

SERIES: EFFICIENT ENGINEERING WITH ZENON
PART 6

Automated Engineering

What would a topic series entitled “Efficient Engineering”
be without the topic of automated engineering?
Anyone who has been familiar with zenon for a while knows
that this is a standard “function” in our software.
But what does this mean exactly?

TEXT: MARKUS HELBOK,
HEAD OF TECHNOLOGY SERVICES

Automated engineering allows projects to be created partly or fully automatically. Depending on the application, this can happen by means of setting parameters in already existing wizards and tools, such as for the import/export possibilities of zenon Editor. Alternatively, however, zenon is controlled using scripts and macros or an external program, for which programming knowledge is required.

A LONG TRADITION OF AUTOMATED ENGINEERING IN ZENON

Back in 1999, with the release of zenon 5, Visual Basic for Applications (VBA) was introduced in our software. Initially only for Runtime, in order to allow individual customer solutions for our standard product. Our CEO, Thomas Punzenberger, selected VBA because this programming environment, based on Visual Basic (VB) 5, was already sufficiently known from the Microsoft Office packages. The implementation was so successful that

Thomas Punzenberger was even invited by Microsoft to Redmond, to present the solution.

The integration of VBA was also a complete success for our customers and the wish for VBA in the Editor quickly followed. Two fundamental functions for automated engineering were ultimately integrated into zenon 6:

- VBA for the Editor
- XML export/import

The highlights of zenon 6 correspondingly included “automatic engineering” and “efficient reuse”. And this applies now more than ever. Intensive work has been dedicated to these issues in recent years. When VBA was no longer developed and .NET prevailed, we implemented Visual Studio for Applications (VSTA) in zenon – with the possibility to choose development in VB.NET or C#. In addition, we allowed XML export/import for virtually all zenon modules.



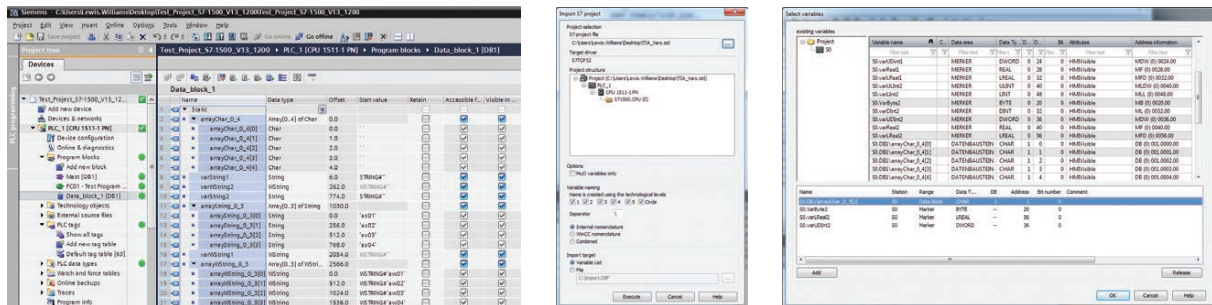


Figure 2: Import of variables from Siemens S7 TIA Portal.

other hand, you can also access the zenon Editor from Excel directly and create, modify or also delete zenon objects.

You have a range of entry points available to start your automation solution in zenon:

- Wizard
- Macro that is started from the macro toolbar
- Macro that is triggered by an Editor event. A very good possibility for starting actions fully automated. E.g. relevant events are triggered when loading a project or when a backup is created. However, it is also possible, when editing a screen for instance, to evaluate a double click on a screen element
- External program that is incorporated in the zenon main menu

AUTOMATE WITH XML

There are now four possibilities for editing variables in the zenon Editor:

1. Import via a driver:
For example, Siemens S7 TIA portal (see Figure 2), Allen-Bradley ControlLogix, OPC UA and many more.
2. Export/import in dBase format:
An old format that can still be used by Excel and Access.
3. Export/import in CSV (comma-separated text file):
This file format is particularly suitable for semi-automated engineering in Excel, where mass processing is often easier to handle than in zenon, for example when modifying addresses.
4. Export/import with XML
The following particularly speak in favor of this supreme method:
 - a. XML export/import is available for virtually all zenon modules.

- b. All data is exported (e.g. dBase and CSV only support four limit values).
- c. All files can be easily used with modern programming methods such as LINQ to XML.
- d. Very good performance.

Methods 2 to 4 (see Figure 3) are particularly well-suited to automation purposes, because they can be used to create, modify and delete variables (and also other objects with XML).

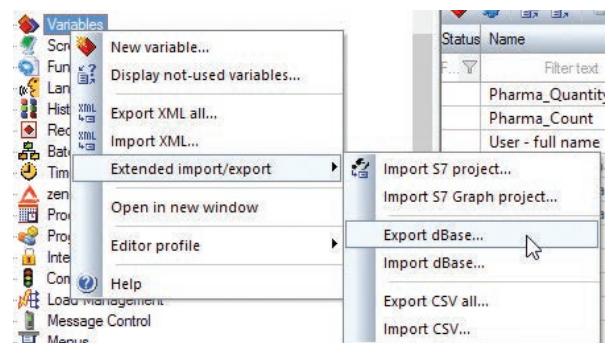


Figure 3:
Export/import in dBase format, as CSV or as XML.

In addition to manual XML export/import, there is the particularly useful method of automating this process using the API. We even expressly advise doing this, because this way objects can be created or modified using the API directly. And with XML export/import, this usually works much more quickly because each object does not need to be handled individually.

We recommend the following procedure: export all existing objects by means of XML. Then carry out the necessary changes in the automation tool (for example, by LINQ to XML) and import the file again. A freely-available example in which the XML import is shown is the project wizard.

Another benefit of the XML method is that you can easily create and save templates with it. Two methods are available:

1. Create the corresponding objects (variables, screens, functions, scripts, recipes, users, etc.) once manually in the zenon Editor and then export these by means of XML into a template folder. This can also be automated if required. What is important for this is a uniform naming convention, for example when all templates start with the "tpl_" prefix. This way, an automation tool can identify all templates and process them automatically.
2. With the screens, it is also possible to create templates using the screen template mechanism. The advantage in doing this is that the name, description, screen size and preview screen are included in the zip file.

IDEAS FOR PRACTICAL USE

The most frequent application for automated engineering is surely use in conjunction with variables. Be it just transferring variables from other tools or issuing Modbus addresses, correcting driver configuration information or converting a driver to a different one, including manipulating address information. The large amount of data and the error rate of manual entries speak in favor of automation here.

A very exciting possibility is offered in conjunction with zenon Logic, our integrated PLC system. The user positions graphic objects in a screen, in the form of symbols. In the background, the variables and detailed screens (faceplates) are then created by a wizard and even the PLC code is created.

A further example in conjunction with zenon Logic is the metering point administration: when a metering point is created, the code to calculate the relative values is also created in zenon Logic.

Many machine manufacturers use automation to individualize their projects. Instead of offering all options in a large project where options that are not required are deactivated for each individual machine, the machine project is tailor-made from a modular system with a wizard. The individual configurations are then stored as

templates in XML format and are compiled as required. This configuration can also be carried out in part with external tools.

EXAMPLES OF AUTOMATED ENGINEERING IN ZENON

THE ZENON EXCEL MACRO

The zenon Excel macro was developed by COPA-DATA consultants and shows how easy it is to create drivers and variables in zenon from Excel. The example (see *Figure 4*) consists of an Excel table with its own macros and four buttons: the connection to zenon Editor is established with the first button; the second button provides all available projects; the drivers defined in the Excel table are created with the third button and the predefined variables are created in zenon with the fourth button. You can request this Excel file from COPA-DATA anytime.

THE ZENON AUTOMOTIVE GENERATOR (ZAG)

"zag" is a very efficient example of automated engineering. You can find details about it in the article entitled "Generate projects in zenon automatically: zag – the wizard for the automotive industry" on page 45 in this issue of IU.

THE PROJECT WIZARD

The project wizard is supplied with zenon and is automatically opened with each new project. It is programmed in C# and covers the issues described very well with:

- Direct modifications using zenon API: creating a driver for example.
- Import of existing XML files from stored templates: importing a demo project.
- Handling of screen templates (see *Figure 5*): zenon standard and user-defined screen template selection and import.

The wizard also shows how multithreading can be implemented. For example, this is how the zenon screen templates are read in their own thread. The wizard is VSTA-based and is available as source code.

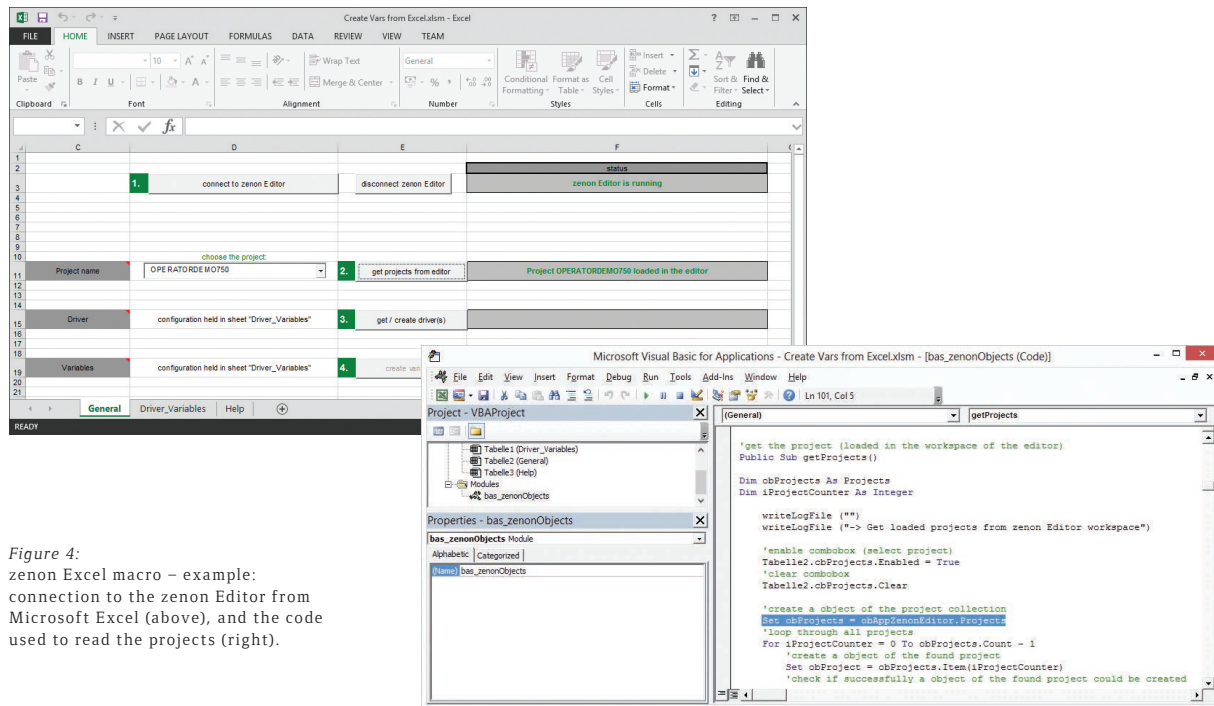


Figure 4:
zenon Excel macro – example:
connection to the zenon Editor from
Microsoft Excel (above), and the code
used to read the projects (right).

CONCLUSION

With automated engineering in zenon, you have a powerful tool at hand; compared to manual engineering you can save a great deal of time in project configuration and provide a significant increase in accuracy and reproducible results. zenon does indeed offer many possibilities out of the box for efficient project creation. However, with automated engineering, these can be multiplied and efficiency can be increased accordingly.

We wish you much success with your automation projects!

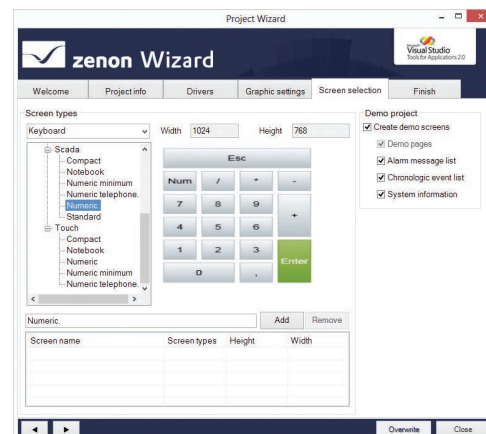


Figure 5: The screen selection for screen types in the zenon project wizard.