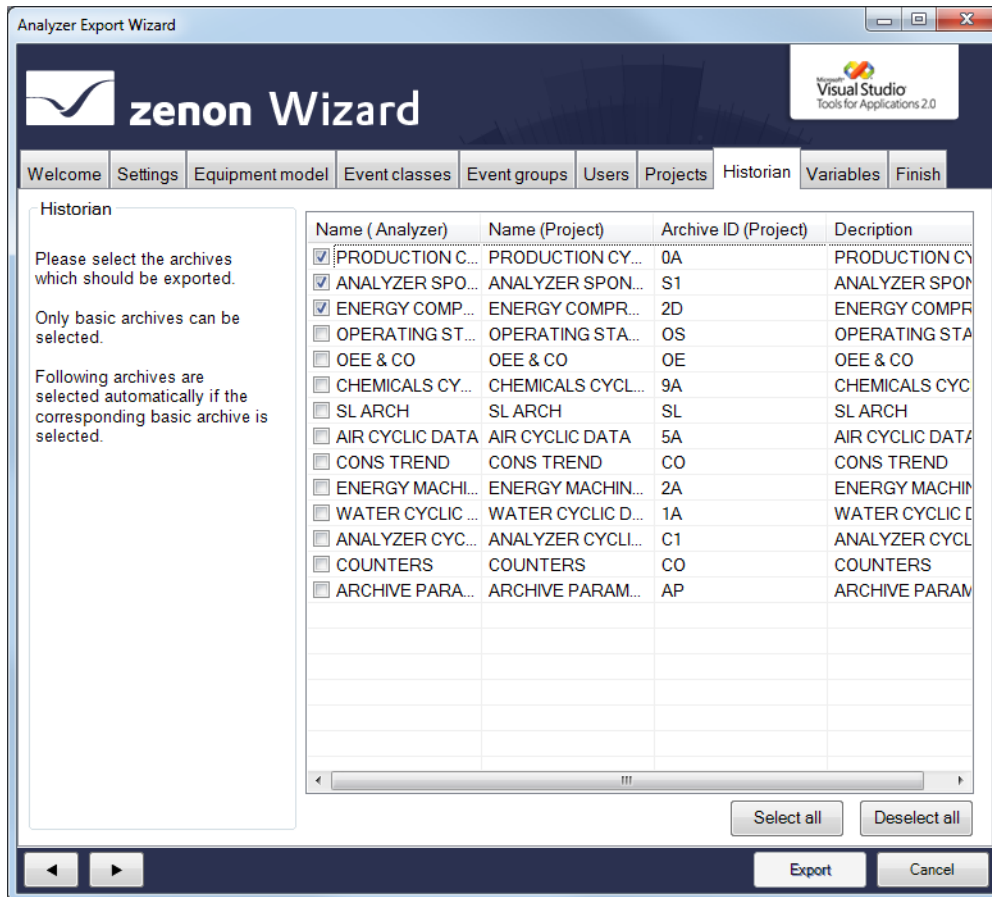2. Enter the desired name for the server and standby-server

If the name of the server or standby server is changed in the zenon project, then this is only updated in th analyzer database if the `Network active` setting was activated in the project properties.

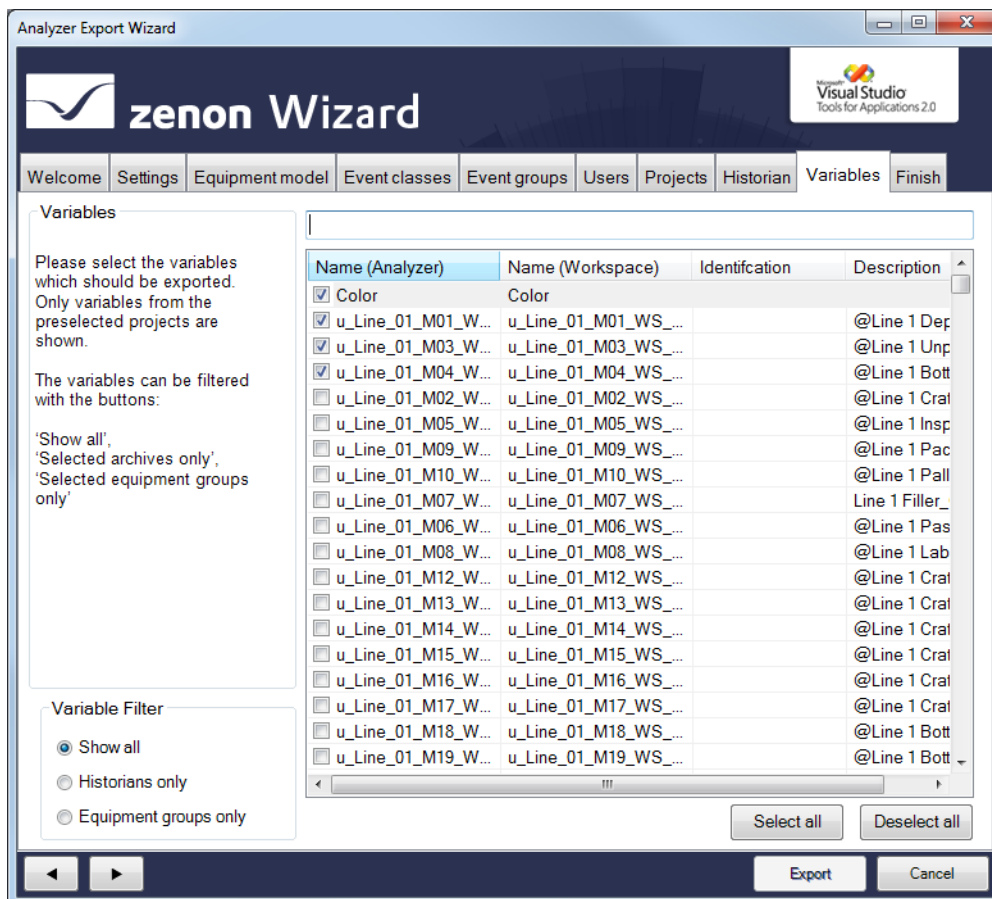| Parameters	Description
`Projects`	Information and notes on exporting.
`Project list`	List field with possibility to select for projects. To select an entry, activate the checkbox in front of the entry. **Sorting**: Clicking on the column identifier sorts the entries after this column upwards or downwards. **Multiple selection**: If several rows are highlighted, a click in the check box sets the options for all selected rows. If, in the **Settings** tab, the option `Keep the existing data in the Analyzer database` was selected, amended objects in the database are deleted or updated.
`Server`	Address of the server for the project selected in the list window.
`Standby`	Address of the standby server for the project displayed in the list window.
Select all	Selects all entries in the list and activates the checkboxes.
Deselect all	Selects all entries in the in the list and deactivates the check boxes.

Historian

Selection of the archive from the selected projects (on page 24). Only the main archive is displayed. Subsequent archives are not displayed in the list, but are also selected with the main archives and written to the database.

Parameters	Description
`Historian`	Information and notes on export.
`Archive list`	List field with possibility to select for archives. To select an entry, activate the checkbox in front of the entry. `Sorting`: Clicking on the column identifier sorts the entries after this column upwards or downwards. `Multiple selection:` If several rows are highlighted, a click in the check box sets the options for all selected rows. If, in the `Settings` tab, the option Keep the existing data in the Analyzer database was selected, amended objects in the database are deleted or updated.
`Select all`	Selects all entries in the list and activates the checkboxes.
`Deselect all`	Selects all entries in the in the list and deactivates the check boxes.

Variables

Configuration of the variables to be exported from local projects. When selecting variables, the entries offered can be prefiltered.

| Parameters	Description
`Variables`	Information and notes on exporting.
Variable Filter	Selection of the variable filters: ▸ Show all: all variables are displayed ▸ Historians only: only historian variables are displayed ▸ Equipment groups only: only variables that are in the selected equipment models (on page 16) are displayed.
Filter row	Input of alphanumerical characters according to which the **List of variables** is to be filtered.
`List of variables`	List field with possibility to select variables. To select an entry, activate the checkbox in front of the entry. **Sorting**: Clicking on the column identifier sorts the entries after this column upwards or downwards. **Multiple selection**: If several rows are highlighted, a click in the check box sets the options for all selected rows. If, in the **Settings** tab, the option Keep the existing data in the Analyzer database was selected, amended objects in the database are deleted or updated.
`Select all`	Selects all entries in the list and activates the checkboxes.
`Deselect all`	Selects all entries in the in the list and deactivates the check boxes.

Finish

To export the configured data:

1. In the Finish tab, click on the `Export` button
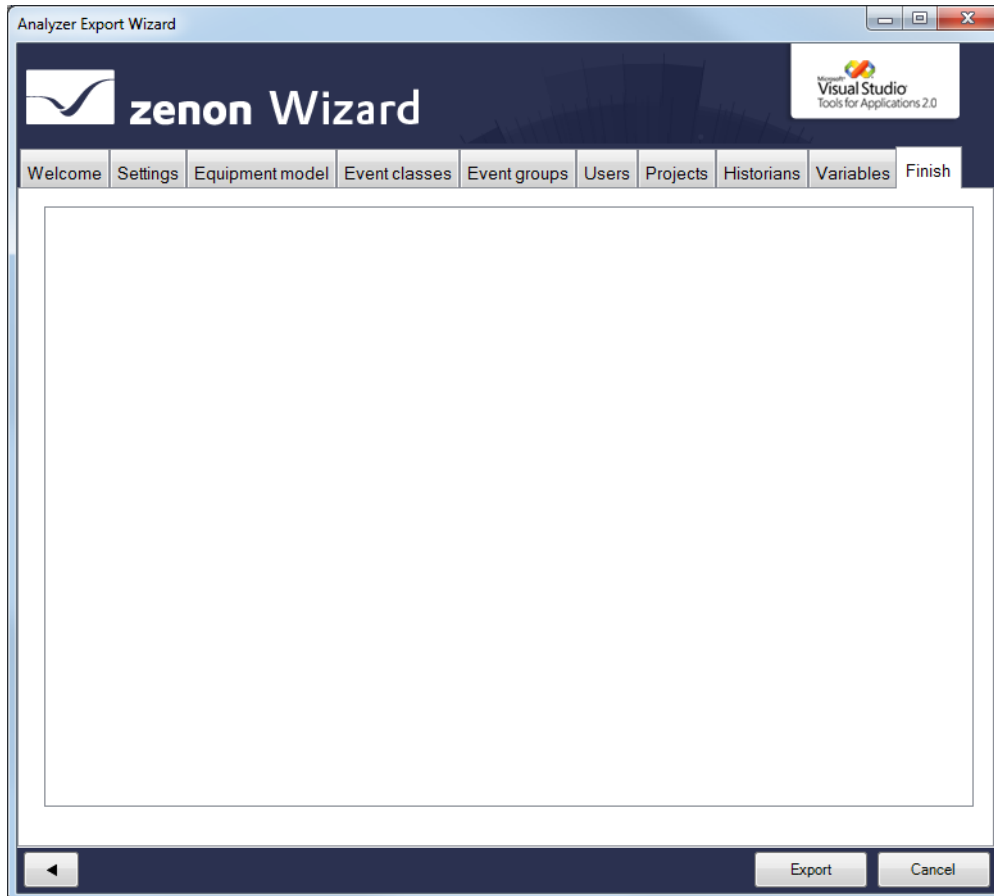
2. the export is started

3. The exported elements are shown in the output window with the attendant success and error messages
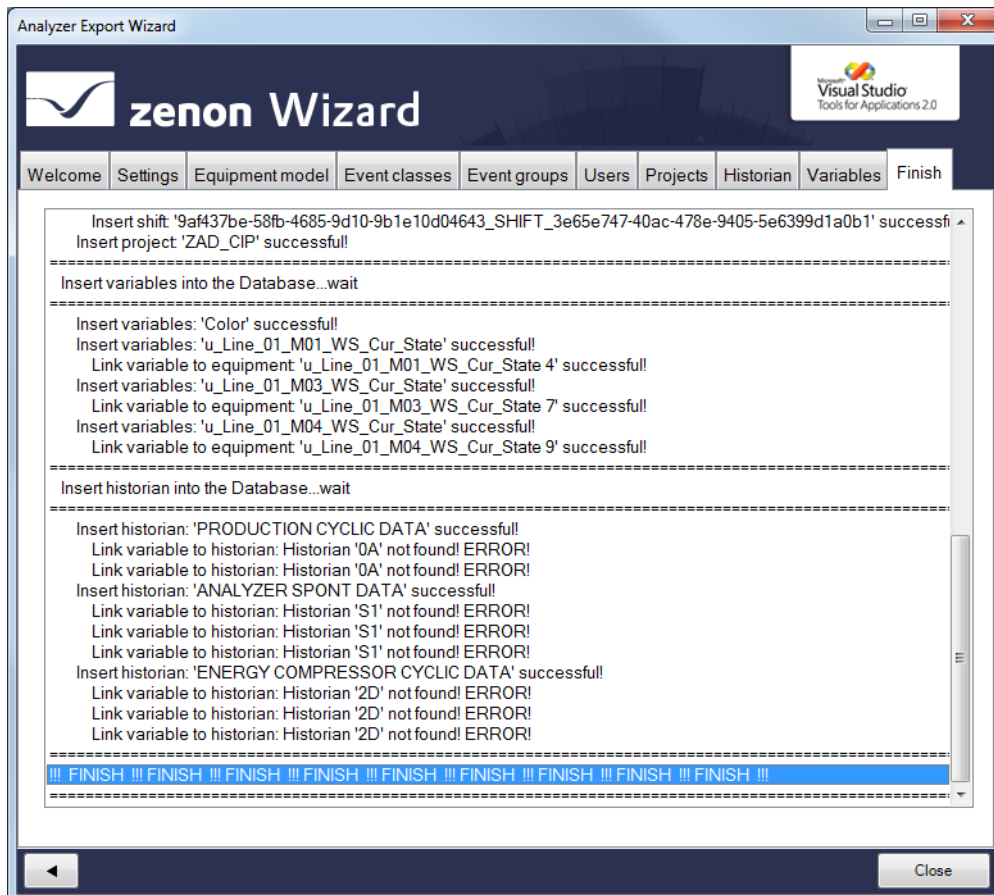
4. Click on **Close** to close the wizard

RECONFIGURING THE WIZARD

To reconfigure the wizard:

1. Open the Settings (on page 12) tab

2. click on button Load data

3. Configure the tabs

Exit wizard

To close the wizard:

▸   Click on the `Cancel` button

▸   A dialog asks if the configuration should be saved

- Clicking on `Yes` writes the settings configured in the Settings (on page 12) tab to the registry and closes the wizard; the wizard is opened with this configuration next time it is started

- Clicking on `No` closes the wizard without saving the configuration

2.1.2   Import - Export

Wizards for export and import of data.

FactoryLink import wizard

The FactoryLink import wizard is an assistant that supports the user when porting a FactoryLink project to zenon.

> ⚠ **Attention**
>
> *The zenon Editor language should be set to English, in order to ensure that the FactoryLink project is imported with as few errors as possible.*

Export of project data from FactoryLink

FactoryLink project data is exported via the Menu *Display - Library Converter* in the ClientBuilder application. Here, the desired libraries all all project data to be converted must be selected.

> ⚠ **Attention**
>
> *ASCII must be selected as the target format.*

In addition, there must be access to all bitmaps used in the project. All files must remain in the file structure that was created by FactoryLink.

Import of the project

A selection dialog is displayed using the *File - Wizards...* menu, which displays all wizards available in zenon. The FactoryLink Import wizard is in the *Wizards - Export/import - FactoryLink Import Wizard* group.

Welcome

On the first page of the wizards, the process and the following pages of the wizard are briefly explained.

Preparation

Basic information for executing the wizard is available on this page. To exclude the possibility of two names of screens and templates when importing, it is recommended that you create a new project in zenon. The zenon dialog to create a new project can be called up by clicking on 'Create empty project' if the project that is currently loaded is not to be empty.

> ⚠ **Attention**
>
> *After creating a new project, the zenon wizard selection dialog opens automatically. This selection dialog must be closed, because the FactoryLink Import Wizard remains active in the background.*

Tag selection

On this page, the file imltags.asc of the FactoryLink project to be imported must be selected. All of the project's variable information is imported into zenon by clicking 'Import!'. The wizard opens the file with the variable information and reads names, types and descriptions of the variables. The internal driver is used as a standard driver in zenon.

> 💡 **Info**
>
> *After the variables have been imported, the drivers used in zenon can be changed via Properties -> Addressing -> Driver Connection -> Driver.*

Mimic selection

On this page, all relevant folders in which picture data, templates, bitmaps and project symbols are located must be selected.

In addition to normal import as a picture symbol, there is also the possibility to create a symbol in the wizard. In doing so, the wizard creates a picture with all symbols contained in the project. These symbols must now be manually copied into the local symbol library of the zenon project. If the project contains

FactoryLink animations (for example Symbols Bit Group), in which variables with symbols are linked, the wizard can transfer these animations to a zenon combined element.

Events

A summary of the conversions is shown on this page. The list can be filtered for certain event if required.

List of importable objects

- Import of variable names, description and types, transfer of FactoryLink connections to the zenon internal driver

- The import and creation of FactoryLink templates and mimics.

- Import of FactoryLink Bitmaps in the picture folder of the zenon project manager

- Import and creation of static picture elements:

- Rectangle

- Lines

- Rounded rectangle

- Circle/ellipse

- Text

- Polygons

- Polylines

- Pie charts

- Bitmaps

- Import of MultiLang texts from text elements in zenon language tables

- Fonts from FactoryLink projects are approximated in size and adapted to existing standard project fonts

- Possibility of creating a symbol, which contains all symbols for manual import into the local symbol library

- FactoryLink symbol import as zenon symbols

- Alternative symbol import from zenon symbol library, if manual import was carried out

- ▶ Conversion of CB animations (ColorBit) to a zenon combined element

- ▶ Conversion of CB animations (ColorBit) to a zenon combined element

- ▶ Conversion of DR animations (display register) to a zenon text element

- ▶ Conversion of SR animations (send register) to a zenon text button

PDiag import wizard

The zenon PDiag import wizard supports PDiag messages during import. This wizard can also serve as a template for your own expansions in this respect.

The PDiag import wizard is only available in English.

Requirements

The wizard imports an XML file exported from the `process diagnosis (PDiag)` Simatic module. This export can be carried out with the menu item `Process diagnosis | Export` in Simatic Manager.

The XML file created in this way is imported with the wizard in zenon. In doing so, variables that are based on the S7-TCP driver, of driver object type `Alarm S` are created. Each message and each accompanying value corresponds to a zenon variable.

Only UINT variables are imported as accompanying values. In doing so, the message text is analyzed for the Simatic PDiag identifier "@1X%2d@", "@1X%3d@" und "@1X%4d@" and adapted for zenon accordingly. It is also possible to use the language table for dynamic texts. In doing so, the texts outside the wizard are to be transferred to the zenon.

This wizard also imports S7 graph messages, which are automatically generated and thus also contained in the XML file.

Settings

To start the wizard:

1. Select `Wizards..` in the `File` menu.

   Alternatively: The key combination `Alt+F12`

2. The dialog to select the zenon wizards is opened

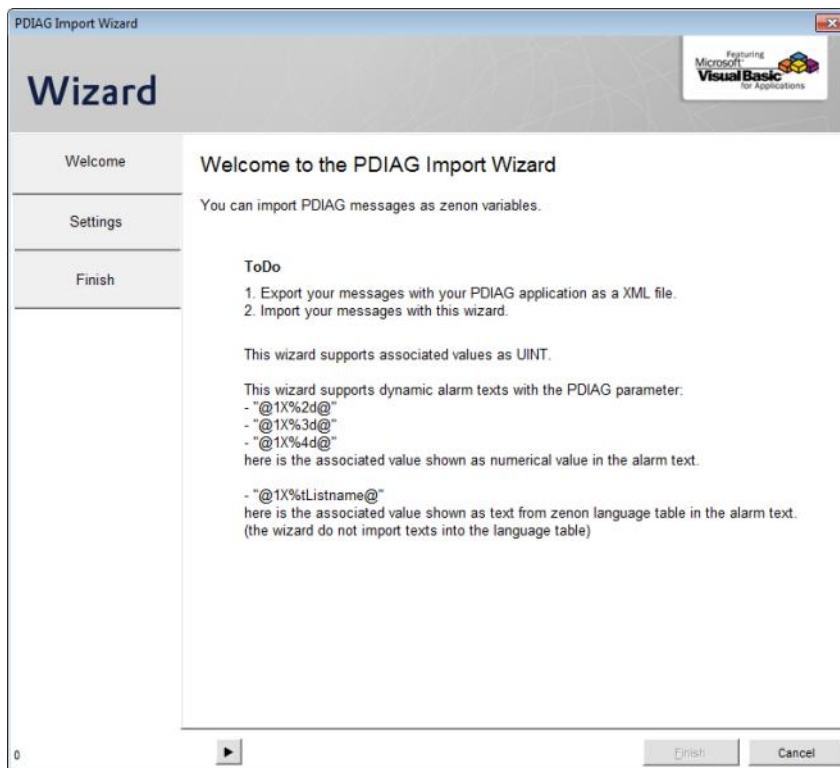3. Open the **Export/Import** nodes

4. Select **PDIAG wizard**

The wizard leads to, via the

▶ Welcome (on page 37)

▶ Settings (on page 38)

▶ Finish (on page 40)

tabs, to the import of an XML file.
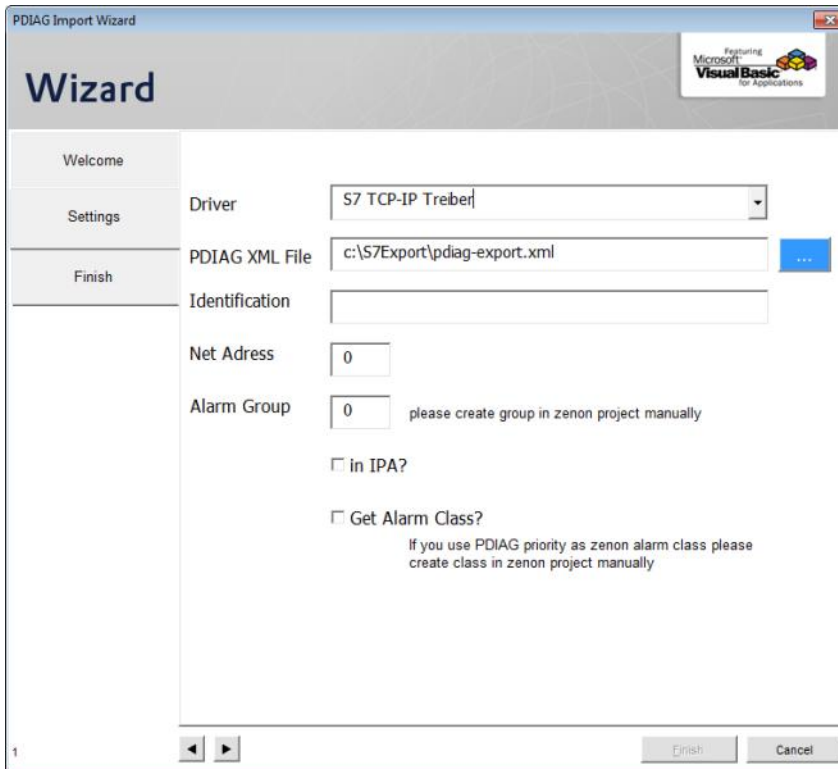
Welcome

The **Welcome** tab contains a short explanation of how the wizard works and what it requires.

To get to the Settings (on page 38) tab, click on the **Settings** menu or on the arrow on the bottom left.

Settings

You can adjust the import settings in the `Settings` tab.

Setting	Description
Driver	Selection of the S7-TCP driver from the active project for which the import is to be carried out.
PDIAG XML file	Selection of the XML file to be imported.
Identification	Optional entry for variable identification. This entry can be filtered in the variable list.
Net address	Defines the net address for the zenon variable addressing. You can see the valid net address in the driver configuration.
Alarm group	Sets the alarm/event group of the messages to be imported. **Attention**: The wizard does not create alarm/event classes in zenon independently. These must be manually created before the import.
In IPA	`Active`: Sets the `Save in IPA database` property for the variables. This transfers the messages to the industrial performance analyzer.
Get Alarm Class	`Active`: The message priority set in Simatic Manager is interpreted as zenon alarm/event class. **Attention**: The wizard does not create alarm/event groups in zenon independently. These must be manually created before the import.

The import is started with **Finish**. This button is only active in the Finish (on page 40) tab. Click on **Finish** in the menu or on the arrow at the bottom left.
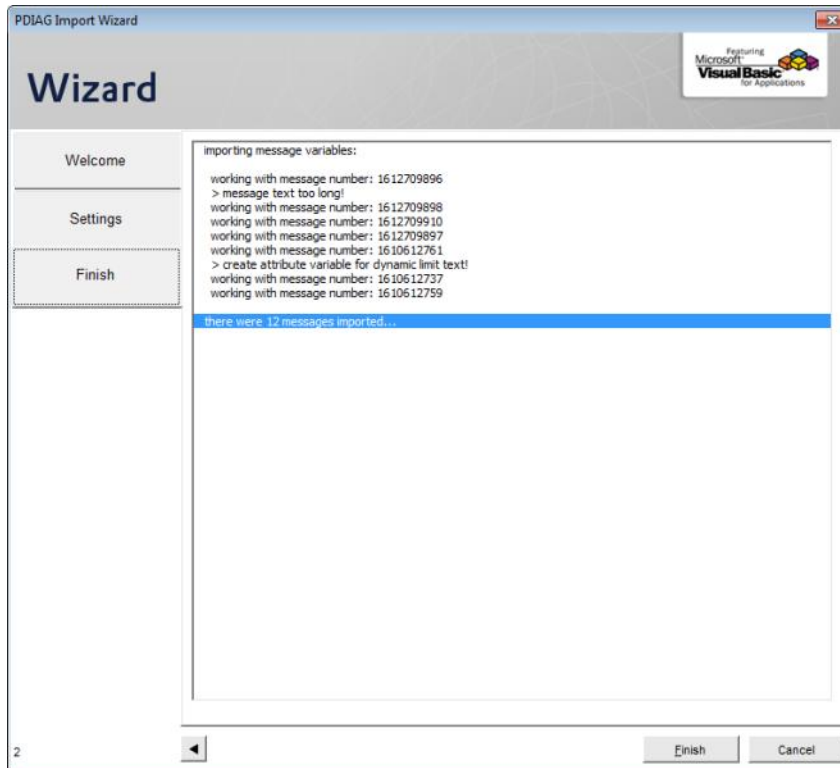
Finish

To start the import, click on the `Import` button.
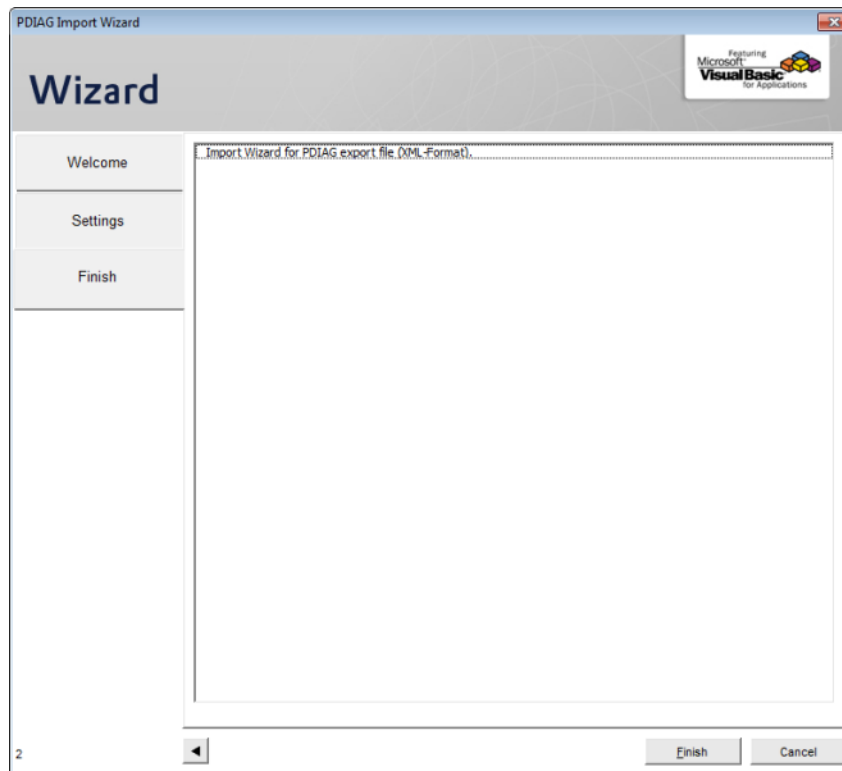
Import of messages

Notices are given during the import:

After the import has finished, there is a note stating how many variables were imported.

IMPORTANT MESSAGES

Message	Description
`message text too long`	The alarm text is too long and will be cut to the valid length in zenon.
`creating attribute variable for dynamic limit text`	In addition to the message variable, an accompanying value variable for the dynamic limit value text is created.

WinCC Import Wizard

The `WinCC Import Wizard` imports selected parts of an existing WinCC project to the currently loaded zenon project. The import of the WinCC project data is carried out using two programs:

- ▶ WinCC Graphics Converter (on page 45): exports WinCC screens, frames and symbols from WinCC in an XML file

- ▶ WinCC Import Wizard (on page 48): imported
  - • Data blocks (structure TAGs), TAGs, alarms and archives directly
  - • Screens, frames and symbols via XML files which were previously created with the help of the WinCC Graphics Converter (on page 45)
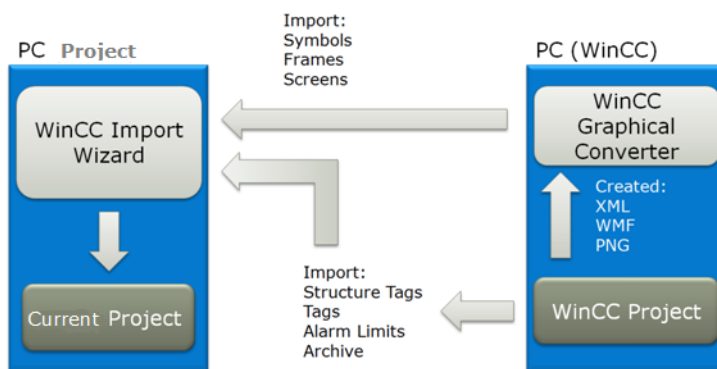
Installation

To execute the WinCC Import Wizard, you must first install all components.

> 💡 **Info**
>
> *Note that zenon should NOT be installed on the same computer as WinCC.*

INSTALLATION WINCC GRAPHICS CONVERTER

In order to access the WinCC information, the `WinCC Graphics Converter` must be installed on the computer on which the WinCC project runs. The program is located on the zenon installation medium in folder `\Additional Software\WinCC Graphics Converter`.

After the installation you can find the converting tool for different WinCC versions under *Start - All Programs - COPA-DATA - WinCC Graphics Converter*. Always start the tool for matching version.

Info

For using the converter, `.NET Framework 3.5` must be installed. When installing the converter, it is checked whether it is available. If the framework is missing, the installation is canceled. In this case first install `.NET Framework 3.5` and then start the installation of the converter again.

## INSTALLATION OF THE WIZARD FOR ZENON 7.0 AND HIGHER

The wizard is automatically installed together with the zenon Editor. No separate settings are needed. You can start the wizard right away in the zenon Editor under *File - Wizards...* and there under Export/Import.

INSTALLATION OF THE WIZARD FOR ZENON 6.51

As the wizard is not a part of 6.51 SP0, you must carry out the followuing steps for the installation:

▶ Install at least Build 6 of zenon 6.51. You can request Build 6 from your distributor or from the COPA-DATA Support (mailto:support@copadata.com).

▶ Install the wizard together with the WinCC Graphics Converter. This setup installs the WinCC Graphics Converter tool and also the wizard if zenon 6.51 SP0 is installed. You can request the setup from your distributor or from the COPA-DATA Support (mailto:support@copadata.com).

▶   After the installation, start the zenon Editor. The dialog for updating the wizard is displayed. Via Start update the wizard is added to the VSTA workspace.

If you want to carry out this step later, you can return to this dialog in the zenon Editor via menu *File - Update wizards…*.
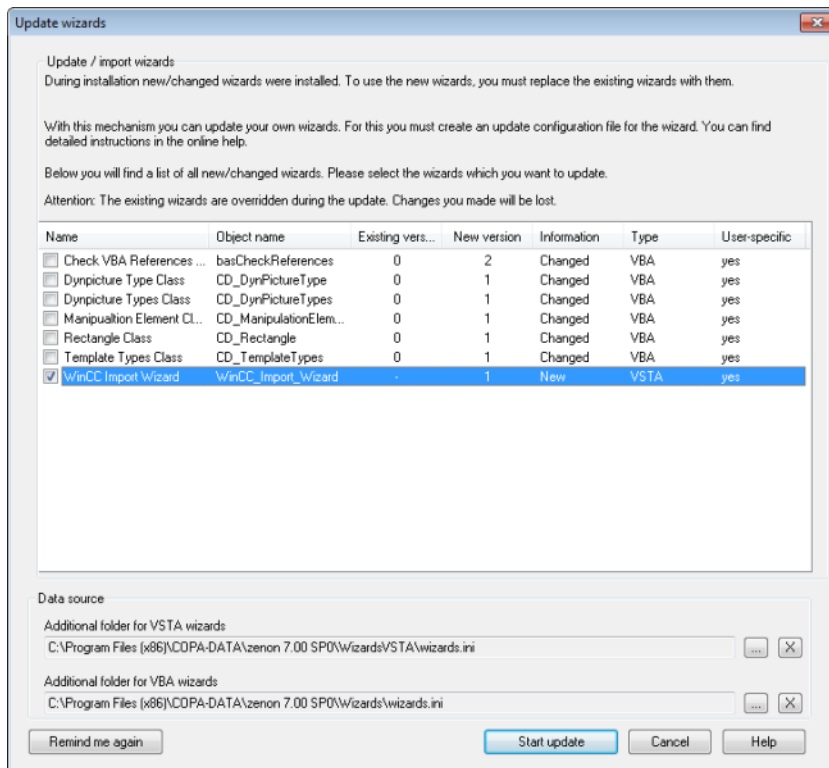
▶   Start the VSTA Editor in the zenon Editor via menu *File - Open VSTA Editor…* and select node `References` in window `Project Explorer`. Carry out menu item *Add Reference..* in the context menu in order to add two missing references:

•   MSDASC: In the Add Reference dialog click on tab COM and add the component `Microsoft OLE DB Service Component 1.0 Type Library` to the project.

- ADODB: In the Add Reference dialog click on tab `Browse`. Navigate to the folder
`C:\Program Files (x86)\Microsoft.NET\Primary Interop`
`Assemblies` and select file `adodb.dll`.
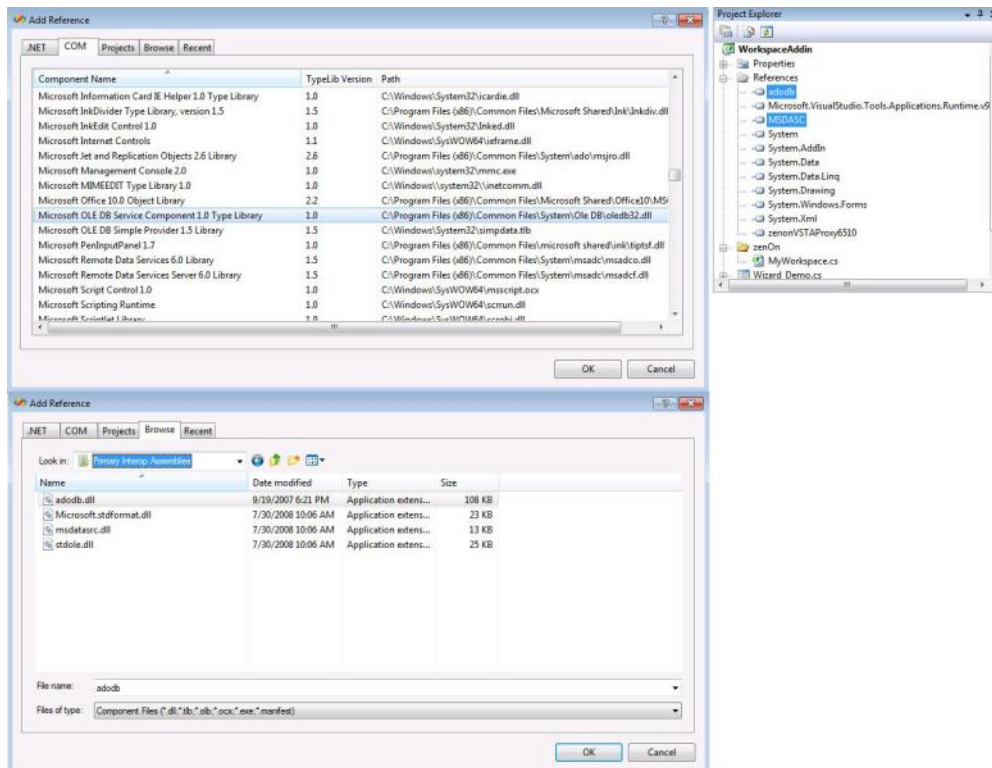
▶ In window `Project Explorer` you can now select node `WorkspaceAddin` and compile the add-in via menut item `Build` in the context menu.

After the compiling was successful, the wizard is available in the zenon Editor under *File - Wizards...* and there under Export/Import.

WinCC Graphics Converter

The `WinCC Graphics Converter` makes it possible to select screens, frames and symbols in WinCC projects and export them as XML files.

💡 **Info**

At the moment XML files can be created from WinCC projects of versions `7.0` and `7.0 SP1`.

Welcome

With the help of the WinCC Graphics Converter you can convert WinCC graphics files (PDL) to an XML format which the WinCC Import Wizard can read. Existing graphics information are saved as PNG files and WMF files together with the XML files and stored in a selected folder.

To execute the converter:

1. click on Start

2. navigate to `COPA-DATA -> WinCC Graphics Converter`

3. start the **WinCC Graphics Converter**

4. follow the instructions of the wizard
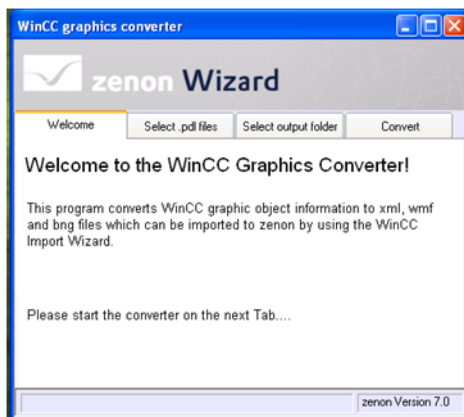
Select .pdl files

On this tab you select the PDL files which should ne exported from the WinCC project. To do this:

1. click on button **Select .pdl files**

2. navigate to the project folder which contains the PDL files

   Note: In order that the files can be selected, the WinCC project must be loaded on the computer!

3. select the desired files

4. all selected PDL files are displayed in the preview window
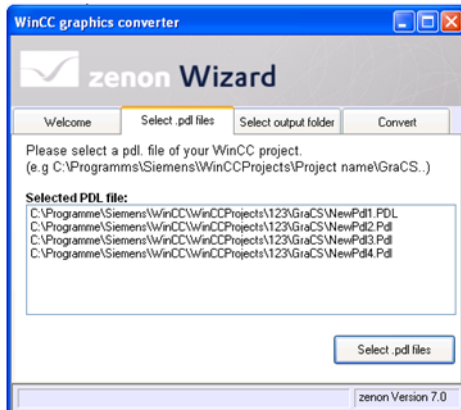
Select output folder

On this tab you select the folder in which the export files should be saved. To do this:

1. click on button **Select output folder**

2. avigate to the folder in which the export files should be saved

3. **Note:** You can create a new folder in the selection dialog
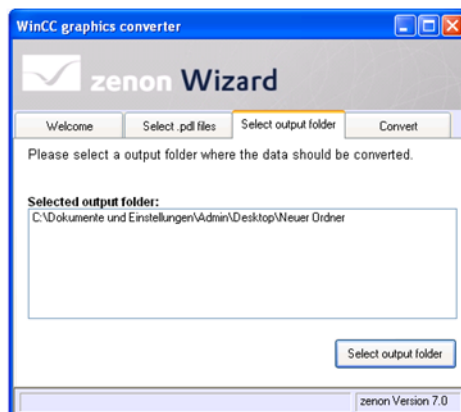
Convert

On this tab you can carry out the conversion.

After the successful export copy the folder to the computer with the zenon project in which the data should be imported or make sure that the computer with the zenon project has access to the export folder.

WinCC Import Wizard

The WinCC Import Wizard is started via the wizard dialog of the Editor and can be used to import the following WinCC elements:

▶ Import of the Screens (on page 58) (the XML files created with the WinCC Graphics Converter (on page 45) are converted to frames, screens and symbols in zenon)

▶ Import of the WinCC Tags (on page 53) (only S7 TCP)

▶ Import of the WinCC Structure Tags (on page 52) (only S7 TCP)

▶ Import Alarm Limits (on page 54)

▶ Import Archive Tags (on page 57)

STARTING THE WIZARD

For wizards to be displayed, the settings for VBA and/or VSTA must be set correctly in file `zenon6.ini`:

`[VBA]`

`EIN=1`

`[VSTA]`

`ON=1`

If VSTA wizards are not displayed although the settings are correct, set entry `LOADED=` to `1` in area `[VSTA]`.

To start the wizard:

1. in zenon open menu `File`
    or press the shortcut `Alt+F12`

2. select the entry `Wizards...`

3. the selection dialog is opened

4. navigate to the `Export/Import` node

5. select the `WinCC Import Wizard`

6. by clicking `OK`, you start the wizard

The wizard is divided into areas:

▶ Welcome (on page 49): Overview over the wizard.

- ▶ Settings (on page 50): Settings for the connection to the WinCC project. After the connection has been established successfully, the tabs for the direct import are displayed:

  - Data Blocks (on page 52): Structure Tags from WinCC

  - Tags (on page 53): Tags from WinCC

  - Alarm Limits (on page 54): Alarm classes and alarms from WinCC

  - Archive Tags (on page 57): Archive files from WinCC

- ▶ Screens (on page 58): Import of the screens from WinCC via the WinCC Graphics Converter (on page 45)

Welcome

The start page of the wizard informs you about all other import steps:

The direct import of data is only possible after you have configured the connection to the WinCC project on tab `Settings.`

Settings

On this tab you configure the connection to the WinCC project whose data should be imported.
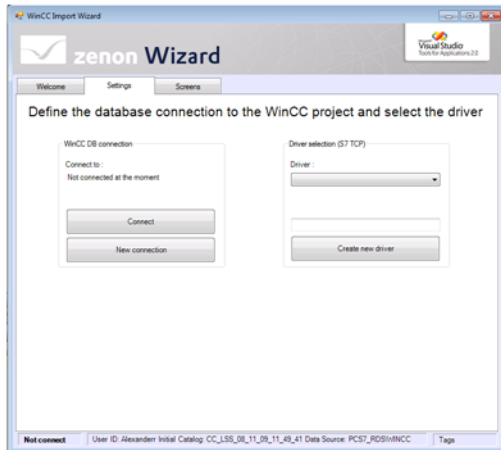
Parameters	Description
`WinCC DB connection`	Configuration of the connection to the WinCC database.
`Connected with`	Display of the active connection.
`Connect`	Establishes a connection.
`New connection`	Opens the dialog for configuring a new connection.
`Driver selection (S7 TCP)`	Configuration of the zenon drivers.
Driver	Selection of a zenon driver from the drop-down list.
`Create new driver`	Opens the dialog for creating a new driver.

> ⚠ **Attention**
>
> *In order that the connection can be established, the WinCC project must be active or opened in the WinCC Explorer on the PC with which the connection should be established.*

CONFIGURATION OF THE CONNECTION

To establish a connection:

1. click on button `New connection`

2. The dialog for the connection settings is opened
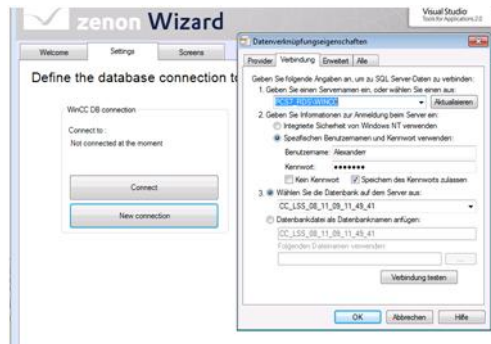
3. on tab **Provider** select the provider Microsoft OLE DB Provider for SQL Server

4. on tab **Connection:**

    a) For `server name` enter the instance of the SQL server in which the WinCC project
        is located. For example: HOSTNAME\WINCC
        Important: The WinCC SQL server instance (sqlsrv.exe) must be enabled in the
        firewall.

    b) For `logon information` enter your access data. At first you must create the
        access data with the help of SQL Server Management Studio in the SQL server
        instance.
        Important: Activate option `Allow saving password`

    c) For `database` select the WinCC Editor project. This is the database name without
        the suffix _R
        Note: The project must be loaded and running in the WinCC Editor. Otherwise the
        project is not available in the SQL server.

    d) Test the connection

5. Close the configuration dialog with **OK**

6. after that you can establish the connection to the WinCC project in the wizard via button
    Connect.

7. select a zenon driver
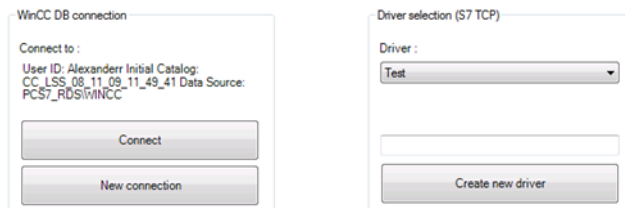
After a successful connection and the selection of a zenon driver, the tabs for the direct import are displayed.
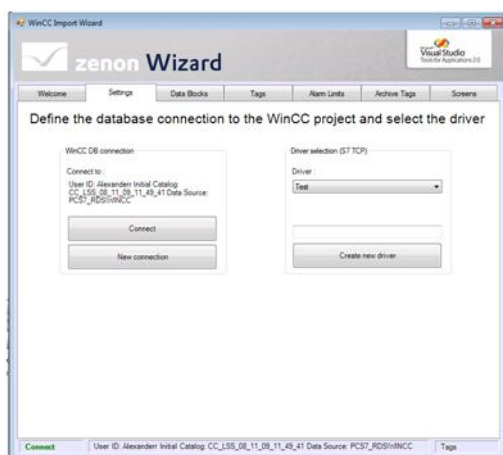
Data Blocks

On this tab you select the data blocks which are displayed in the WinCC Explorer under `Data Blocks` and then imported as data types to zenon.

The WinCC data blocks are grouped according to drivers and are displayed sorted according to block type and name.
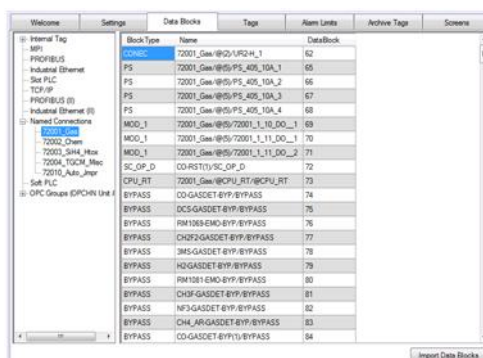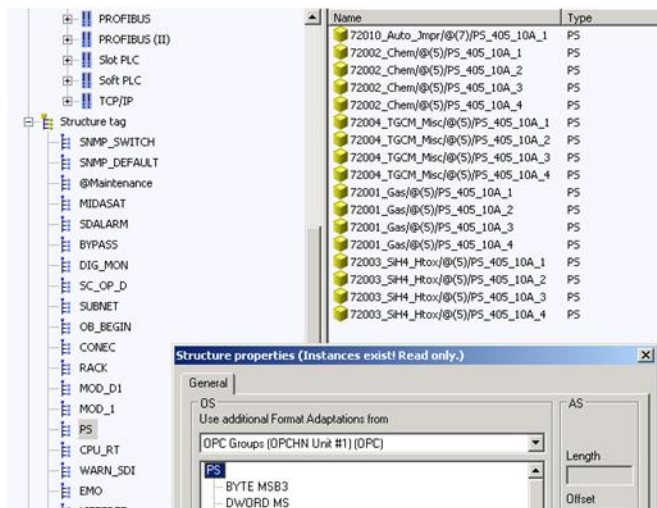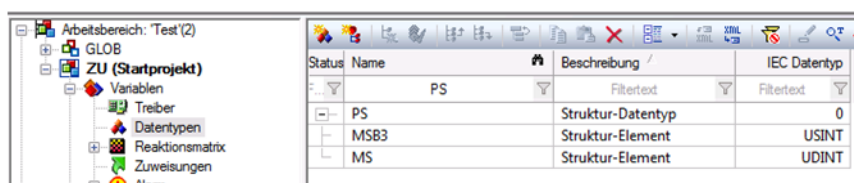

To import data blocks:

1. select the desired data blocks

2. click on `Import Data Blocks`

RESULT

Data blocks in WinCC:

Data types in zenon

TAGs

On this tab TAGs (S7) are selected and imported as zenon variables, which are displayed as `Tag Management` in the WinCC Explorer. The export can be carried out directly to zenon or to a CSV file.
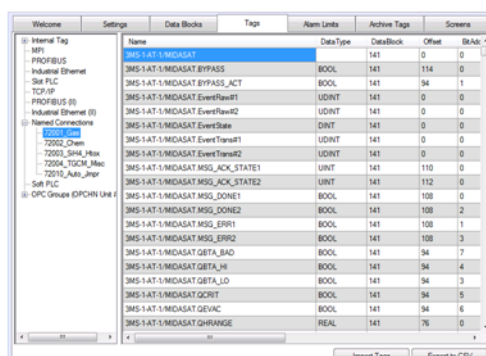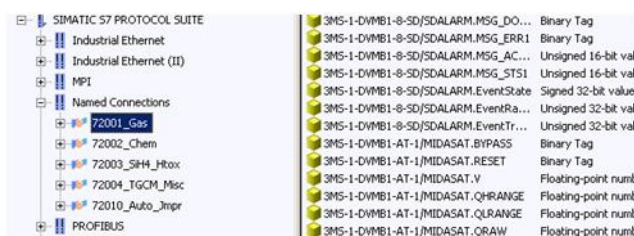
To import tags:

1. select the desired TAGs

2. click on `Import Tags` or `Import to CSV`
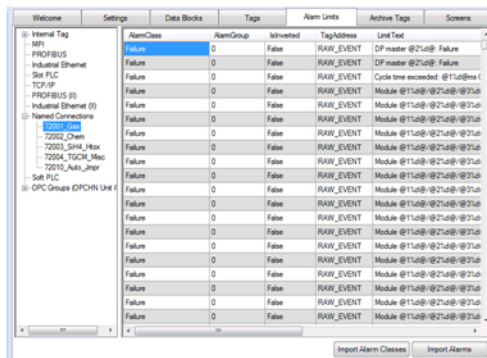
RESULT

Tags in WinCC:

Variables in zenon:

Alarm Limits

On this tab alarm classes and limits are imported:

▶ Import Alarm Classes : Imports alarm classes to an existing global project.

▶ Import Alarms: Imports alarm classes and groups to the local zenon project and creates variables for the limits.

IMPORT ALARM CLASSES

Imports alarm classes from WinCC to a global project in zenon. The global project must already exist and must be active in zenon.
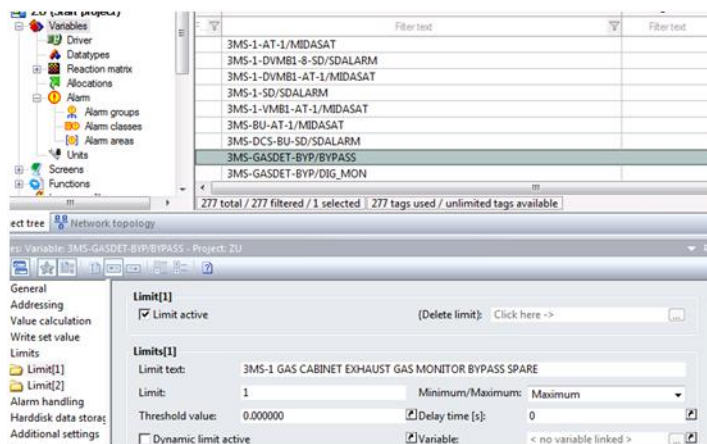

To import alarm classes:

1. select the desired alarm classes

2. click on `Import Alarm Classes`

RESULT

Alarm classes in WinCC:

Alarm classes in zenon:

IMPORT ALARMS

With this kind of import all alarm classes and alarm groups are imported to the zenon project. The WinCC limit texts are replaced by limit variables.

To import alarms:

1. select the desired alarms

2. click on **Import Alarms**

RESULT

Alarms in WinCC:

Alarms in zenon:

Archive TAGs

On this tab TAGs which are entered under `Tag Logging` in the WinCC Explorer can be selected and imported. The import is carried out in one of the two newly created archives BINARY and ANALOG in the zenon project.

> ⚠ **Attention**
>
> *TAGs are only created in the zenon archive if they were imported as TAGs beforehand.*
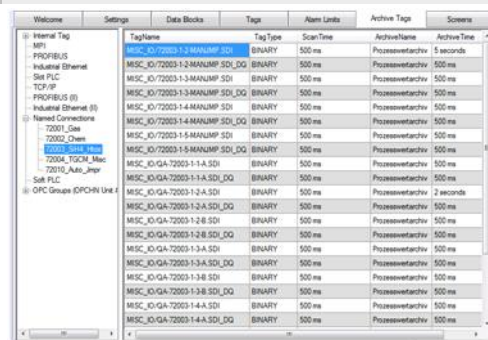
To import Archive Tags:

1. select the desired Archive TAGs

2. click on `Import Archiv Tags`

RESULT

Archive TAGs in WinCC:

Archives in zenon:

Screens

On this tab you can import the XML files which were created with the WinCC Graphics Converter (on page 45) to the active zenon project. In zenon frames, screens and standard screen elements are created based on the information stored in the XML files and based on the WMF and PNG files which are stored in the folder.
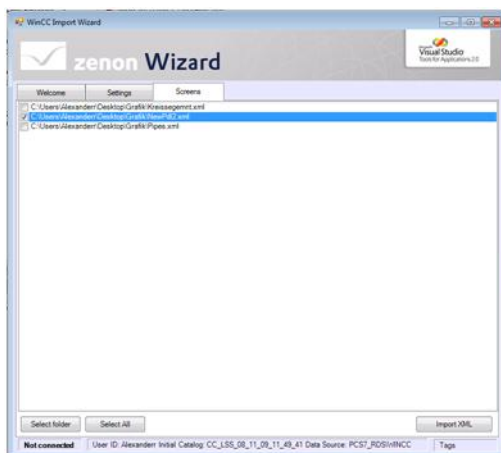
Parameters	Description
`List of files`	Shows all existing files in the selected folder.
`Select Folder`	Opens the dialog for selecting the folder with the import files.
`Select All`	Selects all existing files on the screen.
`Import XML`	Starts the import.

To import screens:

1. select the desired XML files

2. click on `Import XML`

RESULT

Screens in WinCC:
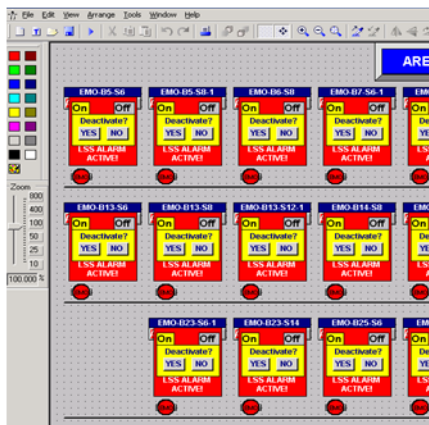
Screens in zenon:
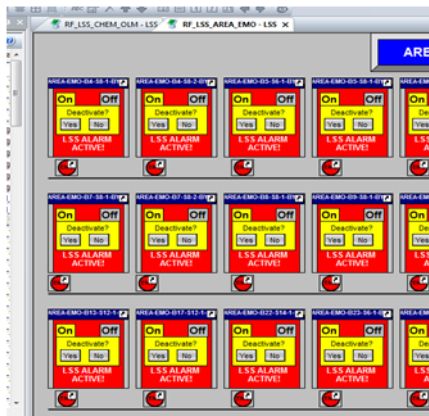
WHICH WINCC SCREEN ELEMENTS ARE AUTOMATICALLY CREATED IN THE ZENON EDITOR?

STANDARD SCREEN ELEMENTS

- ▶ Line

- ▶ Polygon

- ▶ Polyline

- ▶ Ellipse

- ▶ Circle

- ▶ Ellipse segment

- ▶ Pie segment

- ▶ Ellipse arc

- ▶ Circular arc

- ▶ Rectangle

- ▶ Rounded rectangle

- ▶ Static Text

TUBE OBJECTS

▶ Polygon tube

▶ T-piece

▶ Double T-piece

▶ Tube bend

SMART OBJECTS

▶ I/O field

▶ Graphic object

▶ Windows objects

▶ Button

For all other WinCC objects a placeholder is created in zenon.

TROUBLESHOOTING

Errors when importing screens are displayed in the zenon output window when the Runtime files are created.

Error message	Troubleshooting
`The symbol "could not be found in the symbol library"!`	A placeholder for non-interpretable elements has been placed in the screen. Background: A linked symbol is created in the screen using the wizard for each non-interpretable element. This is not in the symbol library however. You therefore have the opportunity to check to see if elements (placeholders) in a screen still need work carried out on them.
`WRN:(FDV_RECETTE_TUNNEL_T POS_DETAILS_FOU2->(ZONE_CNS0)Variable: could not be found in the project!`	Signifies variable names, that are stored in WinCC in `I\O Field Element` but are not (including in WinCC) created as variables.

XML export wizard VSTA

You export all desired modules of a project into a folder of your choice with the XML export wizard. An independent XML file is created for each module.

The wizard is only available in English.

Start wizard

To start the wizard:

▶ Click on *File-> Wizards...*

- • or press the short cut `Alt+F12`

▶ The selection window with the available wizards opens

▶ Select the `Export/import` folder
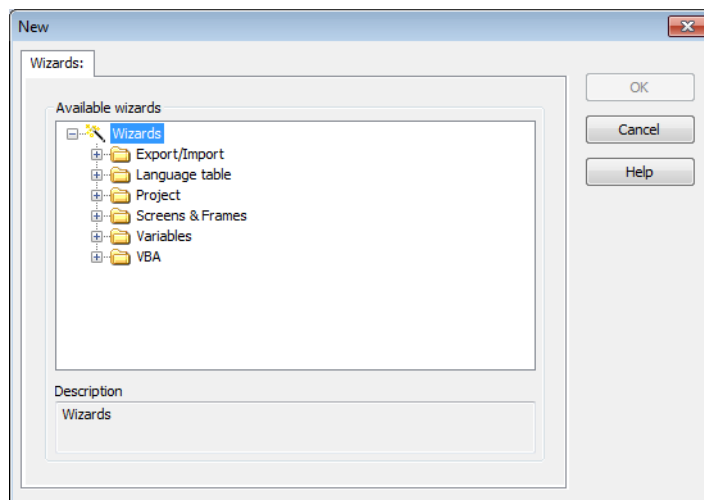
▶ Select **XML export wizard** there
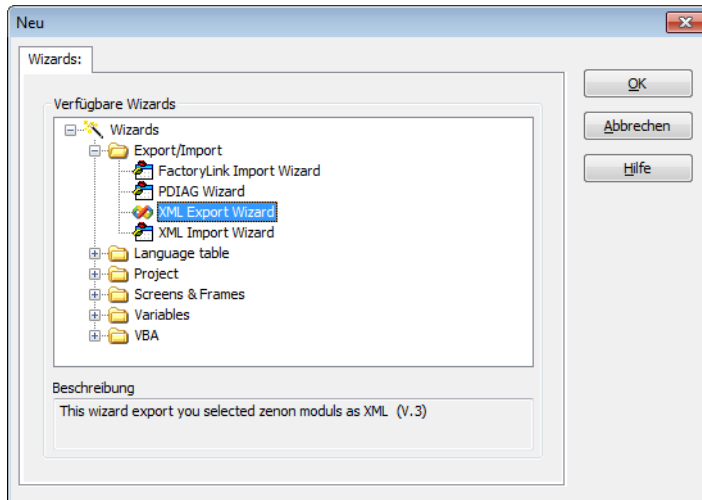
▶ click on **OK**

▶ The wizard starts with the welcome page and displays:

- brief instructions

- the zenon version

- the name of the project from which the export is taking place

- Project GUID

- Project path
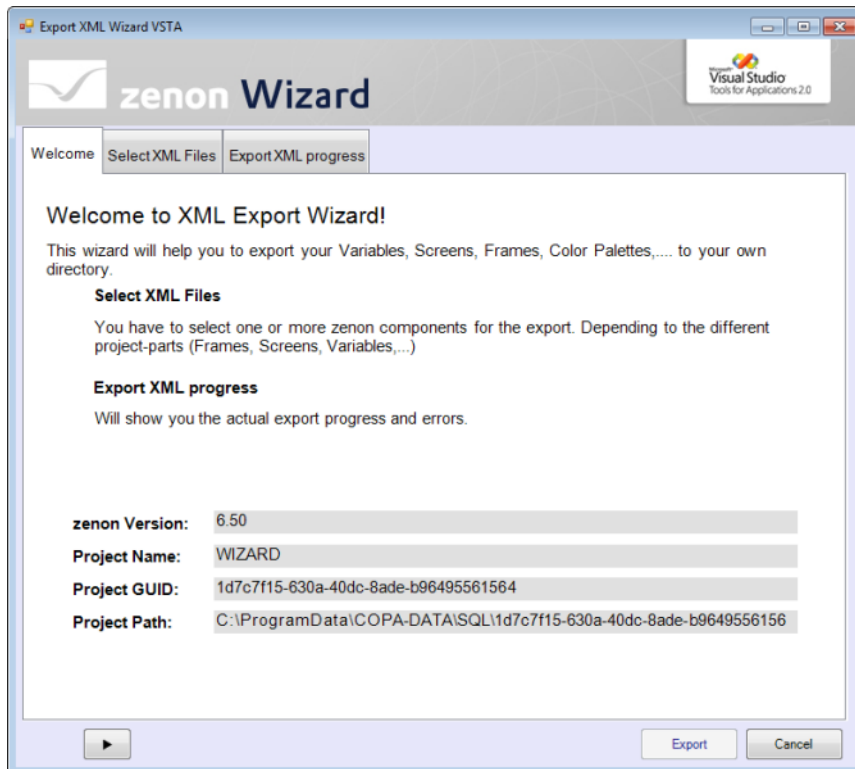
▶ The cursor key leads you step by step through the wizard

- Alternatively, clicking a tab opens the respective setting

▶ To activate the **Export** button, the **Export XML progress** page must be open

Select XML files

Select which module of the project is to be imported:
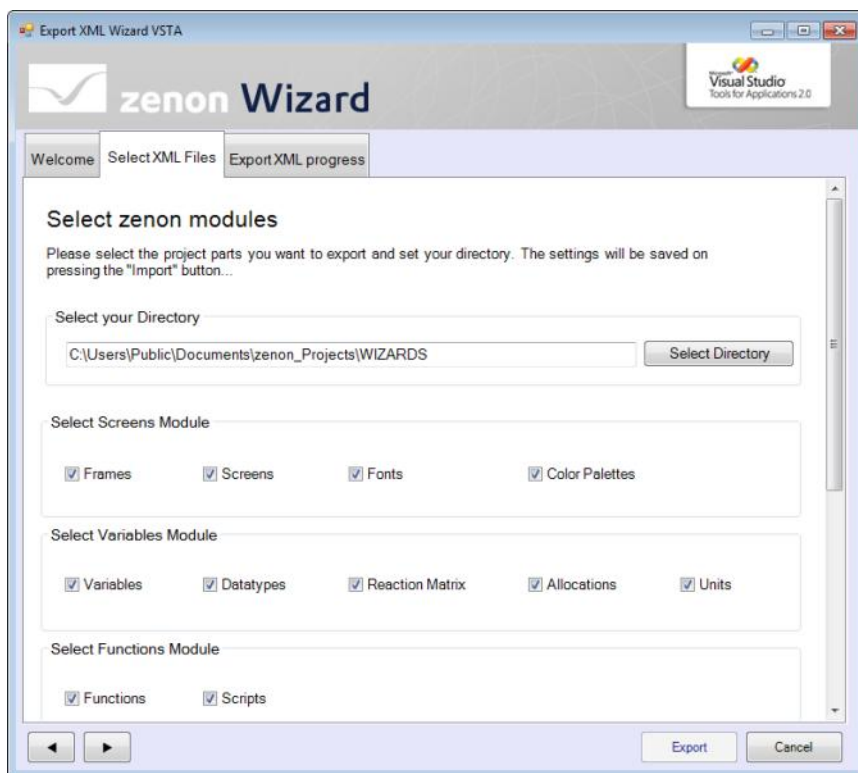
▶ Click on `Select Directory` to define the folder for export

▶ Select the modules and elements that are to be exported by ticking the checkboxes (scroll down if necessary)
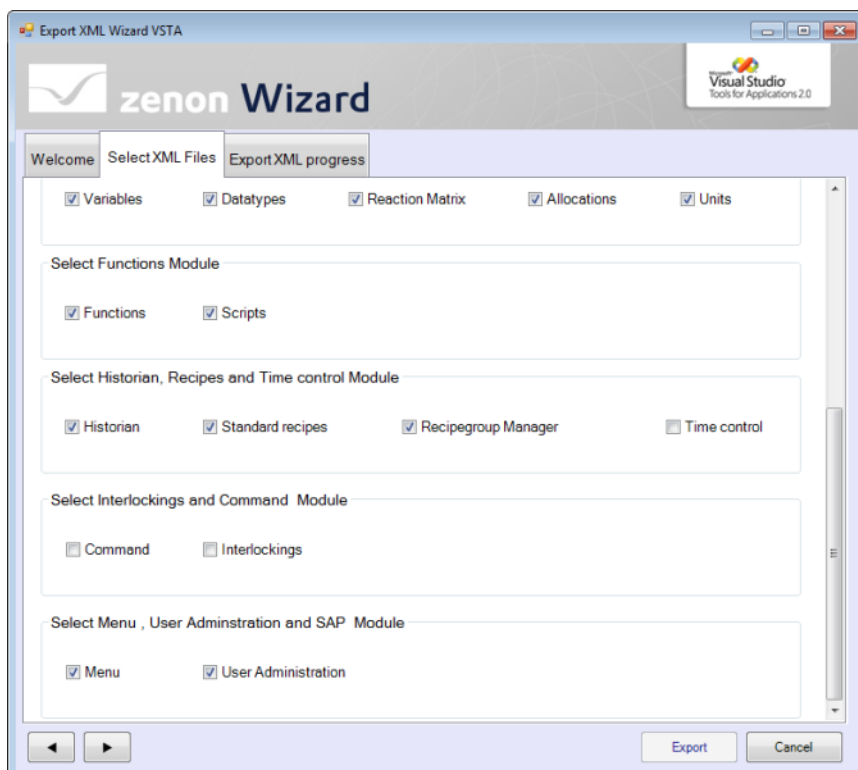
Export

To start the export:

▶ click on the **Export** button

▶ the desired modules are exported

▶ The output window displays which modules are exported with what success

▶ The wizard closes automatically three seconds after the end of the export
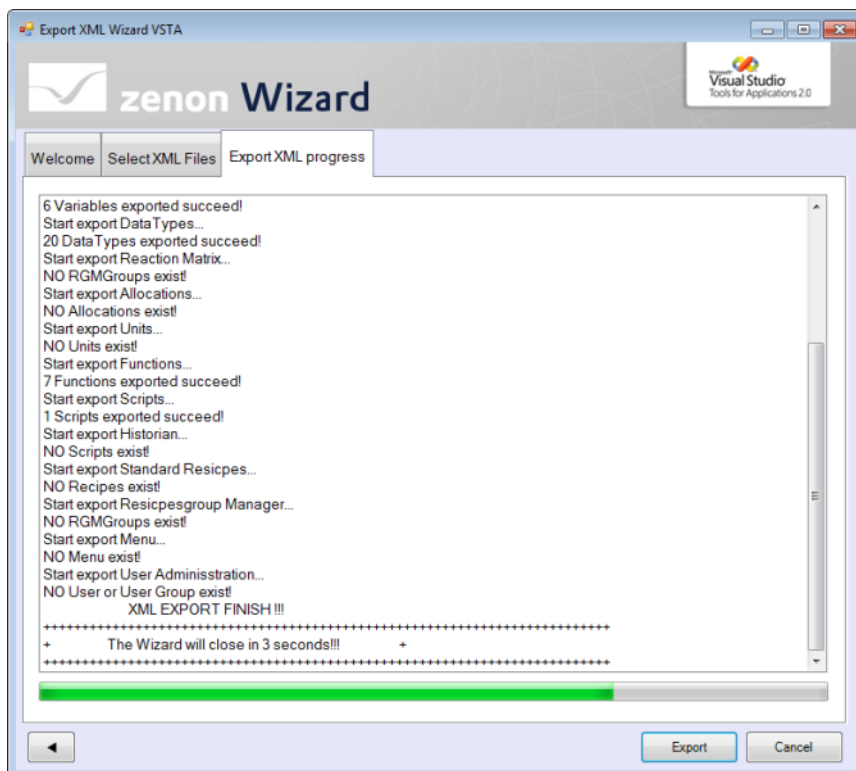
XML Import Wizard

This wizard helps with importing variables, functions, screens and scripts from a XML file.

⚠ **Attention**

This wizard does not support distributed engineering and is not available in multiuser projects.

2.1.3 Language Table

Wizards for language switching.

Language Table Wizard

The Language Table Wizard replaces the old Language Change Wizard (VBA).

The Wizard

▶ Searches the active projects for translatable texts or key words (text marked with a "@") and

▶ writes this

- either to the selected language table in the active project or
- in the global project as an option.

For reference purposes, at least on table (`ZENONSTR.TXT`) must be selected for the import.

If texts without as "@" character are found, these can be set as a key word in a project. To do this, a "@" is written at the start of the text.

Note: The wizard is only available in English.

Calling up the wizard

The wizard can also be selected directly in the dialog for starting wizards.
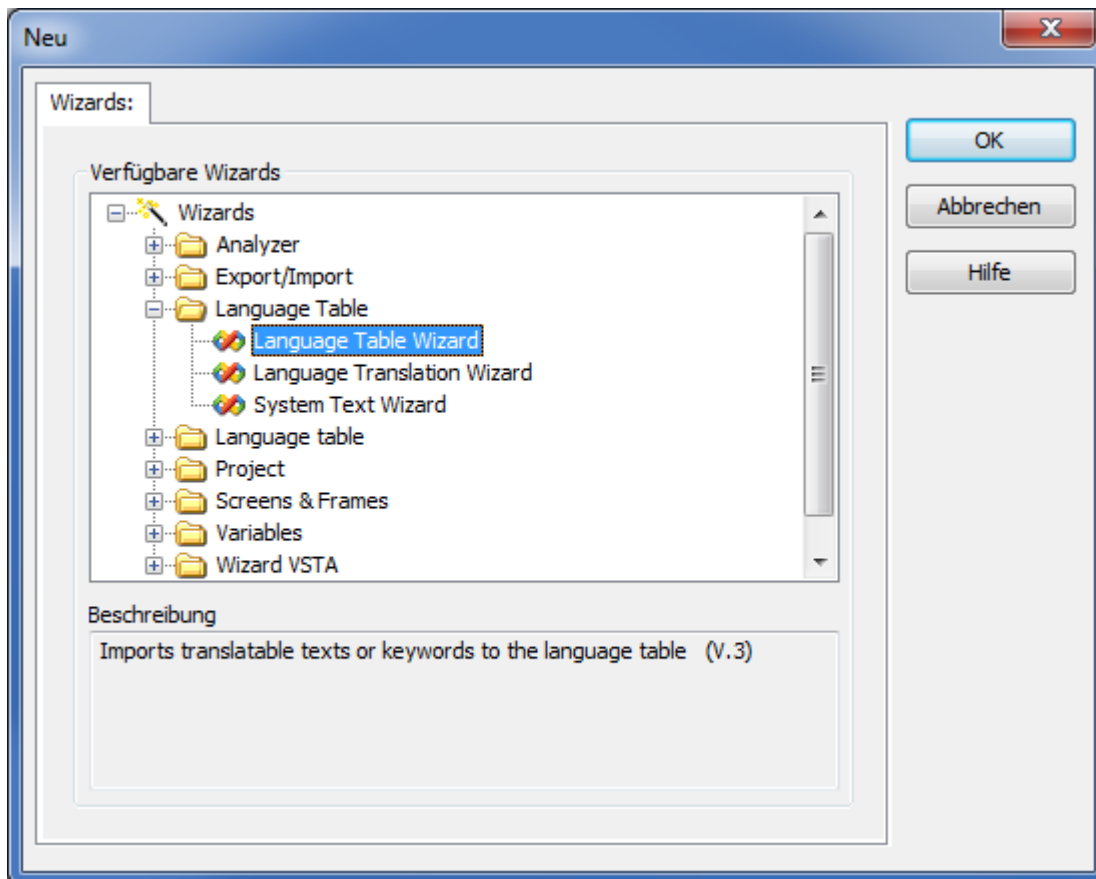
To start the wizard manually:

1. Click on *File-> Wizards...*

   or press the short cut `Alt+F12`

2. The selection window with the available wizards opens

3. Select the `Language Table` folder

4. Select `Language Table Wizard` there

5. click on `OK`

6. The wizard starts with the welcome page

Start window

When opening the wizard, you get an overview page with English-language documentation for the wizard.

The navigation through the wizard is done by clicking on the individual tabs or step by step by clicking on the arrow keys.

Click on **Cancel** to close the wizard.

Settings

IMPORT OPTIONS

It is possible to select the following in the "`Import options`" group:

▶ Whether existing texts are to be overwritten in the selected language table,

▶ Whether translatable texts found are to be set as key words in the project (texts are marked in the project with a @) and

▶ whether the language tables are to be used by the global project.

SELECT LANGUAGE TABLES

In the "`Select language tables`", depending on the setting of the "`Add keywords to language table global project`" option, the language tables of the active project or the global project are displayed. The key word is entered in the selected language tables as a text. At least one table must be selected.

PROJECT OPTIONS

In the "`Project options`" grouping, it is possible to select which project-related texts of key words are to be searched for.

SCREEN ELEMENTS

In the "`Screen elements`" grouping, it is possible to select which screen elements are to be searched.

Finish
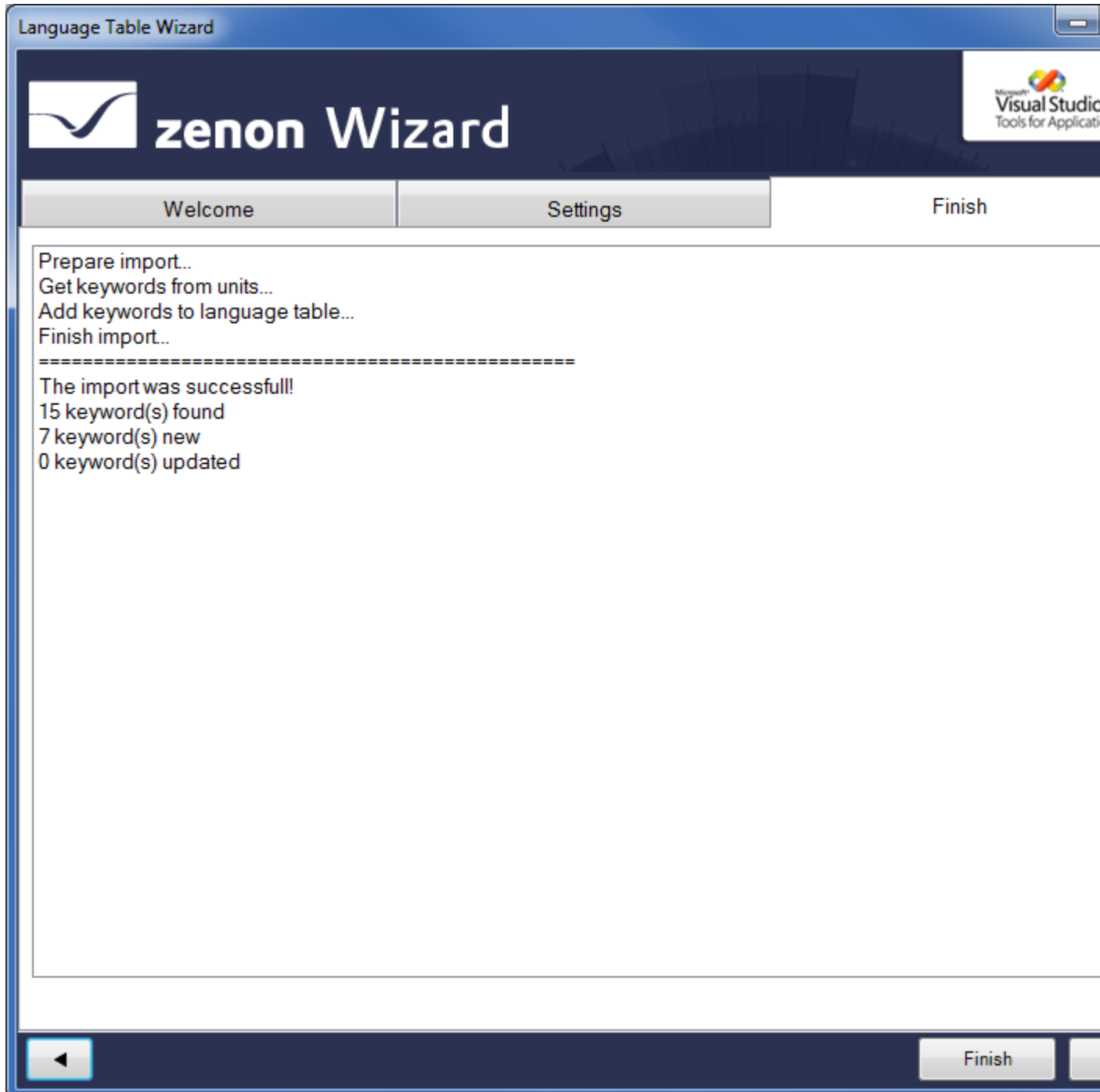
The search for translatable text and key words can now be started on this tab. To do this, click the `Finish` button.

After this, a search is carried out and it is possible to write to the selected language tables.

Language Translation Wizard

This wizard is for the preparation of data for the Project Translation Interface, a translation tool.

In doing so, all relevant data for translation is prepared and compressed into a ZIP file. This ZIP file can then be unzipped in the Procect Translation Interface and the language tables can be edited or supplemented. After editing, the ZIP file can be loaded into the wizard and the edited language tables can be reimported into the project.

Note: The wizard is only available in English.

Start wizard

To start the wizard:

1. Click on *File-> Wizards…*

    or press the short cut `Alt+F12`

2. The selection window with the available wizards opens

3. Select the `Language Table` folder

4. Select `Language Translation Wizard` there
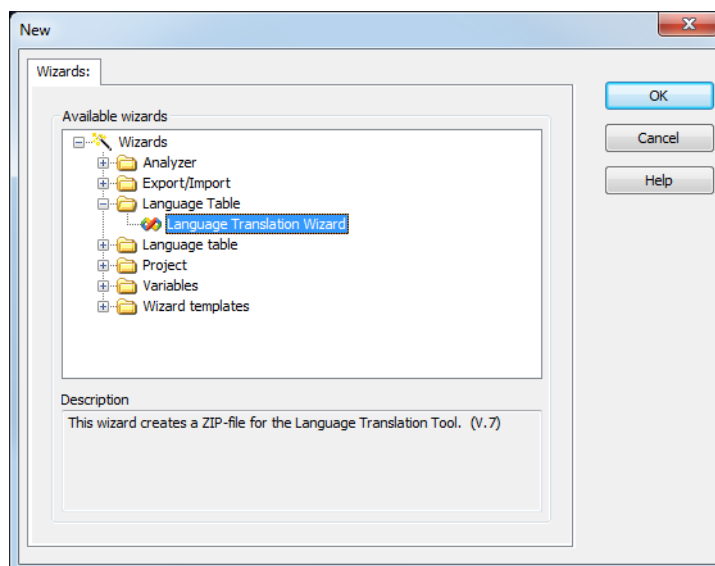
5. click on `OK`

6. The wizard starts with the welcome page
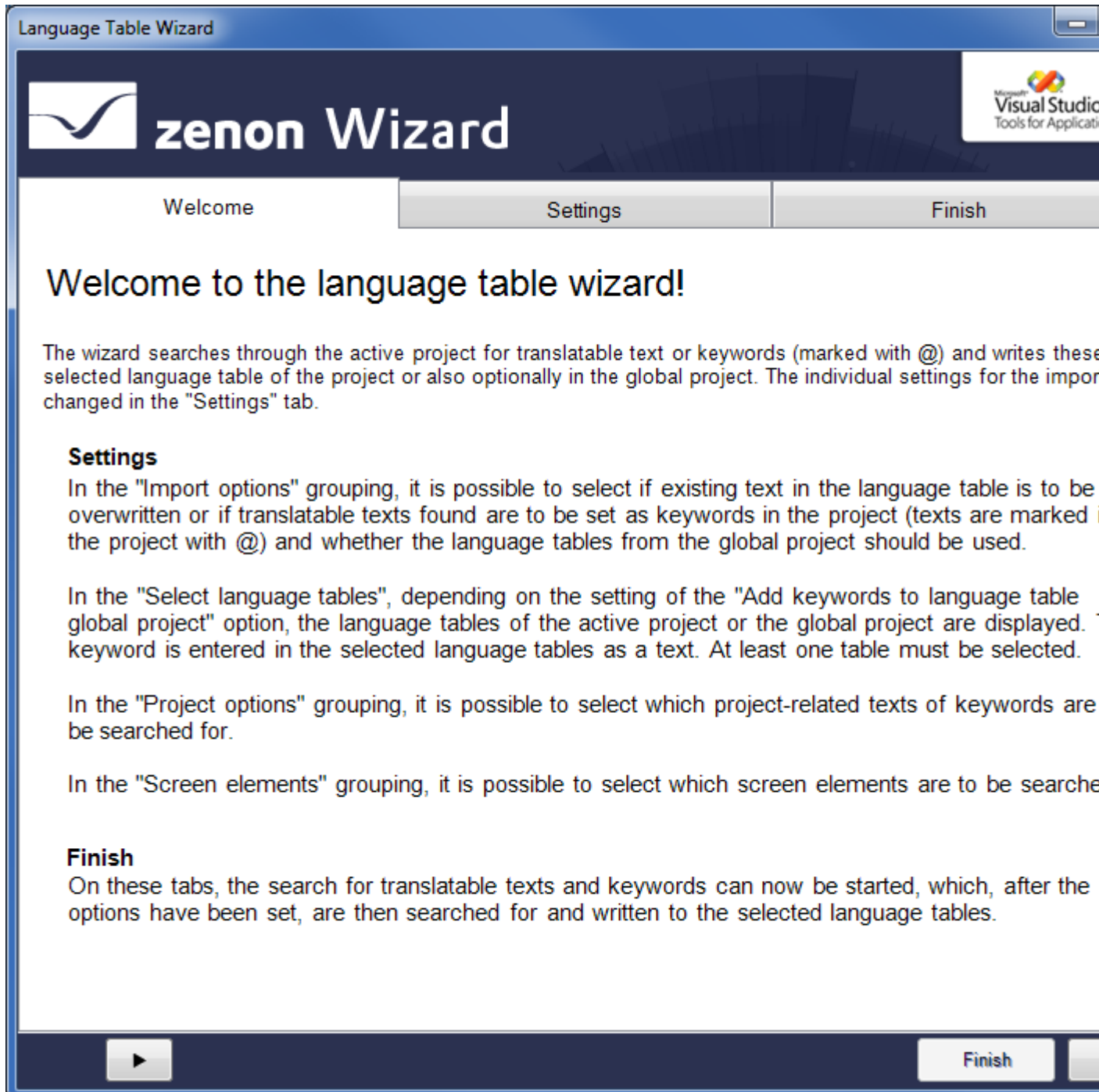
Start window

When opening the wizard, you get an overview page with English-language documentation for the wizard.
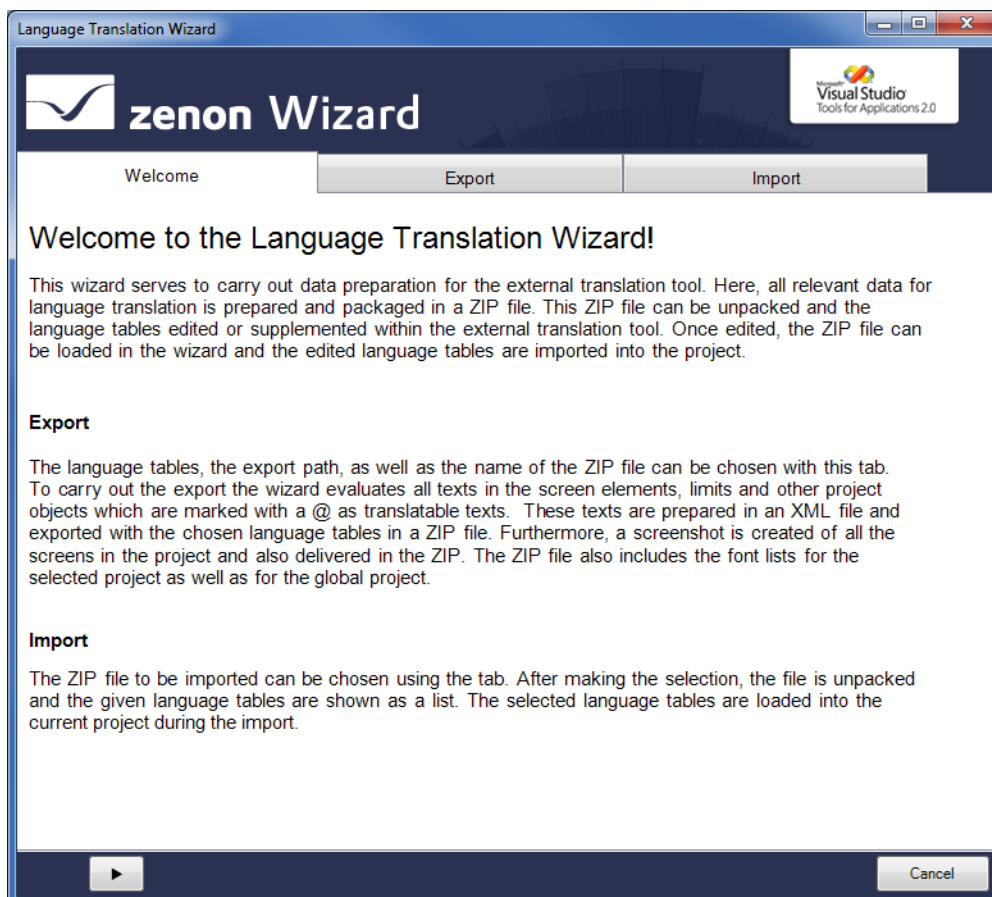
The navigation through the wizard is done by clicking on the individual tabs or step by step by clicking on the arrow keys.

Click on `Cancel` to close the wizard.

Export

When carrying out the export, the wizard evaluates all texts from screen elements, limits and other project objects, which are translatable text marked with a `@`.

These texts are prepared into an XML file and exported with the selected language tables as a ZIP file. In addition, a screenshot of all screens is created and also provided in the ZIP file. The ZIP file also contains the font lists of the selected project and those of the global project.
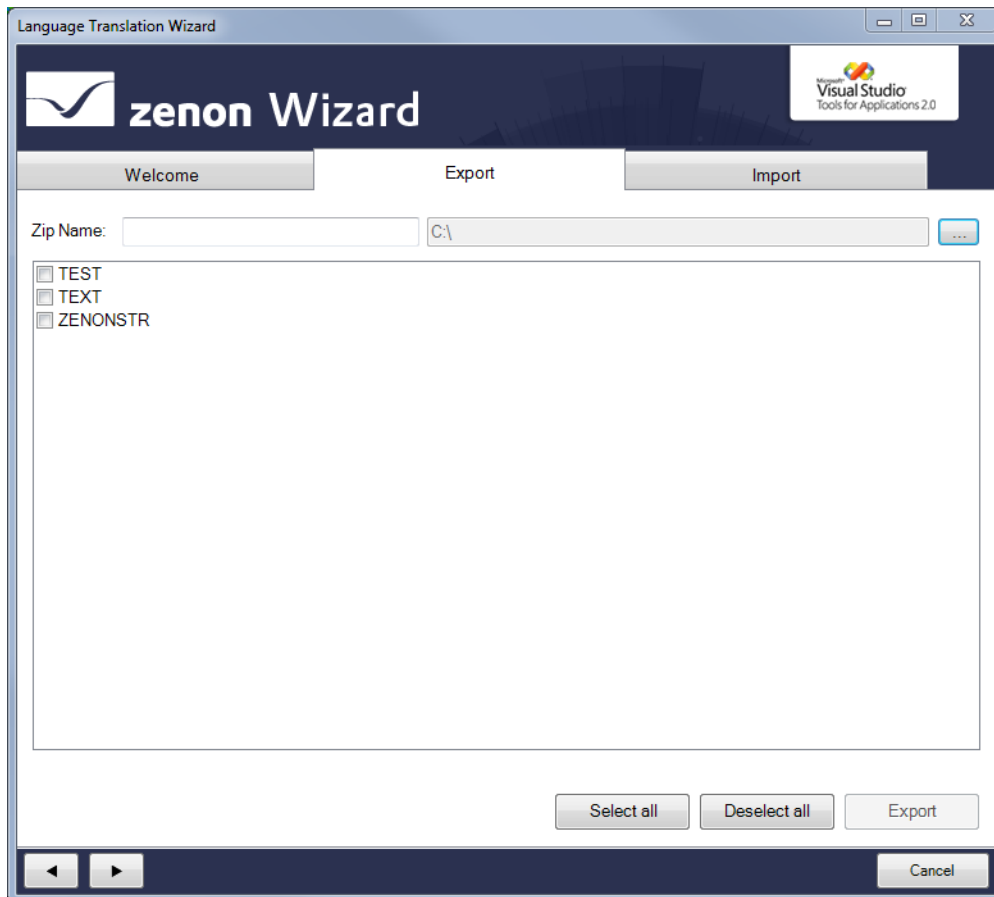
Button	Description
`Zip name:`	File name
`C:\`	Save location of the export file. A click on button `...` Opens the file browser for selecting a target folder.
`List of language files`	List field with the possibility to select the existing language files in the project.
`Select all`	Selects all entries in the list and activates the checkboxes.
`Deselect all`	Selects all entries in the in the list and deactivates the check boxes.
`Export`	Exports the files to the target folder. Is only active if one or more language tables are selected.
`Column left`	Goes back one tab in the wizard process.
`Column right`	Goes forward one tab in the wizard process.
`Cancel`	Closes the wizard without exporting.

> ⚠ **Attention**
>
> *The following special characters must not be used in the screen names:*
> *: / \ * < >*
>
> *These special characters are removed and replaced by a serial number.*

Import

The ZIP file to be imported can be selected in this tab.
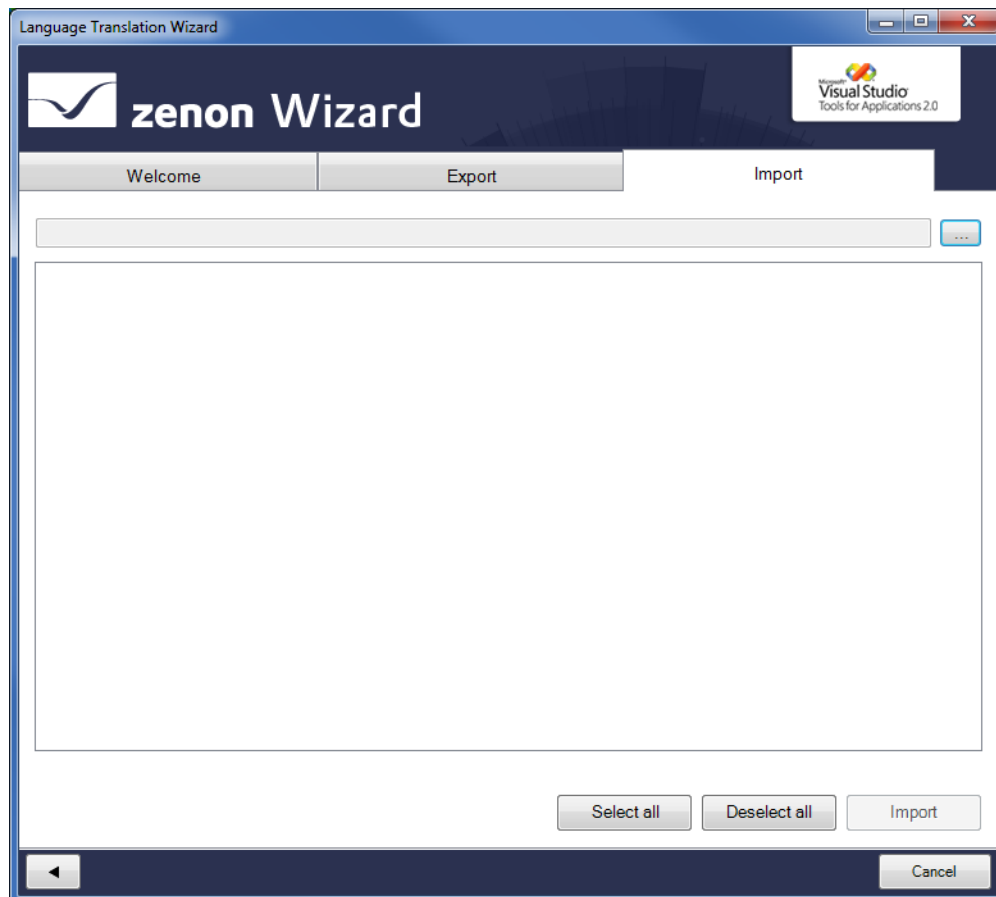
Button	Description
`...`	Save location of the import file. A click on button `...` Opens the file browser to select a file folder.
`List of language files`	List field with the possibility to select the language files to be translated and imported.
Select all	Selects all entries in the list and activates the checkboxes.
Deselect all	Selects all entries in the in the list and deactivates the check boxes.
Import	Imports the selected language tables in the current zenon project. Is only active if one or more language files are selected.
Column left	Goes back one tab in the wizard process.
Cancel	Closes the wizard without importing.

⚠ **Attention**

If a new language file was added in zenon 6.51, the project must be reloaded in the workspace once the wizard has been ended.

The language files are automatically updated in version 7.00 SP0 onwards.

System Text Wizard

The System Text Wizard allows the import of system text into the language table. System texts are Runtime texts that are used in zenon dialogs and menus and cannot be changed by the user. In order to be able to make these texts, which are predetermined by zenon, compatible with language switching, these texts must first be imported into the language table with this wizard. The number of the key words and texts imported can be limited in the settings.

Note: The wizard is only available in English.

Calling up the wizard

The wizard can also be selected directly in the dialog for starting wizards.
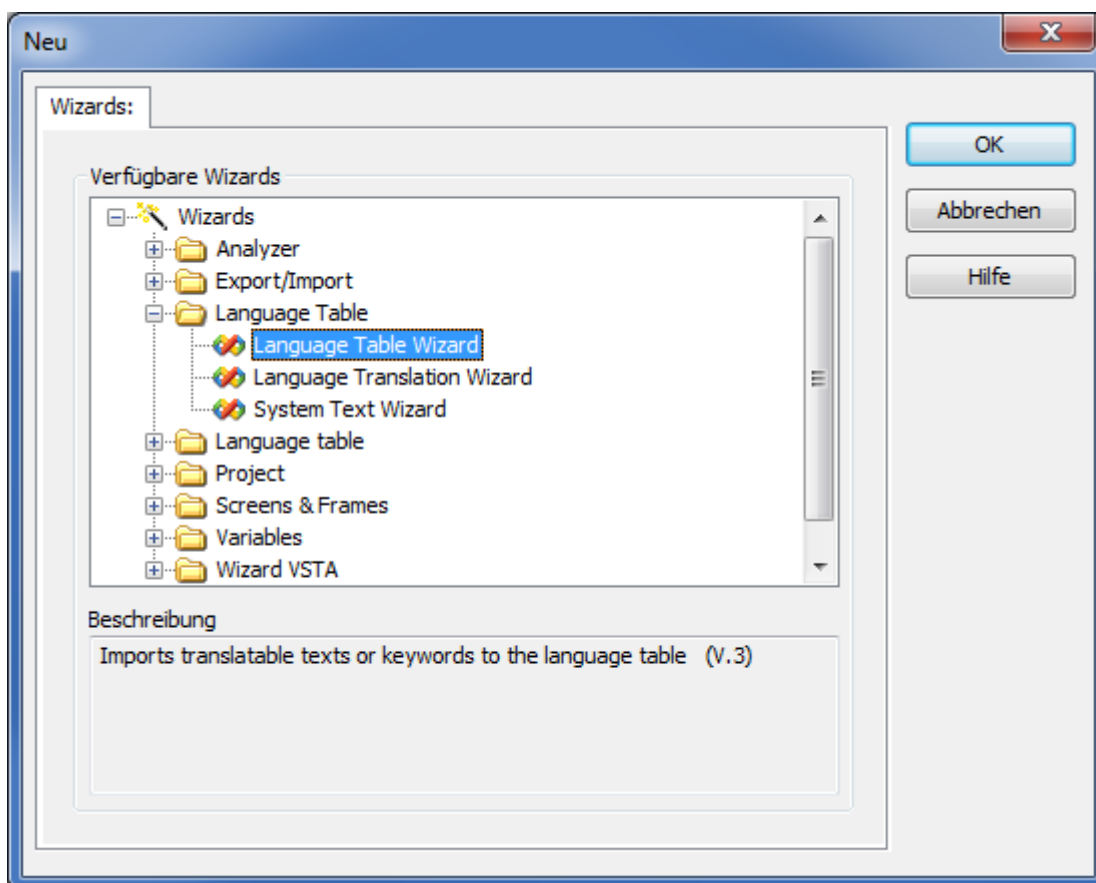
To start the wizard manually:

1. Click on *File-> Wizards...*

   or press the short cut `Alt+F12`

2. The selection window with the available wizards opens

3.   Select the **Language Table** folder

4.   Select **System TExt Wizard** there

5.   click on **OK**

6.   The wizard starts with the welcome page

Start window

When opening the wizard, you get an overview page with English-language documentation for the wizard.
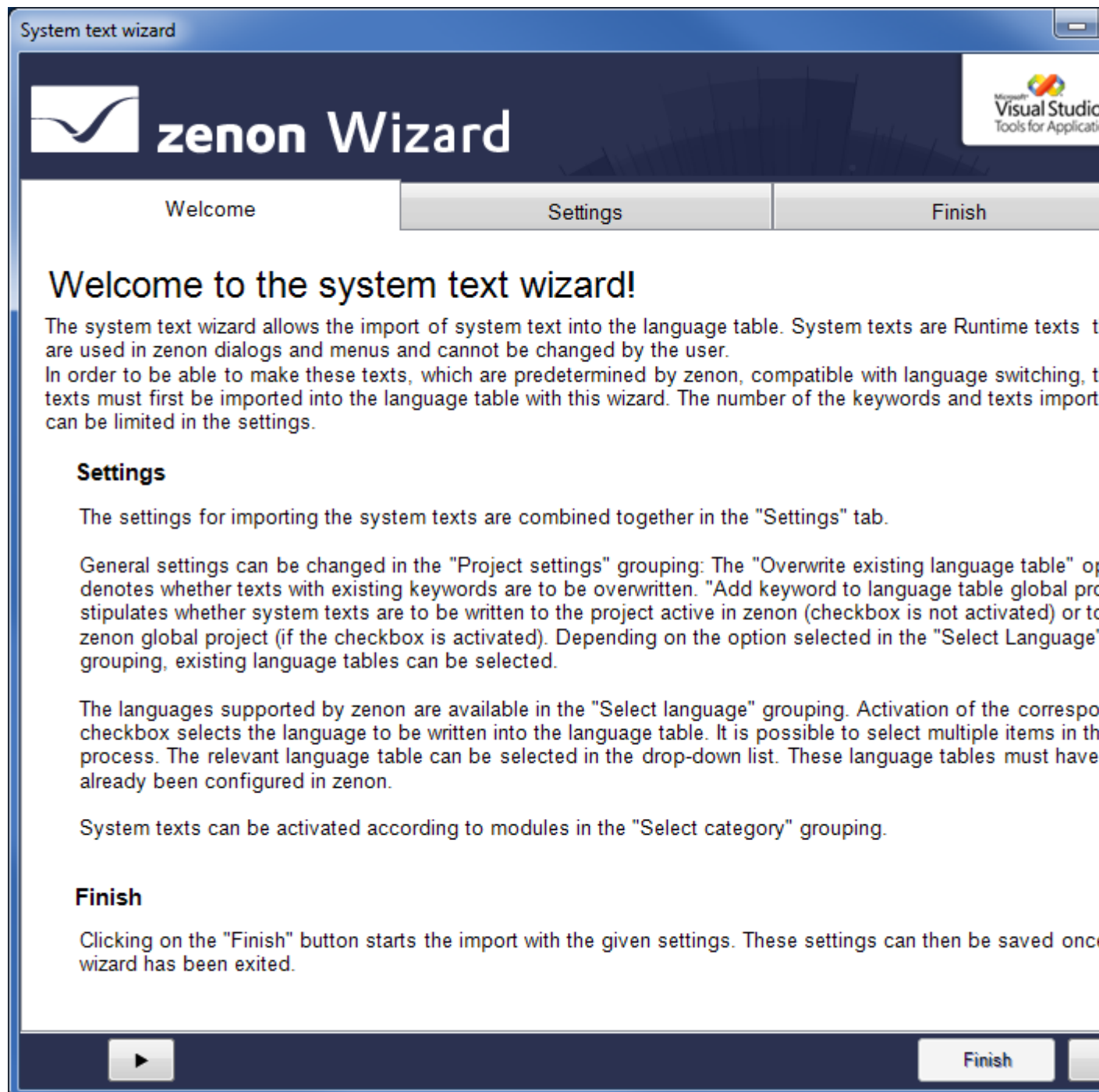
The navigation through the wizard is done by clicking on the individual tabs or step by step by clicking on the arrow keys.

Click on **Cancel** to close the wizard.

Settings

The settings for importing the system texts are combined together in the "Settings" tab.

PROJECT SETTINGS

General settings can be changed in the "Project settings" grouping: The "`Overwrite existing language table`" option denotes whether texts with existing key words are to be overwritten. "`Add keyword to language table global project`" stipulates whether system texts are to be written to the project active in zenon (checkbox is not activated) or to a zenon global project (if the checkbox is activated). Depending on the option selected in the "Select Language" grouping, existing language tables can be selected.

SELECT LANGUAGE

The languages supported by zenon are available in the "Select language" grouping. Activation of the corresponding checkbox selects the language to be written into the language table. It is possible to select multiple items in the process. The relevant language table can be selected in the drop-down list. These language tables must have already been configured in zenon.

SELECT CATEGORY

System texts can be activated according to modules in the "`Select category`" grouping.

Finish

Clicking on the "Finish" button starts the import with the given settings. These settings can then be saved once the wizard has been exited.
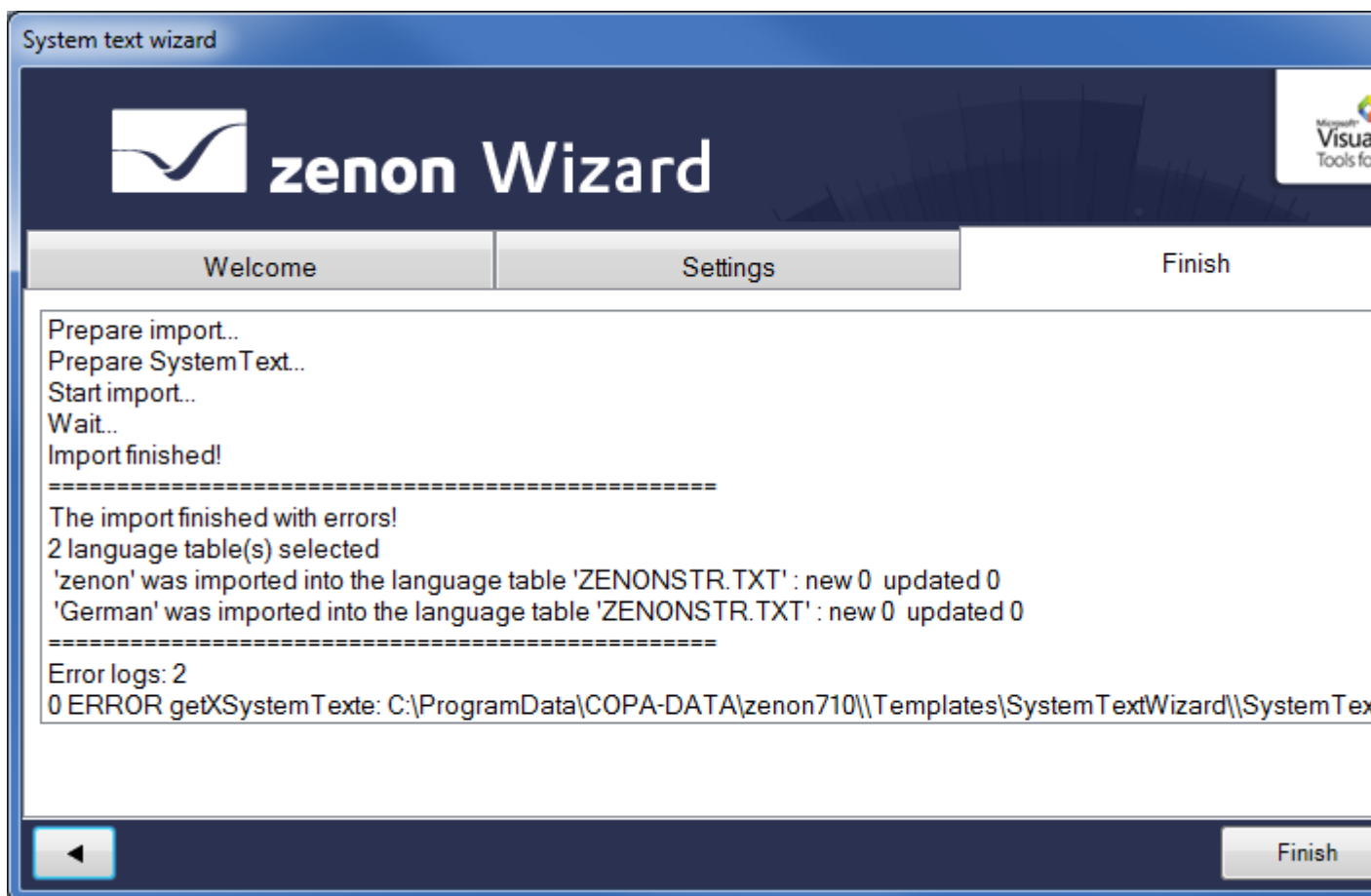
2.1.4   Pharmaceutical

Wizards for the pharmaceutical industry.

Pharmaceutical Wizard

The pharmaceutical wizard enables the management of validated projects for the pharmaceutical industry. It summarizes the relevant settings necessary for a Good Manufacturing Process (GMP) project. These settings can be managed and changed in the wizard. The settings are loaded into the wizard either via an existing project or via a configuration files.

The following settings are managed:

- ▶ General project settings

- ▶ User administration

- ▶ User groups

- ▶ Settings for the Chronologic Event List

- ▶ Settings for the Alarm Message List

- ▶ XML template

Engineered settings in the wizard can be:

- ▶ written back to the active project

- ▶ saved in a new project

- ▶ saved to a special configuration file
  These configuration files created in the wizard can be used over and over again and can be enhanced. However they can be only read and edited with the wizard.

Start wizard

To start the wizard:

1. Click on *File-> Wizards...*

   or press the short cut `Alt+F12`

2. The selection window with the available wizards opens

3. Select the `Pharmaceutical` folder

4. select the **Pharmaceutical Wizard** there
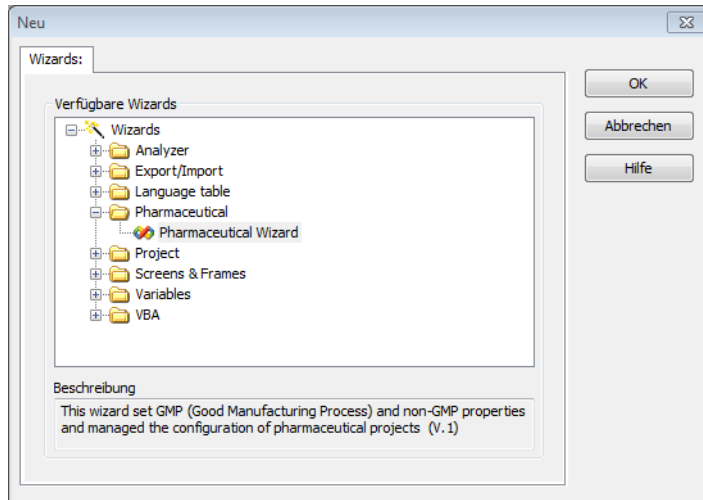
5. click on **OK**

6. The wizard starts with the welcome page

Welcome

Tab `Welcome` informs you about performance and use of the wizard.
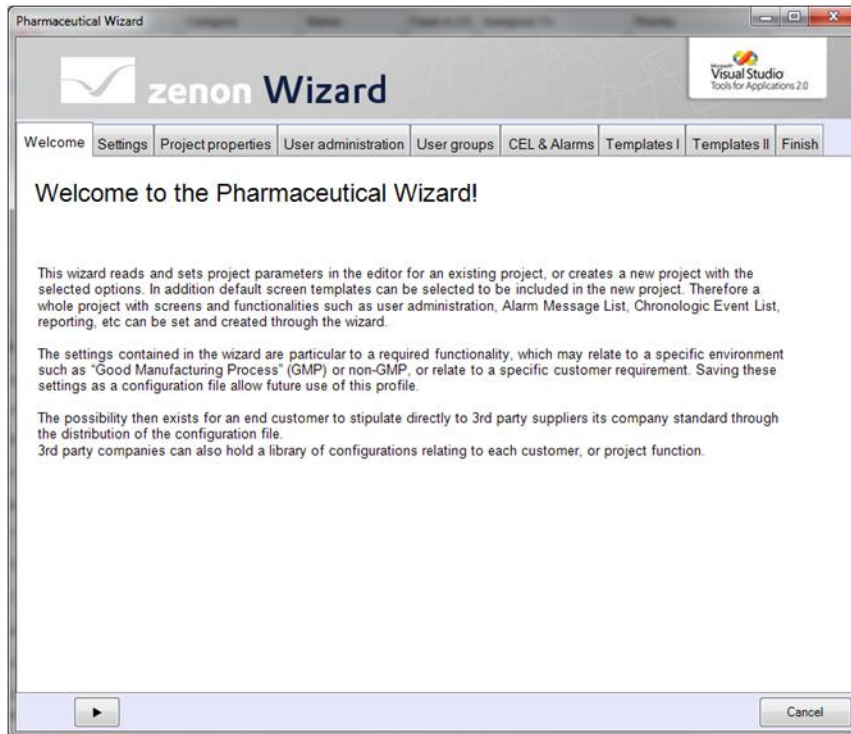
The navigation through the wizard is done by clicking on the individual tabs or step by step by clicking on the arrow keys.
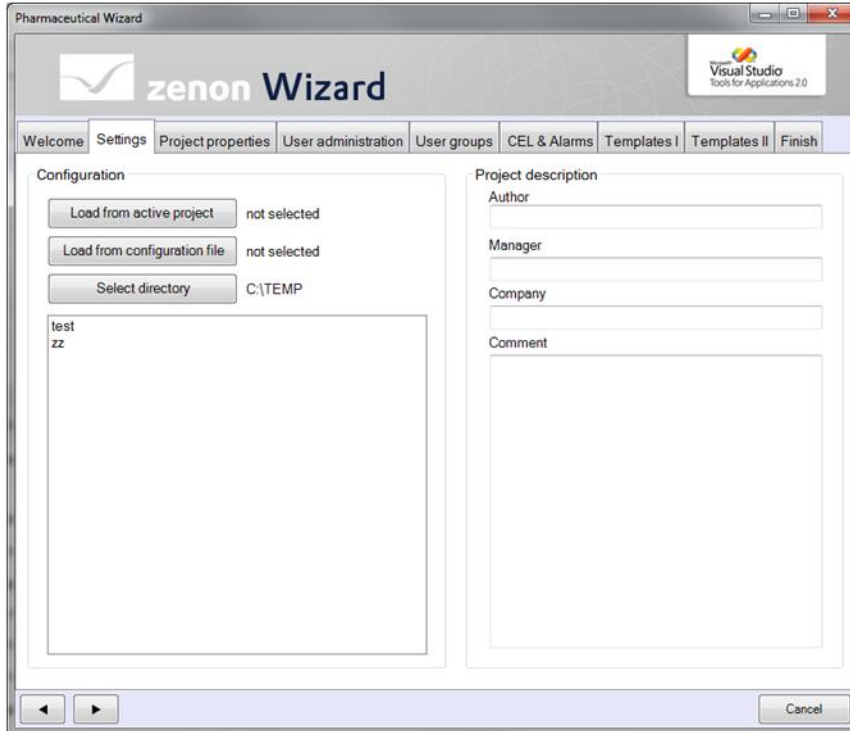
Click on `Cancel` to close the wizard. All changes made to a file or project since the last saving are discarded.

Settings

In this tab:

- ▶ Settings are loaded

▶ Project descriptions are adapted

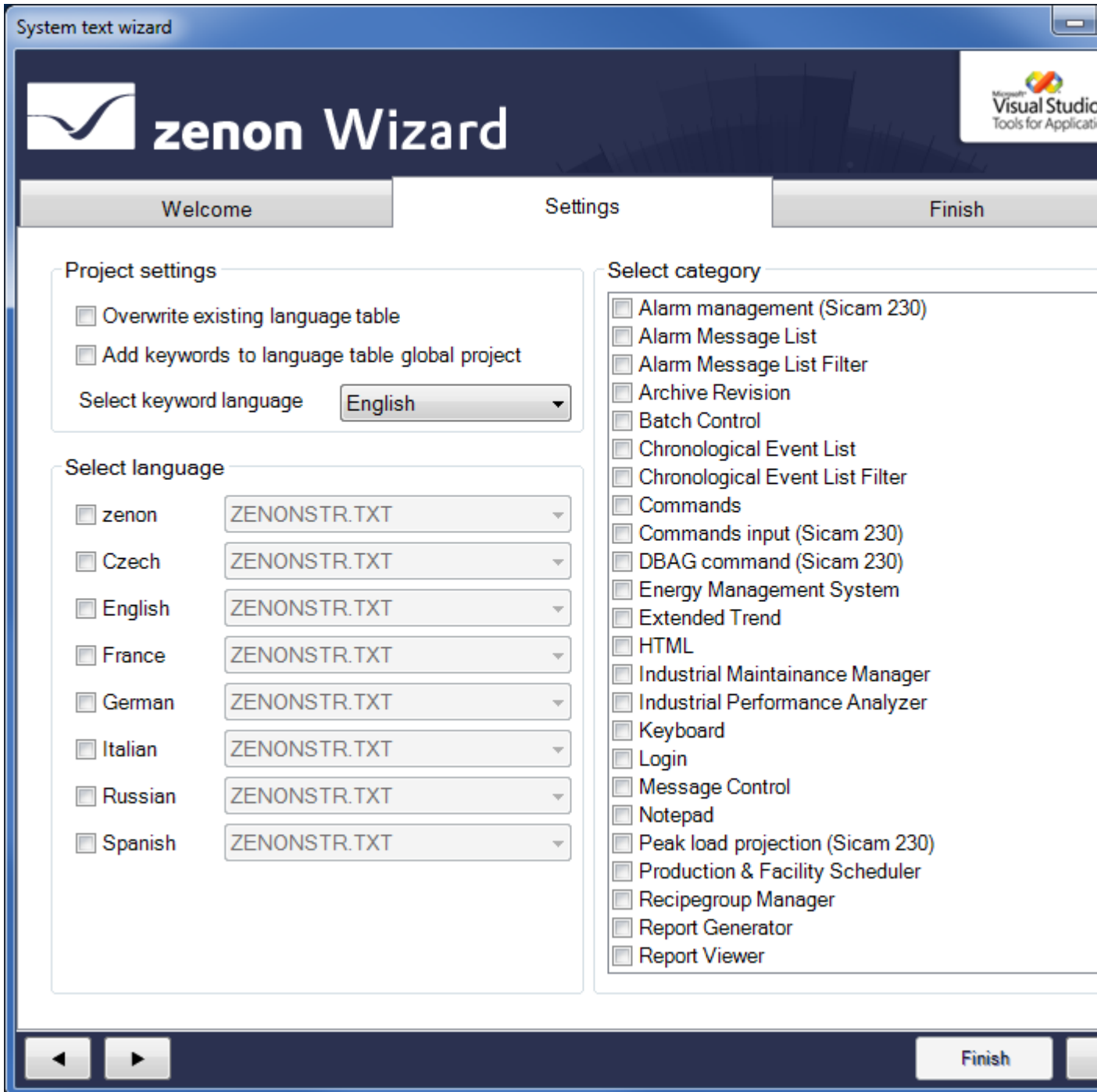| Parameters	Description
`Configuration`	Definition which project is edited.
`Load from active project`	Loads the settings of the active project in the workspace into the wizard.
`Load from configuration file`	Loads data from a saved configuration file (*.cof). The file is selected from the list. The list displays all available configuration files in the selected folder (`Select directory`). **Info:** The configuration files can only be read, created and edited with the wizard.
`Select directory`	Opens file selection dialog in order to select the folder in which the desired configuration files (*.cof) are available. They are displayed in the list below the button.
`Templates`	Settings for XML and XRS templates.
`Project description`	Information about the project as defined in property `Project description` of the dialog. It is taken over by the loaded project and can be edited. For checking purposes all changes are displayed on tab Finish (on page 125).
`Author`	Author of the project.
`Manager`	Project manager.
`Company`	Company.
`Comment`	Comment.

Project properties

Settings for:

- ▶ Network

- ▶ Authorization in the network

▶ History of changes

Parameters	Description
Network	Network Properties for the use of the project in a network. Read more in the online manual This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""
Active	Network active Active: The project is used in a network. A server has to be defined. Inactive: The project is a standalone project. Default: inactive. Read more in the online manual This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""
Server name	Server Only available if property Network active is active. Computer which establishes the connection to the hardware for network projects and which manages the project data. The clients connect to this computer. Hint name: The IP address is not sufficient; the name of the computer must be entered. "localhost" must not be used as name for the Server. If the name is changed, it cannot be reloaded. It is updated only after the Runtime has been restarted.

	More in the online help
	This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""

Standby name	
	Network active `Active`: The project is used in a network. A server has to be defined. `Inactive`: The project is a standalone project. Default: inactive. Read more in the online manual This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""
Redundancy type	
	Redundancy type Only available if a computer has been specified in property `Standby`. zenon supports two types of redundancy: `Software redundancy`: The system consists of one PLC and two redundant control system computers. Both computers must have a connection to the PLC. Both computers communicate with the control and at the same time keep the data from the control updated. The communication to the control is managed by the computer which is the server. The server communicates bidirectionally, the standby communicates unidirectionally. If the server crashes, the standby server takes over the bidirectional communication with the PLC. `Hardware redundancy`: The system consists of two redundant PLCs and two redundant control system computers. Each server communicates bidirectionally with one PLC. Both computers and both PLCs are synchronizing their data. If one component in the first system crashes, the second system takes over. Default: `Software redundancy` More in the online help This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""

`Network authorization`	`Authorization` Properties for the authorization in the network. Read more in the online manual This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""
`Active`	`Authorization in network active` Only available if property `Network active` is active. The authorization in the network makes sure that in the network only one station at a time can carry out active operations (e.g. change set values). Passiv, reading access is always possible regardless of the option. `Active`: Only on computer can operate the project at a time (e.g. acknowledge alarms, write set values). `Inactive`: Several computers can operate the project at the same time. Default: inactive. More in the online help This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""
`Timeout for request [s]`	`Timeout for request [s]` Only available if property `Authorization in network active` is active. If the authorization in the network is blocked by a station, it can be requested by another computer. Within the time limit defined here

	the request must be answered. After the time expires without an answer, the requesting station automatically receives the authorization. **More in the online help** This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""

`Timeout for acknowlewdgement [s]`	`Timeout for authorization [s]` Only available if property `Authorization in network active` is active. Within this time the computer (Client) which has the authorization must report to the Server. If this does not happen, the authorization is released automatically. **Attention:** This value must be smaller than the time defined in property `Timeout [s]`. More in the online help This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""
`History of changes`	`History of changes` Properties for the history of changes. Read more in the online manual This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""
`Active`	`History of changes active` `Active`: Changes in the project are logged. `Inactive`: Changes in the project are not logged. Default: inactive. Read more in the online manual This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""

`Detailing level`	`Detailling level` Only available if property `History of changes active` is active. Selection of details levels from drop-down list. ▸ `Object`: Only the object names of the changed objects are logged. Details concerning properties and their values are not displayed in the History of changes. ▸ `Properties`: Additionally to the object names the changed properties and the new values are displayed in the history of changes. ▸ `Value changes`: This setting causes the most detail level of logging. Not only the new value of a property is displayed but also the old one. This makes a complete tracability of the changes of values possible. Default: `Properties` More in the online help This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""

User administration

Settings for the user administration:

Parameters	Description
`User administration`	`User administration` Project specific properties for the user administration. Read more in the online manual This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""
`Delete users`	`Deleting users` `Active`: It is allowed to delete users in the Runtime. `Inactive`: It is only allowed to mark users as deleted. The users stay in the list of users but are no longer valid for Runtime operation (according to FDA). Default: active. Read more in the online manual This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""
`Minimum password length`	`Min. password length` Number of characters, a password must have as a minimum. Minimum: 0. Maximum: 20. Default: 6. Read more in the online manual This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""

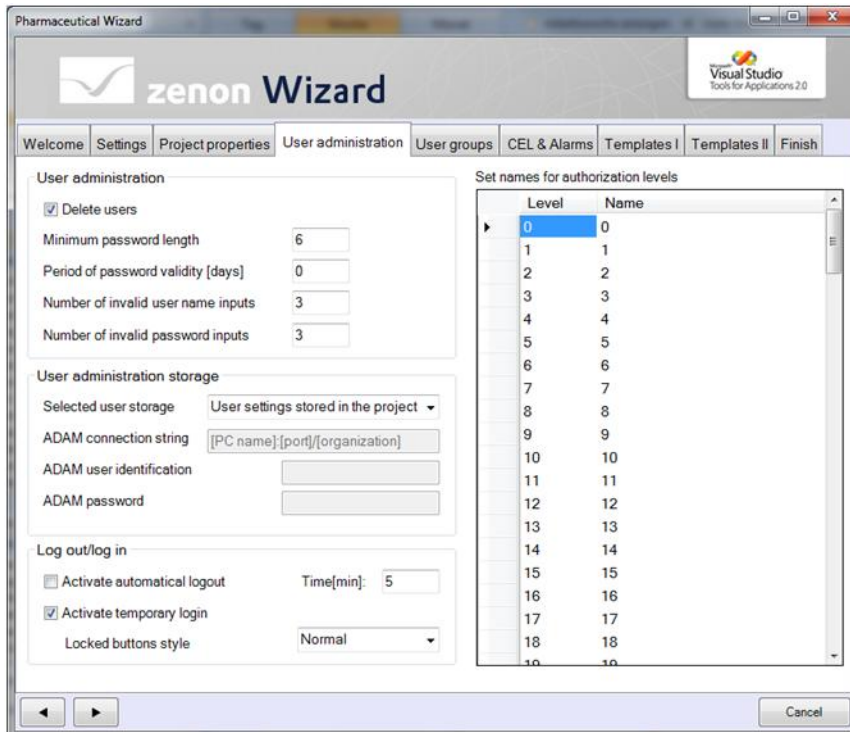|	

`Period of password validity [days]`	`Password - period of validity [d]`
	
	Enter a time period (in days) defining how long a password should stay valid. After the time expired, you must enter a new password.
	`Minimum:` 0 - The password never expires and need not be renewed. For this setting the value `2147483647` is written to system driver variable "**Days until password expires**".
	`Maximum:` **4294967295.**
	`Default:` `0.`
	Attention: For productions according to the FDA guidelines entry `0` is not allowed as the rules of the FDA demand a cyclic change of the password.
	
	More in the online help
	
	This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""

`Number of invalid user name inputs`	`Min. password length` Number of characters, a password must have as a minimum. Minimum: 0. Maximum: 20. Default: 6. Read more in the online manual This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""
`Number of invalid password inputs`	`Max. password error`

	Number of invalid password inputs.
	If this number is exceeded, the according user is locked. An according entry in the Chronologic Event List (CEL) is generated. He can only be unlocked by an administrator.
	Minimum: 0.
	Maximum: 65535.
	Default: 3.
	Read more in the online manual
	This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""

User administration storage	Saving the user administration
Selected user storage	Select where you want to save the user administration: ▸ Project ▸ Active Directory (AD) ▸ Active Directory Application Mode (ADAM)
ADAM connection string	ADAM/AD-LDS connection Connection path to ADAM or AD LDS. You must enter the connection path in the following form: **[PC name]:[port]/[organization]** Example: `w3k:50000/O=710,c=com` More about the AD LDS in the online help. More about ADAM in the online help. This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""
ADAM user identification	ADAM/AD-LDS user identification User name of a local user of the ADAM/AD LDS PC with administration rights. More about the AD LDS in the online help. More about ADAM in the online help. This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""

ADAM password	
	ADAM/AD-LDS password Password of the local user of the ADAM/AD LDS PC. More about the AD LDS in the online help. More about ADAM in the online help. This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""
Log out/log in	Properties for log in and log out.
Activate automatical logout	Activate automatical log out Active: The user is automatically logged out, if no user action happens for the defined period of time. Inactive: The user is not automatically logged off by the system. Default: inactive. Read more in the online manual This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""
Activate temporary login	Temp. login active Active: If a user is not logged in and tries to operate an element which needs a user authorization, he must enter his user name and password. Inactive: If a user is not logged in and tries to operate an element which needs a user authorization, he receives a message that he does not have the necessary authorization. Default: active.

	Note: You can define position and size of the login mask in file zenon6.ini in section **[Befehlsgabe]** Position =. More in the online help This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""
`Locked buttons style`	`Locked buttons` Only available if the property is `Temp. login active inactive.` Defines the look of buttons that are locked due to the configuration of the authorization. Is combined in the Runtime for the operation of keys with property `Interlocked buttons` (graphical design). Possible formats: ▶ `Grey` ▶ `Normal` ▶ `Invisible` Default: `Normal` More in the online help This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""
`Set names for authorization levels`	Conforms to property `Rename authorization levels.` You can assign a name to each of the 128 authorization levels. Click in the right-hand column in order to enable the renaming.

User groups

On this tab you assign authorization levels to user groups.

Parameters	Description
`User groups`	List of user groups
`User group levels`	List of the authorization levels
`Delete`	Deletes selected object without confirmation message.

Click on a user group in order to show its authorization level. Click on the level in order to edit it. The name is automatically adapted. Click on an empty level in order to insert a new authorization level.

When clicking on `Delete`, you can delete both the groups and the assigned authorization levels. At deleting no confirmation message is displayed.

CEL & alarms

Configuration of the Chronological Event List and the Alarm Message List.

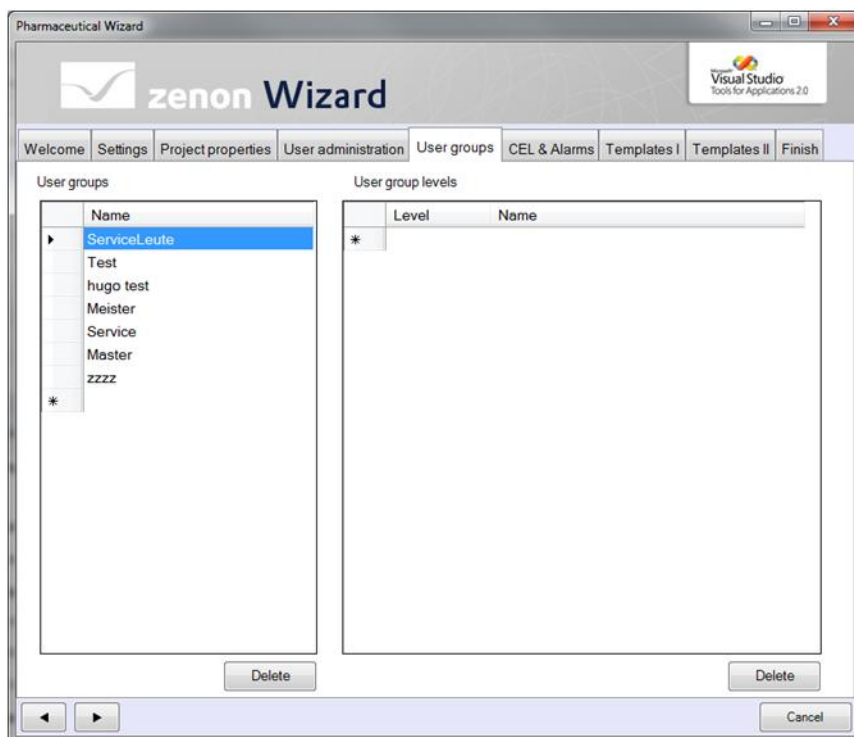| Parameters	Description
Chronlogical Event List and Alarm Message List	Chronologic event list Properties for the Chronologic Event List (CEL) configuration. Read more in the online manual This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""
CEL active	CEL active Active: The Chronological Event List (CEL) is active in the Runtime. Events are recorded and the CEL is available. Inactive: No event are recorded. Default: active Note: Changes take effect after the Runtime has been restarted. More in the online help This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""
Update automatically	Update automatically Only available if property CEL active is active. Active: During the time the CEL is opened in the Runtime, new events are immediately added when they occur.

	`Inactive`: As long as the CEL is opend no new entries are added. The new entries are added when the CEL is opened the next time. Default: `active` **More in the online help** This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""

`Print system messages`	
	`Print system messages` Only available if property `CEL active` is active. `Active:` At online printing system messages are also printed. `Inactive:` At online printing system messages are not printed. Default: `active` More in the online help This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""
`Long dynamic limit texts CEL`	
	`Long dynamic limit texts CEL` Determines whether the comment field for dynamic limit texts is available. The dynamic limit allows you to include the current values of other variables in the limit text of a variable. `Active:` Dynamic contents will be stored in a file with the file format D*.CEL It will be stored in addition to the file C*.CEL. The comment filed can therefore been used for comments. Dynamic limit texts may have a maximum length of 254 characters. `Inactive:` The comment field is used for dynamic limit texts and is therefore not available for comments. Maximum length: 80 character. Default: `inactive`. More in the online help This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as

	follows: ""

Signature text editable	
	Signature text editable `Active`: In the Runtime a dialog for editing the signature text is opened. `Inactive`: The signature text cannot be edited in the Runtime. Default: inactive. Read more in the online manual This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""
Data storage CEL	
	Data storage CEL Properties for the storage of the Chronologic Event List (CEL) entries in the memory and on the harddisk. Read more in the online manual This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""
Size of the ring buffer	
	Size of the ringbuffer Only available if property `CEL active` is active. Size of the CEL ring buffer. If the ring buffer overflows (cel.bin), the entries are transferred to the CEL archive (*.cel). Minimum: `1` Maximum: `32767` Default: `100`

	Note: In the Runtime it is possible that more entries are displayed than you engineered as old entries are only removed from the CEL when the list is updated. More in the online help This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""

`Save CEL data`	
	`Save CEL data` Only available if property `CEL active` is active. `Ring buffer and historic data`: All CEL entries (*.cel) are saved. `Only ring buffer`: Only a defined number of CEL entries (cel.bin) is saved. The number is defined via property `Size of the ringbuffer`. `Default`: On CE devices only the ring buffer (cel.bin) is saved on the hard disk; on PCs the historic entries (*.cel) are also saved. Default: `Default` The files (cel.bin and *.cel) are saved in directory \project directory\computer name\project name. **More in the online help** This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""
`Save ring buffer on change`	
	`Save ringbuffer on change` Only available if property `CEL active` is active. `Active`: Each change of the data of the Chronological Event List (CEL) triggers the saving of the data (cel.bin). **Note:** With this the whole ring buffer is saved at every saving. This can - especially for Flash Disks - lead to substantial system load. `Inactive`: Data of the CEL (cel.bin) are only when the Runtime is closed or when function Save AML and CEL ring buffer is executed. Recommended especially for low performance. Historic data (*.cel) can be saved independently at every value change.

	Default: `inactive`
	More in the online help
	This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""

`Logging`	
	`Logging` General properties for the Chronologic Event List (CEL). Read more in the online manual This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""
`Alarm acknowledgement`	`Alarm acknowledgement` Only available if property `Alarm Message list active` is active. `Active`: If an alarm is acknowledged, an entry is created in the Chronological Event List (CEL). `Inactive`: Acknowledging an alarm does not trigger an entry in the CEL. Default: `inactive`. More in the online help This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""
`Function Set SV`	`Function Set SV` Only available if property `CEL active` is active. `Active`: At successful writing of values to the hardware, a corresponding entry is entered in the CEL. `Inactive`: The successful writing of values is not logged in the CEL. Default: `inactive` **Hint:** To use these properties you must carry out write set

	value via function write set value.
	More in the online help
	This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""

Send recipes	
	Send recipes Only available if property CEL active is active. No logging: Writing standard recipes and recipes of the Recipegroup Manager (RGM) is not logged in the CEL. Log recipes: When writing a recipe, it is logged with the name of the recipe in the CEL. Log recipes and values: When writing recipe, it is logged with the name of the recipe and the name and the old and new values of the variables in the CEL. Default: no logging **More in the online help** This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""
Change recipes	
	Change recipes Only available if property CEL active is active. No logging: Changing standard recipes and recipes of the Recipegroup Manager (RGM) is not logged in the CEL. Log recipes: When changing a recipe, it is logged with the name of the changed recipe in the CEL. Log recipes and values: When changing a recipe, it is logged with the name of the changed recipe and the name and the old and new values of the changed variables in the CEL. Default: no logging **More in the online help** This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""

Alarm groups	Alarm/event groups
Add	Adds the character string entered in the input field as new alarm/event group. Adding can also be carried out via key `Enter`.
Remove	Deletes highlighted alarm/event group.
Alarm classes	Alarm/event classes
Add	Adds the character string entered in the input field as new alarm/event class. Adding can also be carried out via key `Enter`.
Remove	Deletes highlighted alarm/event group.

Templates I

On this tab you can edit the following elements:

- ► Screens
- ► Data types
- ► Color palettes

▶ Reaction matrices
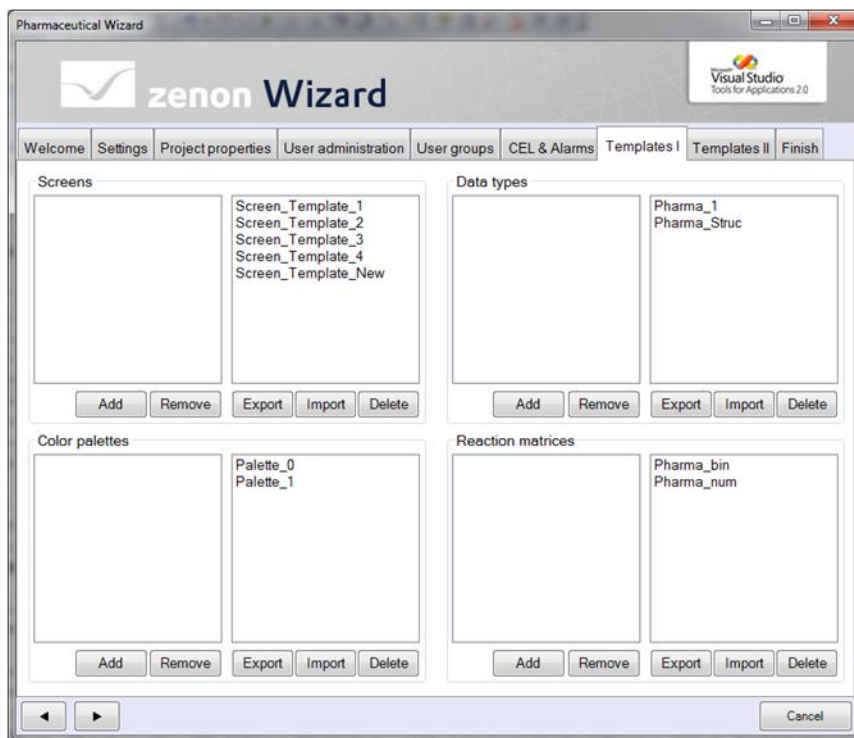
In each right-hand list the XML files, which exist in the current configuration file for the element, are displayed and edited.

Buttons	Function
Right-hand list	
Export	Opens the dialog for exporting XML files which were created in the Editor via command **Exported selected XML**.
Import	Opens the dialog for importing XML files which were created in the Editor via command **Exported selected XML**. Import is carried out in folder `%ProgrammData%\Copa-Data\zenon700\Templates\PharmaWizard`.
Delete	Deletes XML files which were created in the Editor via command **Exported selected XML**.
Left-hand list	
Add	Adds templates from the left-hand list. As an alternative you can also carry out a double click on the template. Saving to the current project or to another project is carried out on tab Finish (on page 125).
Remove	Removes template from the list.

Templates II

On this tab you can edit the following elements:

▶ Symbols

▶ Reports from the Report Generator
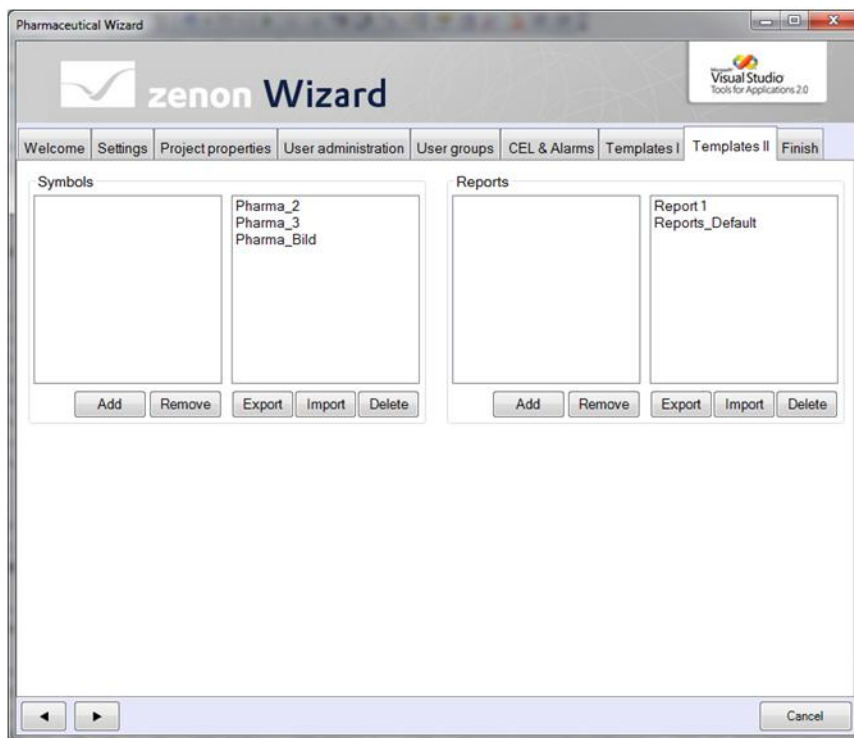
In each right-hand list the XML files, which exist in the current configuration file for the element, are displayed and edited. Reports are saved as XRS files.
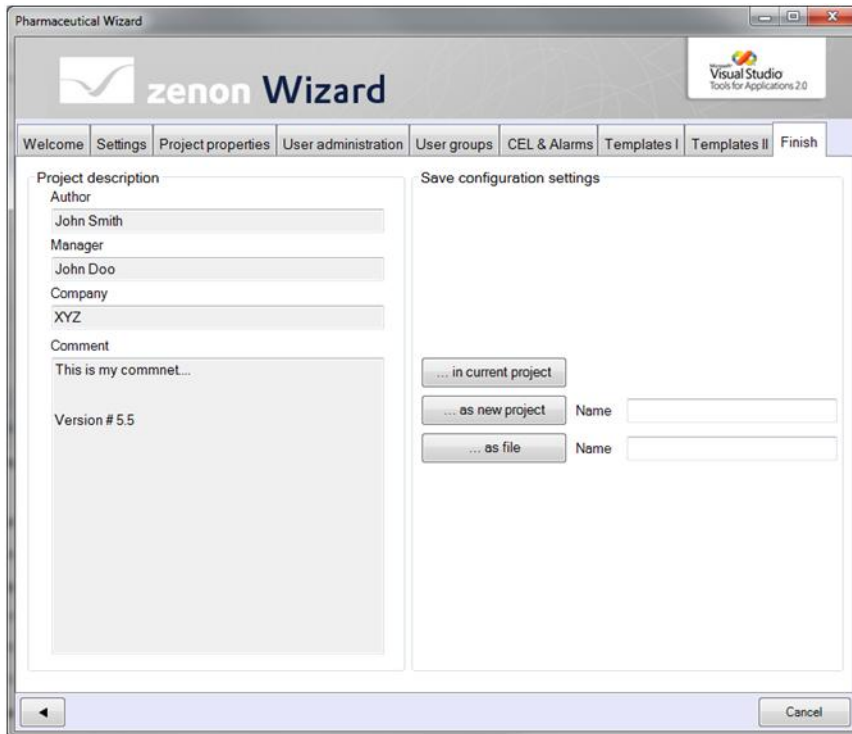
Buttons	Function
`Right-hand list`	
`Export`	Opens the dialog for importing XML files which were created in the Editor via command **`Exported selected XML`**.

`Import`	Opens the dialog for importing XML files which were created in the Editor via command **`Exported selected XML`**.
	`C:\\ProgrammData\Copa-`
	`Data\zenon700\Templates\PharmaWizard.`

`Delete`	Deletes XML files which were created in the Editor via command **`Exported selected XML`**.
`Left-hand list`	
`Add`	Adds templates from the left-hand list. As an alternative you can also carry out a double click on the template. Saving to the current project or to another project is carried out on tab Finish (on page 125).
Remove	Removes template from the list.

Finish

In this tab:

▶   the project description is entered

▶ you define how the changed engineering is saved

Parameters	Description
Project description	Information about the project as defined on tab Settings (on page 90). Display only, cannot be changed.
Author	Name of the engineer.
Manager	Name of the manager.
Company	Company.
Comment	Comments to the project.
Save configuration settings	Options for saving changes done by the wizard.
Message field	Messages about success/failure of save actions.
... in current project	All settings are loaded in the current project. With this the settings in the project are overwritten.
... as a new project	A new project is created with the defined settings and the selected frames. You must first select a project name.
... as file	A new configuration file of file name *.cof is created. If the name of an existing configuration file is used it is overwritten. **Attention:** The created configuration files can only be read, created and edited with the wizard.
Name	Name of the configuration file.

2.1.5   Project

Wizards for:

▶   Project creation (on page 141)

▶   the compare of project backups (on page 128)

▶   the creation of a project documentation (on page 164)

Project comparison

The wizard makes it possible to compare project backups. At this it is analyzed which objects and elements were deleted, added and/or changed. The result can be saved and displayed as XML or HTML file.

REQUIREMENTS

The wizard can read in and compre project backups which:

- ▶ were saved as zip file

- ▶ were created with activated versioning

- ▶ were created with activated XML export

To activate versioning and XML export:

1. open the `General` node in project settings.

2. go to section `Versioning`

3. Activate the `Versioning active` property

4. Activate the `XML export active` property

TEMPORARY FILES

During the compare the wizard unzips the project backup in the temporary fodler `BackUpComparisonWorkingFolder`. It is created in path `C:\Users\Public\Documents\zenon_Projects\Worspace`. This temporary folder is deleted when the wizard is closed.

Start wizard

To start the wizard:

1. Click on *File-> Wizards...*

 or press the short cut `Alt+F12`

2. The selection window with the available wizards opens

3. Select the **Project** folder

4. select the **Backup Comparison Wizard** there
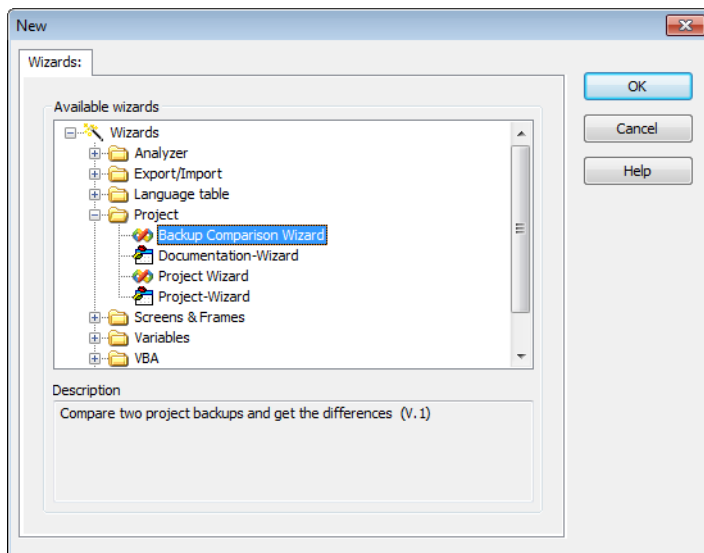
5. click on **OK**

6. The wizard starts with the welcome page

Welcome

Tab `Welcome` informs you about performance and use of the wizard.
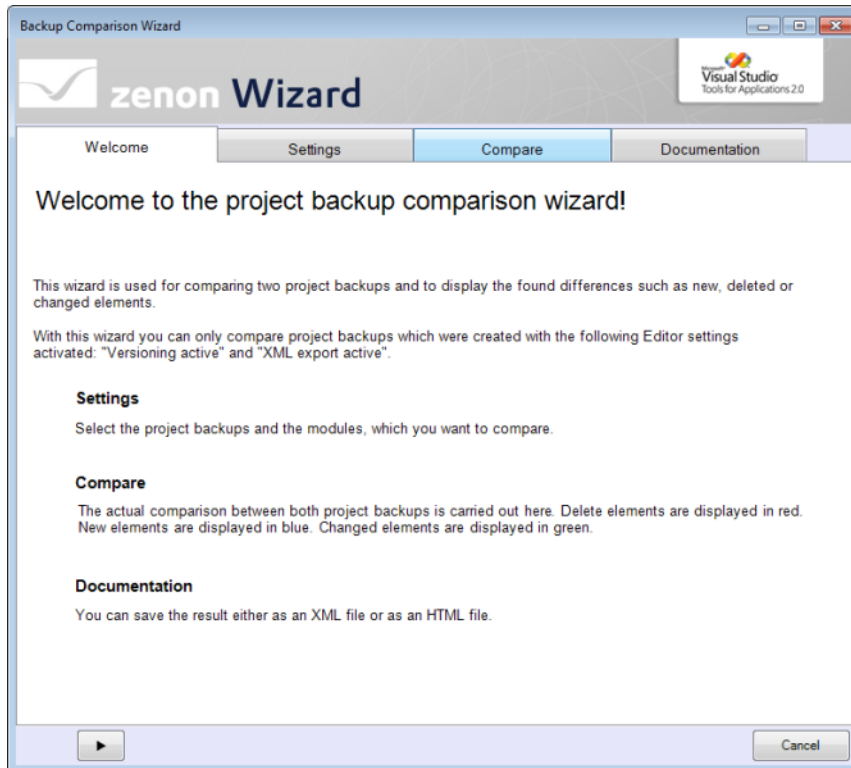
The navigation through the wizard is done by clicking on the individual tabs or step by step by clicking on the arrow keys.

Click on `Cancel` to close the wizard.

Settings

On this tab the two project backups which should be compared are selected and the zenon modules which should be part of the compare.
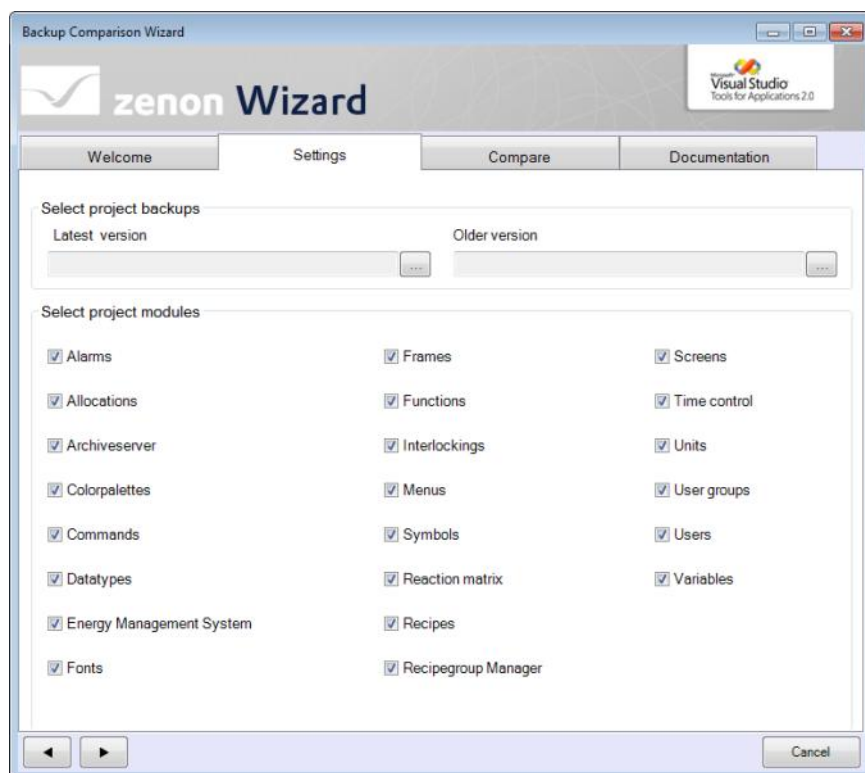
Parameters	Description
`Select project backups`	Selection of the backup files.
`Latest version`	Latest version. A click on button `...` Opens the file browser to select a project backup. Note: Per default project backups are in folder `%ProgramData%\COPA-DATA\SQL2008R2\BACKUP\[Project];` via export however they can be stored in any folder.
`Older version`	Older version. A click on button `...` Opens the file browser to select a project backup.
`Select project modules`	Selection of the modules which should be compared. Selection takes place via activating the checkboxes in front of the module names. These settings are saved for each user individually and are available when the wizard is opened again.
`Cursor keys`	Click on the button to go to the previous or next tab.
`Cancel`	Closes the wizard.

Compare

On this tab the project backups are compared on basis of the selected modules.

Parameters	Description
`List box`	After you click on button **`Compare`** the differences of the project backups are displayed in the list field. <u>Content of the columns:</u> ▸ **`Modules`**: Name of the module. ▸ **`Result`**: Name of the object. ▸ **`Element`**: Display and description of the element. ▸ Parameter: Name of the changed parameter. ▸ **`Old`**: Previous value of a changed parameter. ▸ **`New`**: New value of a changed parameter. <u>Color-coded marking:</u> ▸ `blue`: new objects and elements ▸ `red`: deleted objects and elements ▸ `green`: changed objects and elements ▸ `black`: unchanged objects and elements
`Navigation`	Elements for the navigation in the list.
`Changed elements`	`Active`: Click on the button with the vertical arrow in order to jump to the previous/next changed element.
`Deleted elements`	`Active`: Click on the button with the vertical arrow in order to jump to the previous/next deleted element.
`New elements`	`Active`: Click on the button with the vertical arrow in order to jump to the previous/next new element.
`Show only differences`	`Active`: After you click on button **`Compare`** only the differences are displayed color-coded; unchanged elements are not displayed. If this option is changed, you must start the compare again by clicking **`Compare`**.
`Compare`	Compares the project backups in accordance with the selection and displays them in the list field.
`vertical double arrow buttons`	Click on the button to jump to the previous/next module.

Vertical arrow buttons	Click on the button to jump to the previous/next result of the same type depending on the setting.
	▶ Changed elements
	▶ Deleted elements
	▶ New elements
Horizontal arrow keys	Click on the button to go to the previous or next tab.
Cancel	Closes the wizard.

EXAMPLE PROJECT COMPARE:

ALL OBJECTS:

Some changes:

In module screens:

▶ the screen **batch** was deleted

▶ in screen `SYSTEM_2` a text element was changed

▶ in screen `START` a button named Button_1 was added

▶ in screen `ALARM` nothing was changed

ONLY DIFFERENCES:
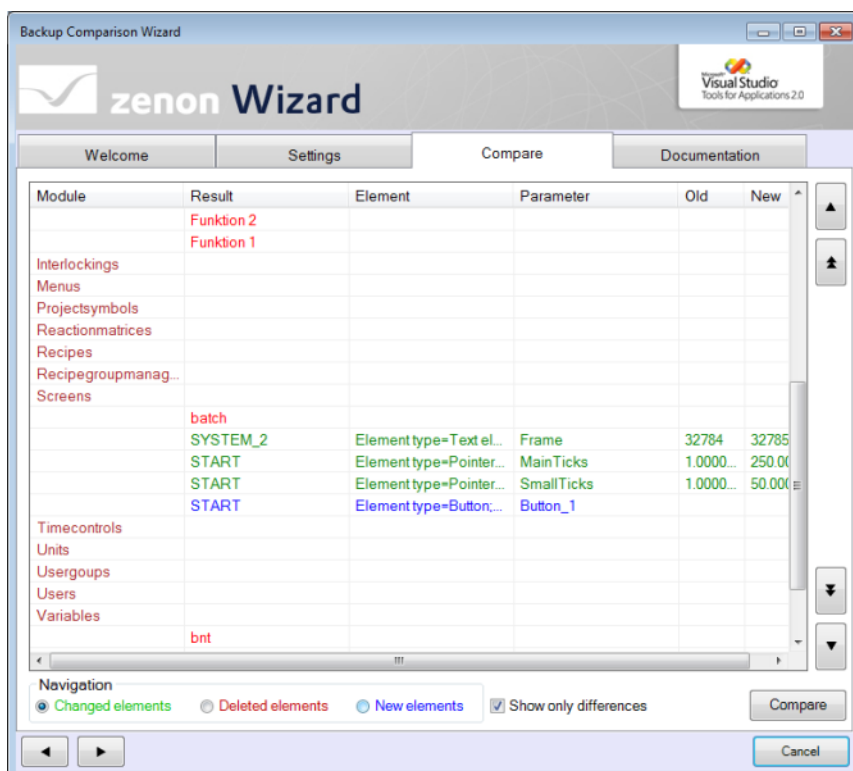
The changes are visible in the same way as in the previous screenshot. Objects and elements which have not been changed are hidden.

Documentation

On this tab you can display and save the result of the project backup compare as XML file or HTML file.
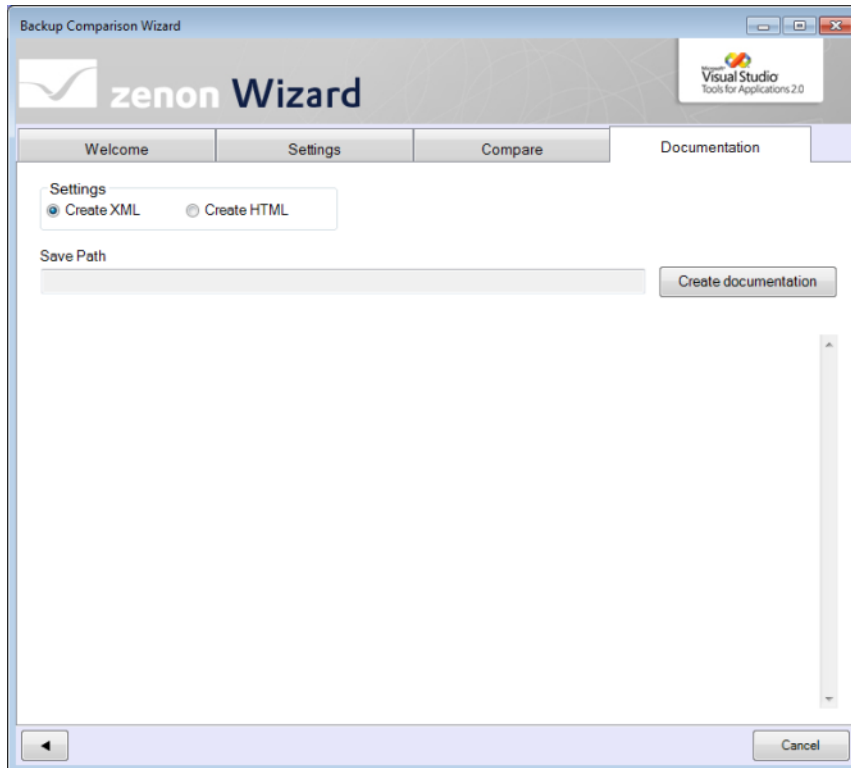
Parameters	Description
`Settings`	Settings for type of documentation.
`Create XML`	`Active`: An XML file is created.
`Create HTML`	`Active`: A HTML file is created.
`Save Path`	Path to the folder in which the file is saved. Display only. Selection is carried out via button `Compare`.
`Create documentation`	Click on button: ▸ to open the file browser: Select the saving location and give a name to the documentation file. ▸ The documentation is saved in the desired type and is displayed in the list field.
`List field`	Display documentation.
`Arrow button`	Click on the button to go to the previous tab.
`Cancel`	Closes the wizard.

EXAMPLE XML FILE

Display in the wizard:

Display as XML file:

EXAMPLE HTML FILE

At the creation of an HTML file, an XML and an XSLT file are also created at the saving location. These two files are used to generate the HTML file dynamically:
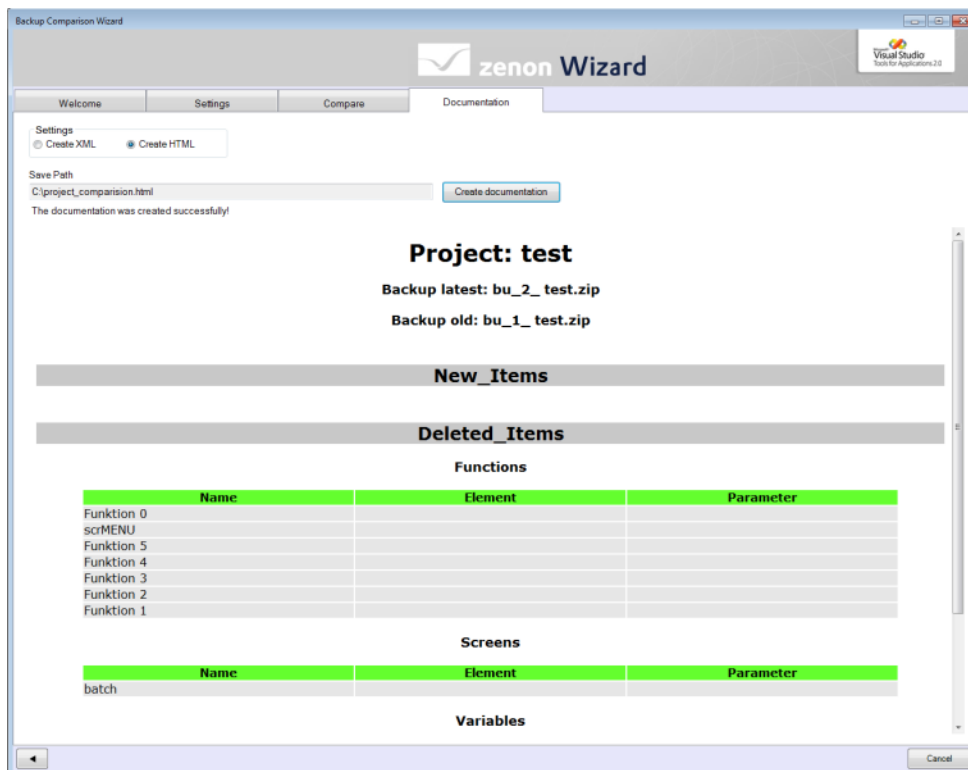
Structure of the HTML file:

Project Wizard

With this wizard you can create basic objects for new projects. You can configure:

▶   Information about the project

▶   Drivers also with driver variables

▶   Graphics settings

▶   Basic screens with symbols, WPF element, AML, CEL, system information and an overview of the simulation variables

▶   Navigation

Settings changed in the wizard are saved in the user profile of the operating system and loaded at the next opening by the same user.

The wizard is executed in English; the language in the project corresponds to that of the open zenon Editor. This wizard is automatically executed when a new project is created.

Start wizard

This wizard is automatically executed when a new project is created. It can also be selected directly in the dialog for starting wizards.

Attention: If the wizard is started in an existing project, existing objects may be changed or overwritten.

To start the wizard manually:

1.   Click on *File-> Wizards...*

 or press the short cut `Alt+F12`

2.   The selection window with the available wizards opens

3. Select the `Project` folder
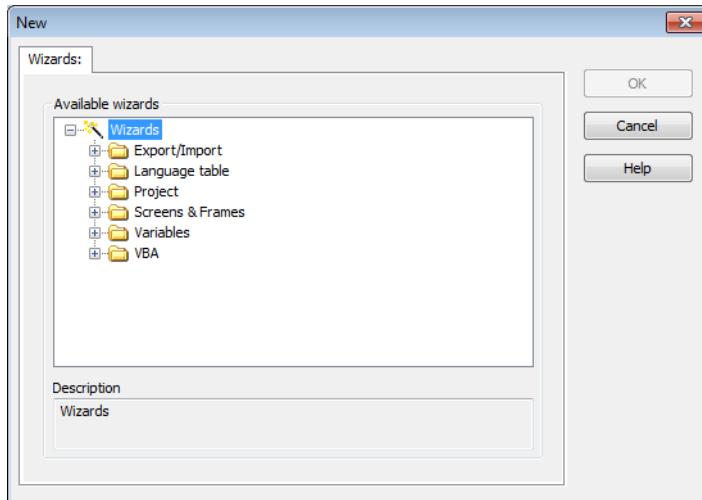
4. Select the `VSTA Project Wizard` there
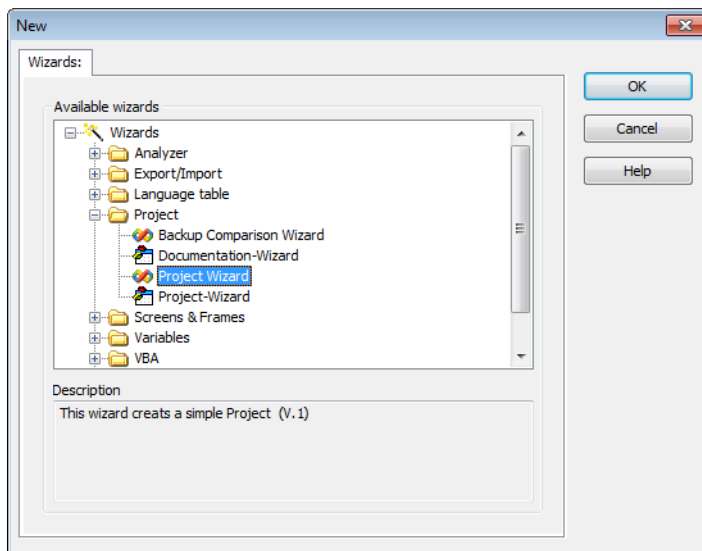
5. click on `OK`

6. The wizard starts with the welcome page

If you call up the wizard from an existing project, you receive a warning:
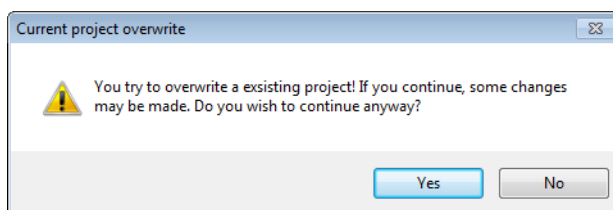
By clicking on `Yes`, you confirm that you accept changes to your existing configuration.

Welcome

Tab `Welcome` informs you about performance and use of the wizard.
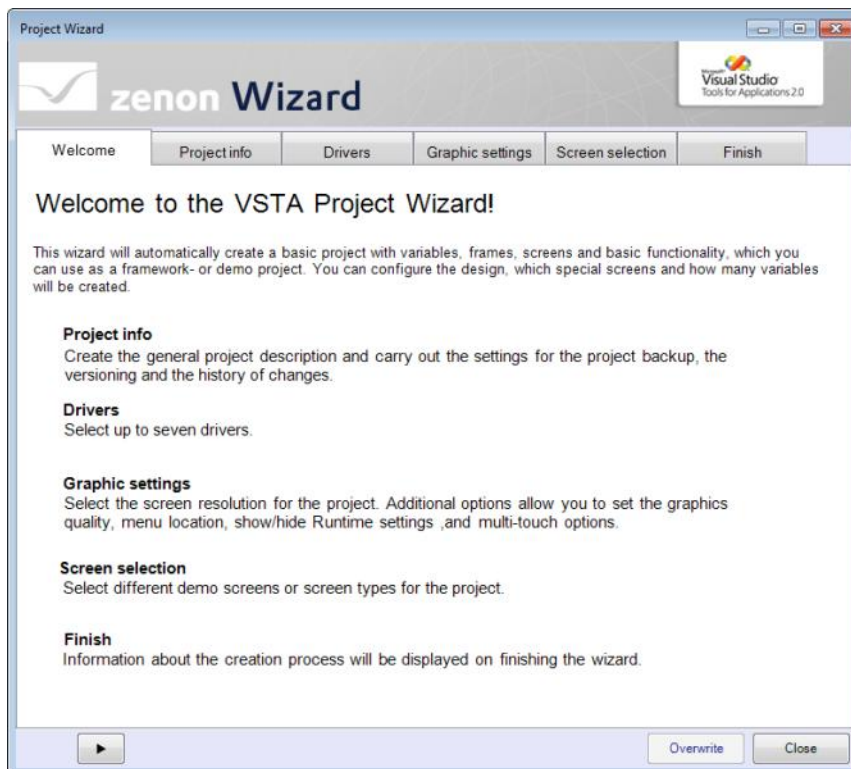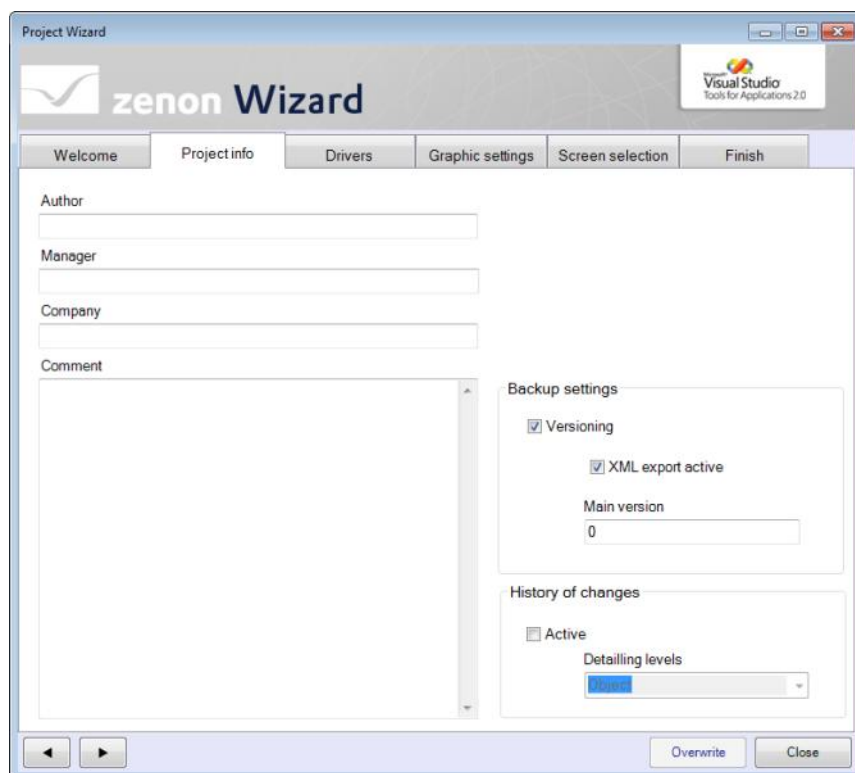
The navigation through the wizard is done by clicking on the individual tabs or step by step by clicking on the arrow keys.

Click on `Close` to close the wizard.

Click on `Overwrite` to create the project and overwrite possibly existing objects. Only active if tab `Finish` is opened.

Project info

On this tab you configure general project settings and settings concerning versioning and history of change.

Parameters	Description
Author	Author of the project.
Manager	Responsible manager.
Company	Company.
Comment	Comments to the project.
Backup settings	Settings for versioning.
Versioning	Versioning active Active: Project versioning is used. Every project backup is saved with an own version number. Inactive: no versioning of the project backup. Default: inactive More in the online help. This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""
XML export active	XML export active Only available if property Versioning active is active. Active: At each project backup an zip file (**version.zip**) is inserted. It includes 24 XML files with the backups of the individual modules. **Note:** For multi-user projects only for local backups. Default: inactive More in the online help. This function is to be addressed for the zenon API (with class

	name) and in XML export (without class name) as follows: ""

History of changes	Settings for history of changes.
Active	`History of changes active` `Active`: Changes in the project are logged. `Inactive`: Changes in the project are not logged. Default: inactive. Read more in the online manual This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""
Detailing levels	`Detailling level` Only available if property `History of changes active` is active. Selection of details levels from drop-down list. ▸ `Object`: Only the object names of the changed objects are logged. Details concerning properties and their values are not displayed in the History of changes. ▸ `Properties`: Additionally to the object names the changed properties and the new values are displayed in the history of changes. ▸ `Value changes`: This setting causes the most detail level of logging. Not only the new value of a property is displayed but also the old one. This makes a complete tracability of the changes of values possible. Default: `Properties` More in the online help This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""

Cursor keys	Move one tab forward or back.
Overwrite	Creates project and overwrites possibly existing object. Only active if tab **Finish** is opened.
Close	Closes the wizard. The made changes can be saved for the current user.

Drivers
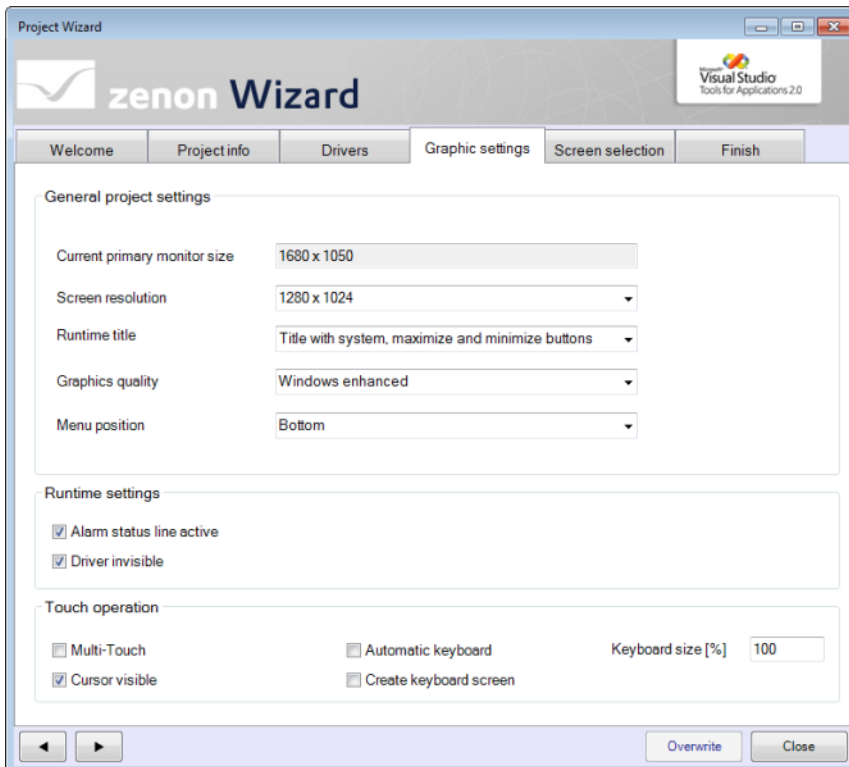
On this tab the necessary zenon drivers are selected.

Parameters	Description
Driver list	List of zenon drivers. Selection is done by clicking on the driver.
Driver information	Display of the information about the selected driver.
Driver name	Name of the driver how it should be displayed in the project. Free text input. Identification Distinct name of the driver. zenon identifies the driver with the help of this name and not with the help of the file name. Thus it is possible to load the same driver several times. This is for example necessary when you must access same control types which are connected to differenent serial interfaces. Attention: This property is not available for language switch. More in the online help This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""
Add	Adds the selected driver to the list of driver to be created.
Remove	Removes the selected driver from the list drivers to be created.
List of drivers to be created	List of drivers to be created with name, description and file name.
Create screen with driver statistic variables	Active: For each driver a screen with driver variables is created.
Cursor keys	Move one tab forward or back.
Overwrite	Creates project and overwrites possibly existing object. Only active if tab Finish is opened.
Close	Closes the wizard. The made changes can be saved for the current user.

Graphic settings

On this tab you configure the resolution and the settings for the Runtime and touch screens.

Parameters	Description
`General project settings`	General graphical settings for the project.
`Current primary monitor size`	Display of the current screen resolution. For multi-monitor systems the resolution of the main screen is displayed.
`Screen resolution`	Selection of desired screen resolution from drop-down list.
`Runtime title`	`Driver invisible` `Active`: Started drivers are not displayed in the Windows task bar in the Runtime. `Inactive`: Started drivers are displayed in the Windows task bar in the Runtime. Default: `inactive`. More in the online help This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""
`Graphics quality`	`Graphics quality` Setting for the quality of the graphics display. `DirectX` allows a higher quality than `Windows Basic` or `Windows Enhanced`. DirectX display is only supported in the Runtime. Generally speaking `DirectX Hardware` is preferable and `DirectX Software` should only be used if necessary. Possible selection: `Windows Basic`: Basic graphics settings. Recommended for resource-weak hardware. `Windows Enhanced`: Provides enhanced functions for the graphical display - needs more resources.

	`DirectX Software:` Graphics calculation is done by the CPU and can lead to high CPU load. `DirectX Hardware:` A part of the graphics calculation is done by the graphics card. If the system does not support the setting, it automatically switches to `DirectX Software.` Default: `Windows Enhanced` **Attention:** `Windows Enhanced` and `DirectX` are not available under Windows CE. **Hints:** When switching the mode during the engineering, there can be slight pixel deviation. There set this property before you create screens. At activating `Windows Base` for all line types which use `Line width [Pixel]` > 1, all line types are set to `solid line.` More in the online help This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""

Menu position	Selection of the menu location from drop-down list.
Runtime settings	Settings for the Runtime.
Alarm status active	Status line active Only available if property Alarm Message list active is active. Active: As soon as an alarm occurs, a red status line with alarm information is displayed at the top of screen in the Runtime. In this status line the alarm can also be acknowledged with a double right click if the logged in user has the corresponding rights. Inactive: No status line is displayed. Default: active *Attention multi-project administration: The setting in the integration project defines the behavior for sub-projects, regardless of the setting of the sub-projects. The alarm status line of the uppermost project is always used in Runtime.* More in the online help This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: "" Note: This wizard automatically activated property Alarm Message list active.
Driver invisible	Driver invisible Active: Started drivers are not displayed in the Windows task bar in the Runtime. Inactive: Started drivers are displayed in the Windows task bar in the Runtime. Default: inactive. More in the online help This function is to be addressed for the zenon API (with class

	name) and in XML export (without class name) as follows: ""

Touch operation	Settings for the touch operaton
Multi-Touch	**Multitouch active** `Active`: Multi-Touch can be used. Flicks (short swipe) and right click (touch and hold) are deactivated. Requirements: All corresponding driver and devices are available. The device must be connected and switched on. More in the online help. This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""
Mouse pointer visible	**Cursor visible** `Active`: The cursor in the Runtime is visible. `Inactive`: The cursor in the Runtime is invisible. So in projects with touchscreen the cursor can be switched off. Default: active. `ATTENTION`: This functionality only works in elements of the control system (screens, elements, ...) but not with Windows standard elements (title bars, menus, scroll bars, ...). For this the cursor has to be deactivated directly in the operating system. Read more in the online manual This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""
Automatic keyboard	**Automatic keyboard** For projects with touchscreen. `Active`: A virtual keyboard is automatically opened, whenever a

	user input is necessary (e.g. setting values or logging in). `Inactive:` No virtual keyboard is opened. Default: inactive. Read more in the online manual This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""
`Create keyboard screen`	`Active:` Creates **DIALOGKBD** for alphanumeric input and **SETVALUEKBD** for numeric input. For details see chapter Create screen of type keyboard. For it to be used in the Runtime, you must activate option `Automatic keyboard`.
Keybooard size (%)	 `Keyboard size [%]` Defines in which size in percent - starting from the original size - the automatic keyboard should be displayed in the Runtime. Minimum: `50 %` Maximum: `300 %` Default: `100 %` More in the online help This function is to be addressed for the zenon API (with class name) and in XML export (without class name) as follows: ""
Arrow keys	Move one tab forward or back.
Overwrite	Creates project and overwrites possibly existing object. Only active if tab **Finish** is opened.
Close	Closes the wizard. The made changes can be saved for the current user.

Screen selection

On this tab you can select screens which should be created in the project.

Attention: The screen switch function to the selected screens are configured with the default settings. No special settings such as filter, variables, etc. are made.

Parameters	Description
`Screen types`	Configuration of the screen.
`Selection field`	Selection of the screen type from drop-down list.
`Frames list`	Selection of the frame for the screen. Size and preview are displayed.
`Width`	Width of the screen. Display only,
`Height`	Height of the screen. Display only,
`Preview`	Preview of the selected screen type with the selected frame. `Standard` is dispalyed as empty.
`Input field name`	Free label of the screen name.
`Add`	Adds screen with selected name to `list of screens`.
`Remove`	Removes selected screen from `list of screens`.
`Screens list`	Lists all configured screens with names, screen types and size. A maximum of 14 screens can be created.
`Demo project`	Settings for a example project.
`Create demo screens`	**Active**: Exemplary screens are created for the engineering. Selection of the demo screens: ▸ Demo pages: Example pages (are always created) ▸ Alarm Message List: AML ▸ Chronological Event List: CEL ▸ System information: Pages with system information, number depending on the resolution
`Cursor keys`	Move one tab forward or back.
`Overwrite`	Creates project and overwrites possibly existing object. Only active if tab `Finish` is opened.
`Close`	Closes the wizard. The made changes can be saved for the current user.

EXAMPLE CONFIGURATION
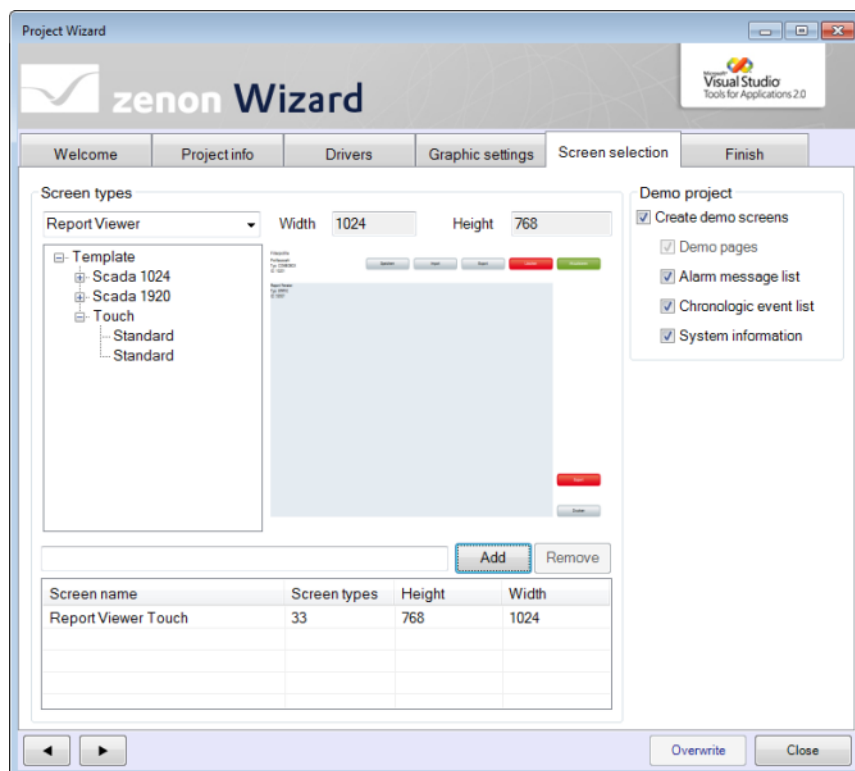
Finish

On this tab the project is created.
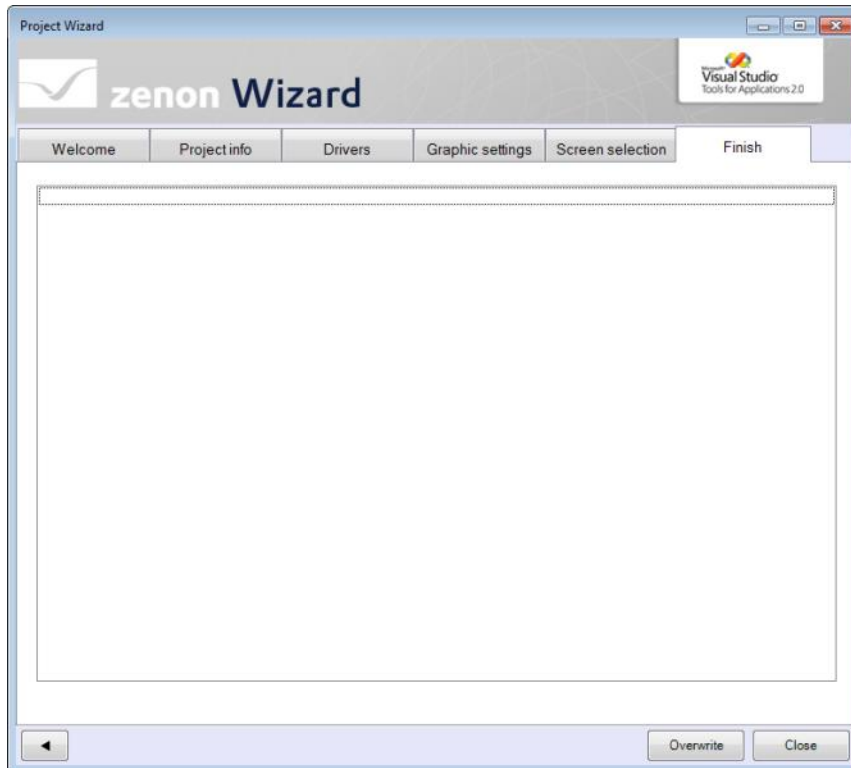
Click on button **Overwrite** to create the project according to the settings on the tabs. Possible already existing objects are overwritten. The wizard remains open for further configuration. To close the wizard, click on button **Close.**

Examples in the Runtime

Below you will find two example how your entry in the wizard effects the display in the Runtime.

▶ Example 1: (on page 161) With active option `Create demo screens` (on page 157).

▶ Example 2 (on page 163): Without example screens.

With demo screens

Start page:

Navigation:

SIMUL information:
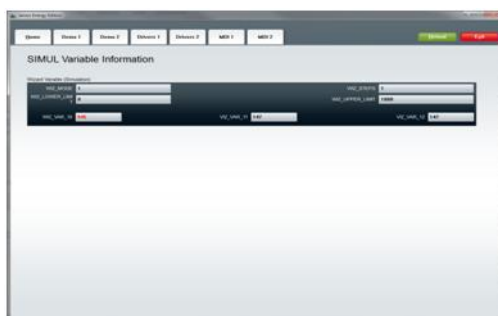
Report Viewer:

AML:
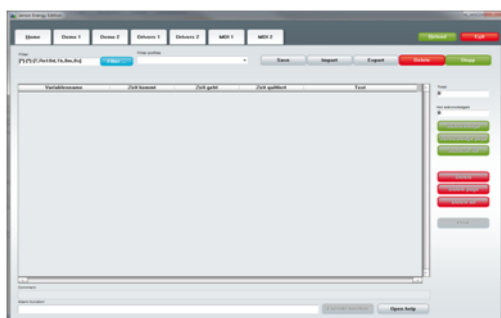
CEL:
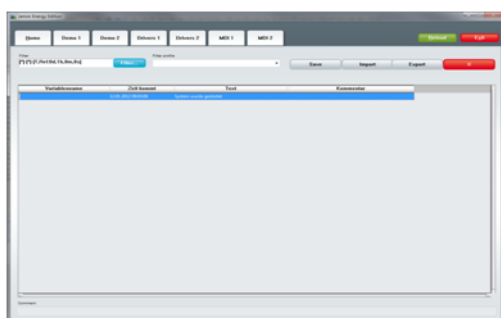
SystemInfo_1 (for each screen resolution 1-3 screens):
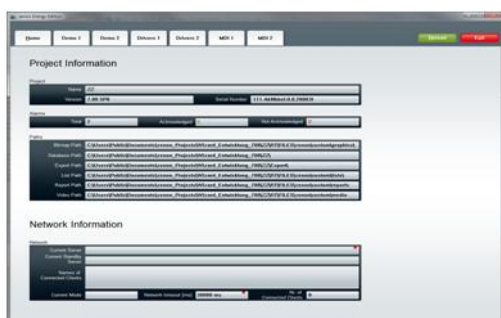
SystemInfo_2 (1280x1024):

Driver:
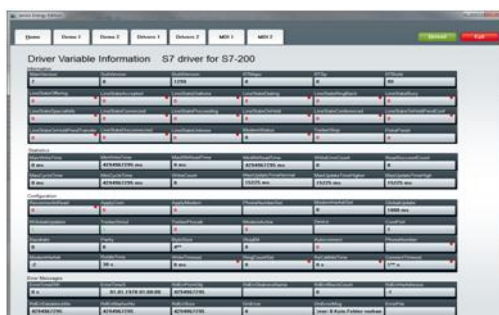
Without demo screens

Start page:
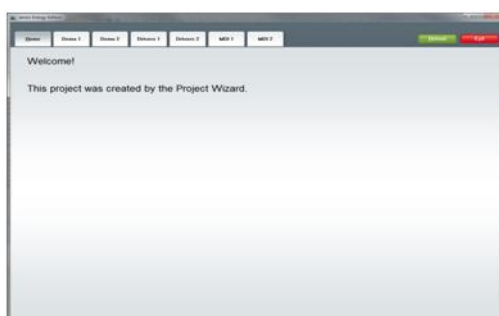
Navigation:

Documentation wizard

This wizard leads you through the steps needed to create a HTML document of the active project.

It can be defined, which modules should be included in the documentation and which not.

The wizard can be edited.

2.1.6   Variables

Wizards for variables.

Variable creation wizard

This wizard serves to create many variables quickly.

> ⚠ **Attention**
>
> *This wizard does not support distributed engineering and is not available in multiuser projects.*

Driver Simulation

The wizard creates an own straton program for each driver in the zenon project for which a driver simulation project is created. This straton program simulates for all variable pairs of the driver, for which a substitution rule applies, a direct allocation from command variable to response variable.

EXAMPLE

- ▸ zenon variables:
  - Test_CO : USINT
  - Test_RV : USINT
  - Switch_CO: USINT
  - Switch_RV: LREAL
- ▸ Rules for substitutions:
  - `*_CO -> *_RV`
- ▸ Results in straton:
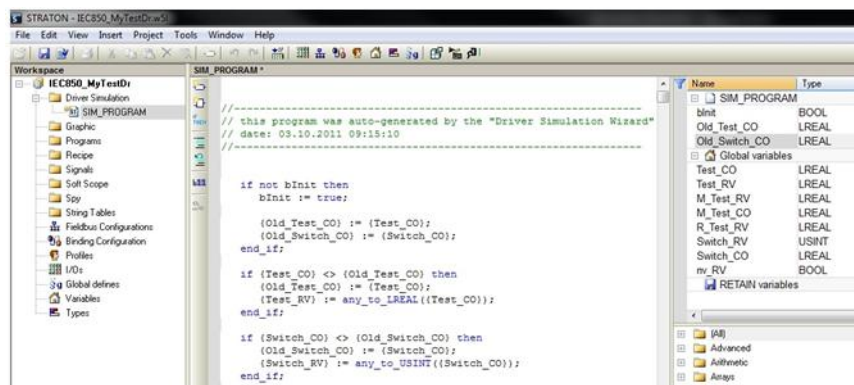
Start wizard

To start the wizard:

1. Click on *File-> Wizards...*

   or press the short cut `Alt+F12`

2. The selection window with the available wizards opens

3. Select the **Variables** folder

4. select the `Driver Simulation Wizard` there
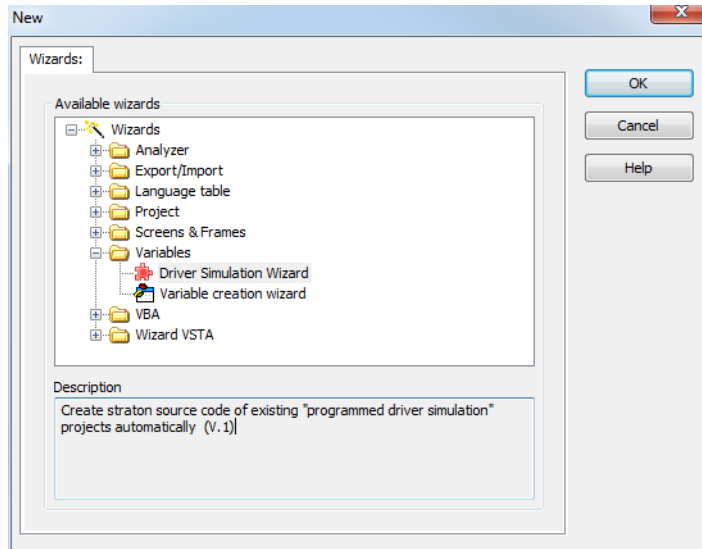
5. click on **OK**

6. The wizard starts with the welcome page

Welcome

Tab `Welcome` informs you about performance and use of the wizard.
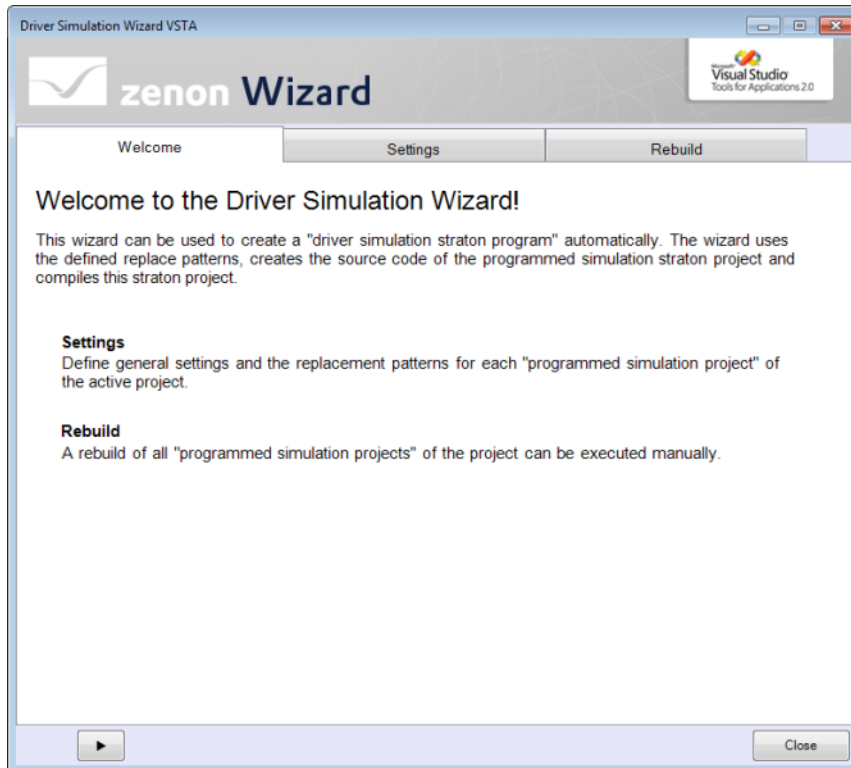
The navigation through the wizard is done by clicking on the individual tabs or step by step by clicking on the arrow keys.

Click on `Close` to close the wizard.

Settings

On this tab the substitution rules are created.
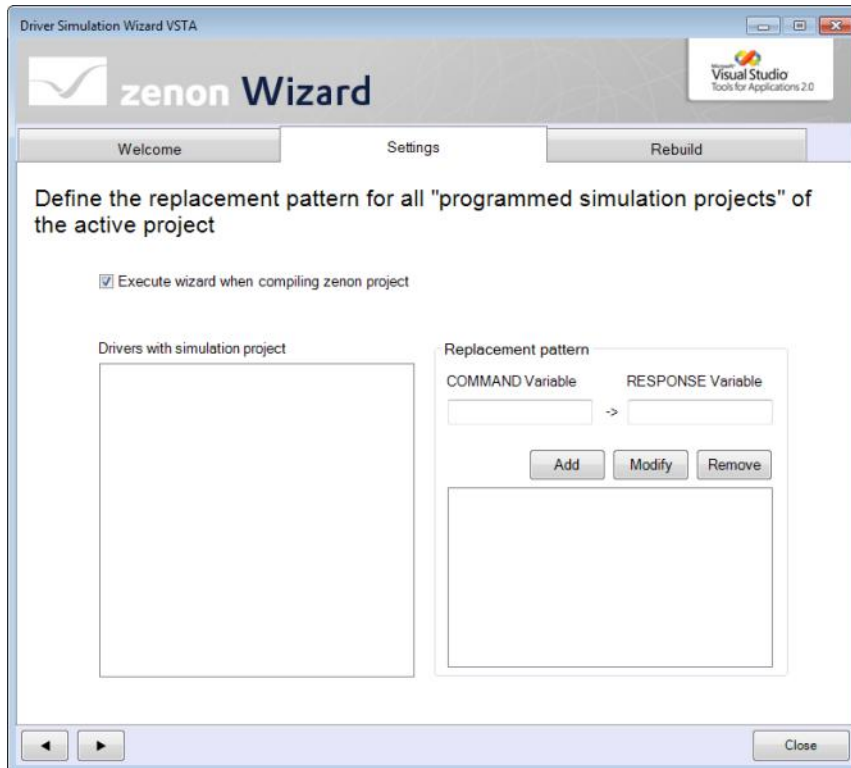
Parameters	Description
`Execute wizard when compiling straton project`	`Active`: As soon as action `Create Runtime files` is executed in zenon, the wizard is also executed. **Note:** The logic for creating the straton project is running in the background. The user interface of the wizard is not displayed.

Drivers with simulation project	Lists all drivers of the zenon project currently active in the zenon Editor for which a driver simulation project was created. If a driver is selected in this list, the defined substitution rules for this driver are displayed in area **Replacement pattern**.
Replacement pattern	Substitution rules.
COMMAND variable	Command variable. Only one wildcard (*) is allowed.
RESPONSE variable	Response variable. Only one wildcard (*) is allowed.
Add	Adds rules to **List of rules**.
Modify	Makes it possible to change selected rules.
Remove	Removes selected rules from the **List of rules**.
List of rules	Lists the defined rules.
Cursor keys	Moves to the previous or next tab.
Close	Closes wizard.

If rules are changed, the recreation of the simulation project is offered when you close the dialog. For this a dialog is opened:
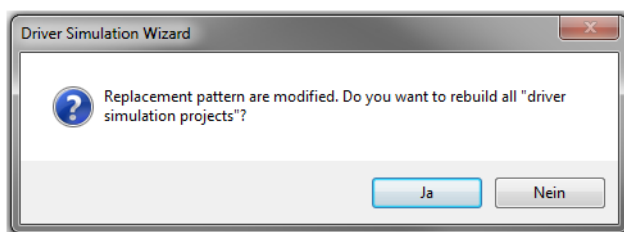
Note: Confirm this dialog with **Yes** if all substitution rules are deleted for a driver. Simulation projects without substitution rules are not considered at the automatic creation of the Runtime files in zenon.

Rebuild

On this tab you can trigger a rebuild of all straton simulation projects.
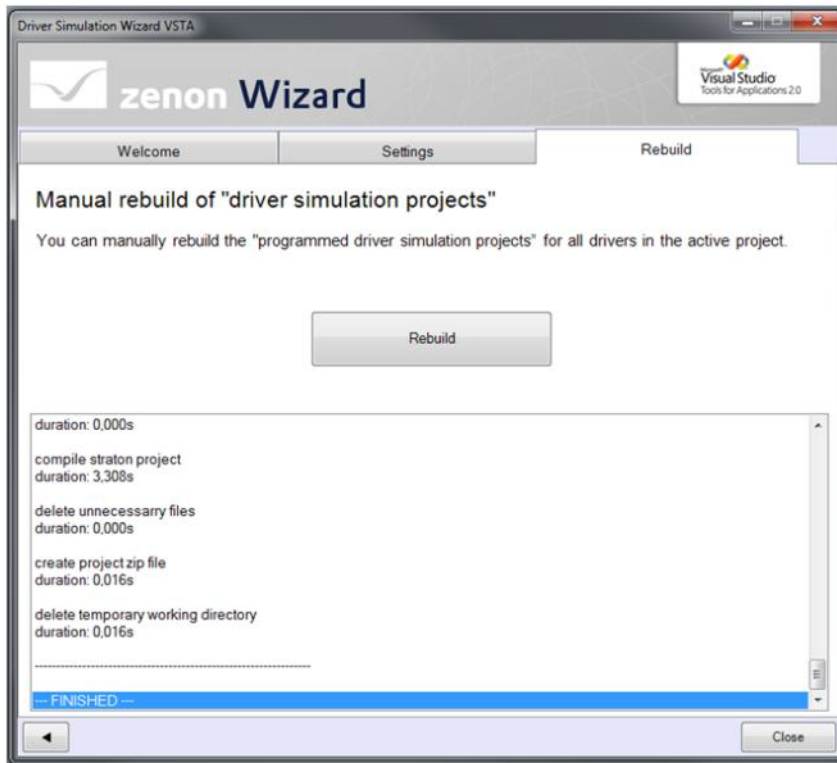
Click on button `Rebuild` in order to start the recreation of the driver simulation project for all corresponding drivers in the project.

2.2 Create and adapt wizards

Wizards are common VBA forms that contain certain functions in the code part. As a result of this, the formulas are recognized as zenon wizard.
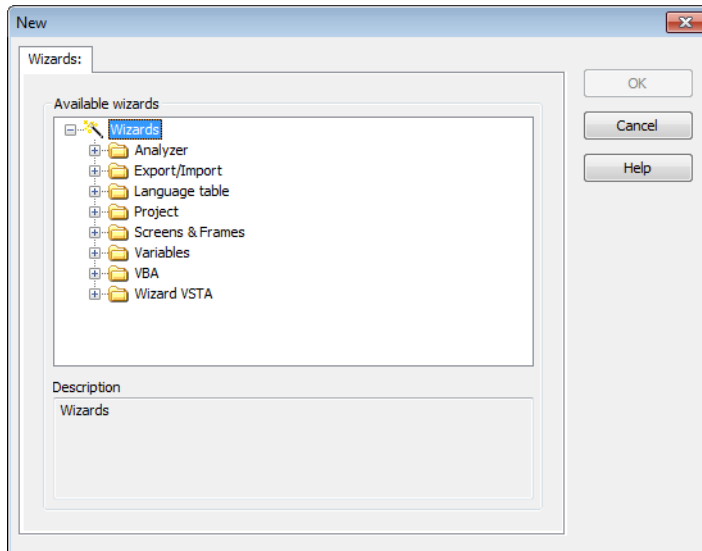
The wizards supplied with zenon can form the basis of your own wizards.

Your own wizards can be stored in your own folders. All required information must be entered into the file named **wizards.ini** (on page 181). This can be included when the wizard is updated (on page 177).

VBA AND VSTA WIZARDS

To create wizards the following information is necessary for VBA and VSTA (on page 173):

- ▶ Name

- ▶ Description

- ▶ Category

- ▶ Display in the dialog

- ▶ Version number

These functions are read by zenon in order to identify forms as wizards and to display corresponding information in the wizard dialog.

For details on creating VSTA wizards see chapter Details VSTA wizards (on page 173).

NAME

States the wizard name as it is displayed in the dialog for the wizard.

Example: **VSTA project wizard**

VBA	VSTA (on page 173)
```Public Function GetWizardName() As String GetWizardName = "Project-Wizard" End Function```	GetWizardName (Type string)

DESCRIPTION

Contains the description of the wizard as it is displayed in area "**Descripton**" of the dialog.

Example: **This wizard creates a simple zenon Project (V.1)**

VBA	VSTA (on page 173)
```Public Function GetWizardInfo() As String GetWizardInfo = "Wizard for creation of a project" End Function```	GetWizardInfo (Type string)

CATEGORY

States the category in which the wizard is sorted.

For example: **Project**

Hint: You can use existing categories. VBA and VSTA wizards can be sorted in the same category.

VBA	VSTA (on page 173)
```Public Function GetWizardCategory() As String GetWizardCategory = "Project" End Function```	GetWizardCategory (Type string)

DISPLAY IN THE DIALOG

Defines whether the wizards is displayed in the dialog. With this you can hide a wizard without deleting all functions or removing them from the add-in:

= `true`: is displayed

= `false`: is hidden

VBA	VSTA (on page 173)
```Public Function IsZenOnWizard() As Boolean IsZenOnWizard = True End Function```	IsZenOnWizard (Type bool)

VERSION NUMBER

States the version number which is displayed at the description. The version number is used to manage the update of the wizards.

Example: **(v.1)**

Note: Within a wizard class there must not be different functions with the same name. This is also true when they report back different parameters.

VBA	VSTA (on page 173)
```Public Function GetWizardVersion() As Integer GetWizardVersion = 6 End Function```	GetWizardVersion (Type int)

2.2.1 Details VSTA Wizard

VSTA WIZARD

```
#region Wizard_Identification
/// <summary>
/// This Static method returns the name of the wizard,
/// which will be displayed in the wizard-tree.
/// </summary>
/// <returns></returns>
```

```
static public string GetWizardName()
{
    string strValue = "Name of the wizard";
    return strValue;
}

/// <summary>
/// This Static method returns the description of the wizard,
/// which will be displayed at the bottom of the wizard-dialog.
/// </summary>/// <returns></returns>
static public string GetWizardInfo()
{
    string strValue = "A more detailed description of the wizard.";
    return strValue;
}

/// <summary>
/// This static method returns the category name of the wizard,
/// which will be used as node-name in the wizards-tree.
/// </summary>
/// <returns></returns>

static public string GetWizardCategory()
{
    string strValue = "Wizard category";
    return strValue;
}

/// <summary>
/// This static method returns a bool which can be used to "switch" the wizard
/// on/off in the wizard dialog (false=wizard is not shown in the tree).
/// </summary>
/// <returns></returns>

static public bool IsZenOnWizard()
{
    bool bValue=true;
}   return bValue;

/// <summary>
/// This static method returns the version of the wizard.
/// Indicated at the bottom of the wizard-dialog.
/// </summary>
/// <returns>wizard version</returns>

static public int GetWizardVersion()
{
```

```
      int nValue = 1;
      return nValue;
}
```

```
/// <summary>
/// This method is called when the wizard has been selected in the
/// wizard dialog and confirmed with "OK".
/// </summary>
```

```
public void StartWizard()
{
      this.Show();
}
#endregion
```

VB.NET

```
'This shared method returns the name of the wizard,
'which will be displayed in the wizard-tree.

Public Shared Function GetWizardName() As String
     GetWizardName = "Name of the wizard"
End Function
'This shared method returns the description of the wizard,
'which will be displayed at the bottom of the wizard-dialog.

Public Shared Function GetWizardInfo() As String
     GetWizardInfo = "A more detailed description of the wizard."
End Function
'This shared method returns the category name of the wizard,
'which will be used as node-name in the wizards-tree.

Public Shared Function GetWizardCategory() As String
     GetWizardCategory = "Wizard category"
End Function
'This shared method returns a bool which can be used to "switch" the wizard
'on/off in the wizard dialog (false=wizard is not shown in the tree).

Public Shared Function IsZenOnWizard() As Boolean
     IsZenOnWizard = True
End Function
'This shared method returns the version of the wizard.
'Indicated at the bottom of the wizard-dialog.

Public Shared Function GetWizardVersion() As Integer
     GetWizardVersion = 1
End Function
```

```
'This method is called when the wizard has been selected in the
'wizard dialog and confirmed with "OK".

Public Sub StartWizard()
    Me.Show()
End Sub
```

C# WORKSPACE

For the Editor to create an instance of the VSTA class dynamically, you must add an additional function to the "Default" Workspace Code. This code segment must exist so that the wizard is displayed after selection in the dialog. This code segment should not be modified!

```
#region Wizard
/// <summary>
/// This Routine Enables the Dynamic creation of VSTA-Wizards.
/// </summary>
/// <param name="strClassname"></param>

public void StartWizard(string strClassname)
{

    //Retrieve the ClassType by its Typename:

    Type t = Type.GetType(strClassname);
    if(t!=null)
    {
        //Since the ClassType has been found, let's create it.
        //The wizard from ClassType %strClassname% is required
        //to have a Constructor with ZenWorkspace Parameter!

        object[] Params = new object[] { this.ZenWorkspace };
        object Wizard = Activator.CreateInstance(t, Params);
        if (Wizard != null)
        {
            t.InvokeMember("StartWizard", BindingFlags.Public | BindingFlags.Instance
| BindingFlags.InvokeMethod, null, Wizard, null);
        }
    }
}
#endregion
```

And for the VB.Net workspace like this:

```
'This Routine Enables the Dynamic creation of VSTA-Wizards,
'and should not be modified or removed!

Public Sub StartWizard(ByVal strClassname As String)
Dim obClassType As Type
Dim obWizard As Object
Dim obParams(0) As Object
```

```
    obClassType = Type.GetType(strClassname)
    If (Not obClassType Is Nothing) Then
        obParams(0) = Me.ZenWorkspace
        obWizard = Activator.CreateInstance(obClassType, obParams)
        If (Not obWizard Is Nothing) Then
            obClassType.InvokeMember("StartWizard",
System.Reflection.BindingFlags.Public Or System.Reflection.BindingFlags.Instance Or
ystem.Reflection.BindingFlags.InvokeMethod, obWizard, Nothing, Nothing)
        End If
    End If
End Sub
```

2.3   Update wizards

When installing zenon service packs, a wizard update is automatically offered.

Existing wizards are not overwritten by the setup. It can therefore be necessary for the wizards to be manually imported and updated.

REQUEST UPDATE

An update can also be initiated manually at any time by selecting `Update wizards` in the `File` drop-down menu.
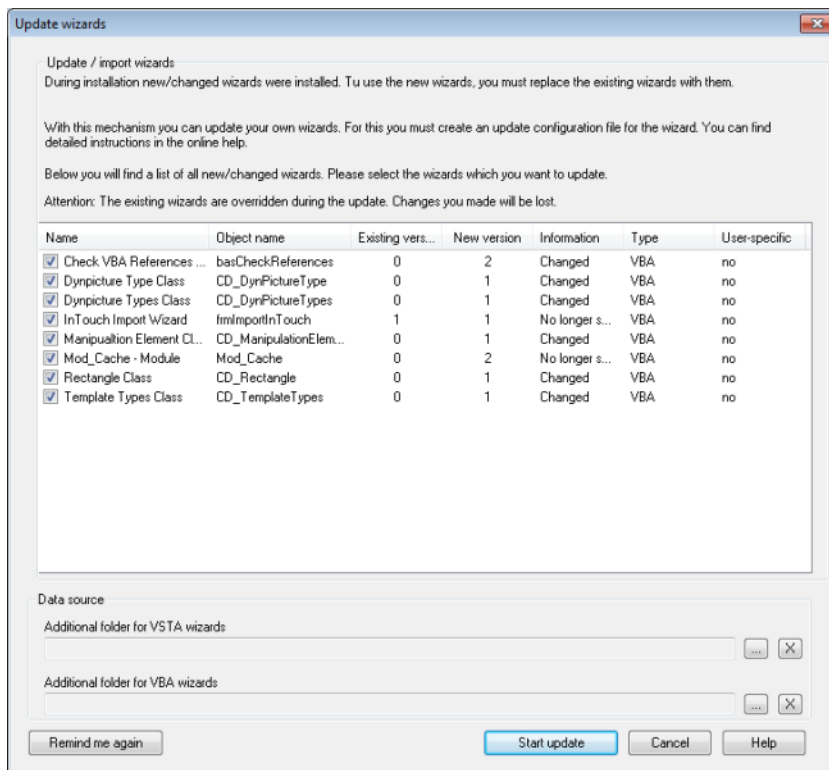
Parameters	Description
Wizard list	Lists all VBA wizards and VSTA wizards present in the add-in that is running.
▸ **Name**	Name of the wizard.
▸ **Object name**	VBA/VSTA object name.
▸ **Previous Version**	Version currently being used.
▸ **New version**	Version that it is being updated to.
▸ **Information**	Status information and information on the pending action: ▸ New: Wizard does not exist in the VBA file. ▸ Changed: A new version is available.

	No longer supported: The existing wizard is obsolete and will be deleted.
▸ **Type**	VBA or VSTA
▸ **User-specific**	wizard was created or changed by the user
Data sources	Possibility of defining your own folder for your own wizards. The save location of the individual **wizards.ini** (on page 181) is entered. The entries for this are saved in zenon.ini in the **[VSTA]** section or **[VBA]** as $WIZARDPATH=$.
Additional folder for VSTA wizards	Individual save location for your own VSTA wizards.
Additional folder for VBA wizards	Individual save location for your own VBA wizards.
Remember me again	The dialog will open again when the Editor is next started.
Start update	The wizards selected in the wizard list are updated.
Cancel	The dialog is ended without updating and is only offered again after the next installation of a service pack.

MAKE SURE THAT YOU ARE UP TO DATE

As objects which are not instanced at the time the dialog is opened are not checked, some objects are always offered for update. This makes sure that you do not work with out-of-date versions. The versions displayed in the update dialog is only used as information for the Consulting and Development departments.

DISTRIBUTE WIZARDS THROUGHOUT THE COMPANY

If you have written your own wizards and would also like to make these available to other users, then you can also use this method.
To do this, you export the wizard from your VBA/VBA development environment and ideally place the export files in an approved network drive. ini files serve to control the imports. These must be created accordingly and also stored in the network.
You can find a description of the files here: For VBA (on page 183) and for VSTA (on page 181).

Now you only need to show your colleagues the location where it is saved and the wizards can easily import these into your Editor. If you have changed or new wizards, you only need to export the new

status, store it and increase the version number. You can therefore easily distribute wizards in your company.

POSSIBLE ERRORS WHEN UPDATING VSTA WIZARDS

Errors	Possible causes
No VSTA wizards are displayed in the update dialog	▸ Only wizards that are in the add-in at the time are listed. If VSTA wizards are not shown, the VSTA add-in must be activated with the Start Editor. To do this, in zenon6.ini, in the **[VSTA]** section, set $LOADED=$ to 1. ▸ If the workspace cannot be compiled due to errors in the code, no wizards are listed. You can read how the original wizards are recovered in the **Recover original wizards** section.
A particular wizard is not displayed.	▸ The wizard does not support the required methods. ▸ The workspace was not yet compiled after the wizard was implemented; the add-in that is running does not contain the wizard.
Self-created wizards are not displayed.	▸ The configured path is incorrect. ▸ The **wizards.ini** in the path configured is obsolete or defective.

RECOVERING ORIGINAL WIZARDS

If wizards are not displayed in the list of the wizards, you can recover the original wizards. To do this:

1. End the zenon Editor

2. Navigate to the folder: `%ProgramData%\COPA-DATA\zenonxxx\VSTAWorkspace` (**xxx** stands for the zenon version)

3. Rename the complete folder

4. Restart the Editor

5. The folder and the wizards are recreated

2.3.1   Creation of the wizards.ini

Creation of the INI file for administering the wizard in VSTA and VBA.

> 💡 **Info**
>
> *This documentation is only available in English.*

VSTA wizards.ini

```
[DEFAULT]: Contains global settings
COUNT: Amount of wizards included in the INI (must be modified when adding/removing a
wizard to the ini
[MYWORKSPACE] Contains settings for the Workspace.cs
VERSION: Current version
[WIZARD_X]: Contains settings of a wizard:
NAME: Name as indicated in the update dialog
CLASSNAME: Name of the form class representing the wizard.
VERSION: Version number
PATH:path-expansion to location of the files.
DELETE: 1 when the wizard is to be removed from the workspace
FILES: The amount of files included in this wizard
FILE_X: The name of a file included in the wizard
TYPE_X: The type of the file (required for the Form.cs and Resx file)
DEP_X: The name of a file on which this file depends
```

EXAMPLE

```
[DEFAULT]
COUNT=3


[MYWORKSPACE]
VERSION=1


[WIZARD_1]
NAME=Import-Wizard
CLASSNAME=Wizard_Exportxml
VERSION=3
```

```
PATH=\Wizard_Exportxml
DELETE=0
FILES=3
FILE_1=Wizard_Exportxml.cs
TYPE_1=Form
FILE_2=Wizard_Exportxml.Designer.cs
DEP_2=Wizard_Exportxml.cs
FILE_3=Wizard_Exportxml.resx
DEP_3=Wizard_Exportxml.cs
TYPE_3=EmbeddedResource


[WIZARD_2]
NAME=Wizard_Project
CLASSNAME=Wizard_Project
VERSION=1
PATH=\Wizard_Project
DELETE=0
FILES=3
FILE_1=Wizard_Project.cs
TYPE_1=Form
FILE_2=Wizard_Project.Designer.cs
DEP_2=Wizard_Project.cs
FILE_3=Wizard_Project.resx
DEP_3=Wizard_Project.cs
TYPE_3=EmbeddedResource


[WIZARD_3]
NAME=Demo Wizard
CLASSNAME=Wizard_Demo
VERSION=1
PATH=\Wizard_Demo
DELETE=0
FILES=3
FILE_1=Wizard_Demo.cs
TYPE_1=Form
FILE_2=Wizard_Demo.Designer.cs
DEP_2=Wizard_Demo.cs
FILE_3=Wizard_Demo.resx
DEP_3=Wizard_Demo.cs
```

```
TYPE_3=EmbeddedResource
```

VBA wizards.ini

```
[DEFAULT]: Contains global settings
COUNT: Amount of wizards included in the INI (must be modified when adding/removing a
wizard to the ini
[MYWORKSPACE] Contains settings for the Workspace.cs
VERSION: Current version
[WIZARD_X]: Contains settings of a wizard:
NAME: Name as indicated in the update dialog
VERSION: Current version
PATH:path-expansion to location of the files.
VB_NAME: Name of the VBA object representing the wizard.
VB_TYPE: 0=form, 1=class
DELETE: 1 when the wizard is to be removed from the workspace
```

EXAMPLE

```
[DEFAULT]
COUNT=3

[MYWORKSPACE]
VERSION=3

[WIZARD_1]
NAME=Wizard for creating variables
VERSION=8
PATH=\CreateVariables\frmCreateVariables.frm
VB_NAME=frmCreateVariables
VB_TYPE=0
DELETE=0

[WIZARD_2]
NAME=Document Wizard
VERSION=12
PATH=\DocuWizard\frmDocuWizardEx.frm
VB_NAME=frmDocuWizardEx
```

```
VB_TYPE=0
DELETE=0


[WIZARD_3]
NAME=Import-Wizard
VERSION=3
PATH=\ImportWizard\frmImportWizard.frm
VB_NAME=frmImportWizard
VB_TYPE=0
DELETE=1
```

Required methods for updating

Example of methods that are required for the wizard to be displayed in the update dialog:

VBA

```
'The following methods define the form as a control system wizard. If IsZenOnWizard is
set to false,
'the wizard does not appear in the Wizard dialog and does not influence the wizard
update dialog.
Public Function GetWizardName() As String
    GetWizardName = "Empty Wizard"
End Function
Public Function GetWizardInfo() As String
    GetWizardInfo = "<TODO: Add description here>"
End Function
Public Function GetWizardCategory() As String
    GetWizardCategory = "<TODO: Add category-information here>"
End Function
Public Function IsZenOnWizard() As Boolean
    IsZenonWizard = False
End Function
Public Function GetWizardVersion() As Integer
    GetWizardVersion = 6
End Function
```

VSTA

```
#region Wizard_Identification
/// <summary>
/// This Static method returns the name of the wizard,
/// which will be displayed in the wizard-tree.
/// </summary>
/// <returns></returns>
static public string GetWizardName()
{
    string strValue = "Demo Wizard";
    return strValue;
}
/// <summary>
/// This Static method returns the description of the wizard,
/// which will be displayed at the bottom of the wizard-dialog.
/// </summary>/// <returns></returns>
static public string GetWizardInfo()
{
    string strValue = "This is our Demo Wizard";
    return strValue;
}
/// <summary>
/// This static method returns the category name of the wizard,
/// which will be used as node-name in the wizards-tree.
/// </summary>
/// <returns></returns>
static public string GetWizardCategory()
{
    string strValue = "Wizard VSTA";
    return strValue;
}
/// <summary>
/// This static method returns a bool which can be used to "switch" the wizard
/// on/off in the wizard dialog (false=wizard is not shown in the tree).
/// </summary>
/// <returns></returns>
static public bool IsZenOnWizard()
{
```

```
    bool bValue = false;
    return bValue;
}
/// <summary>
/// This static method returns the version of the wizard.
/// Indicated at the bottom of the wizard-dialog.
/// </summary>
/// <returns>wizard version</returns>
static public int GetWizardVersion()
{
    int nValue = 1;
    return nValue;
}
/// <summary>
/// This method is called when the wizard has been selected in the
/// wizard dialog and confirmed with "OK".
/// </summary>
public void StartWizard()
{
    this.Show();
}
        #endregion
```