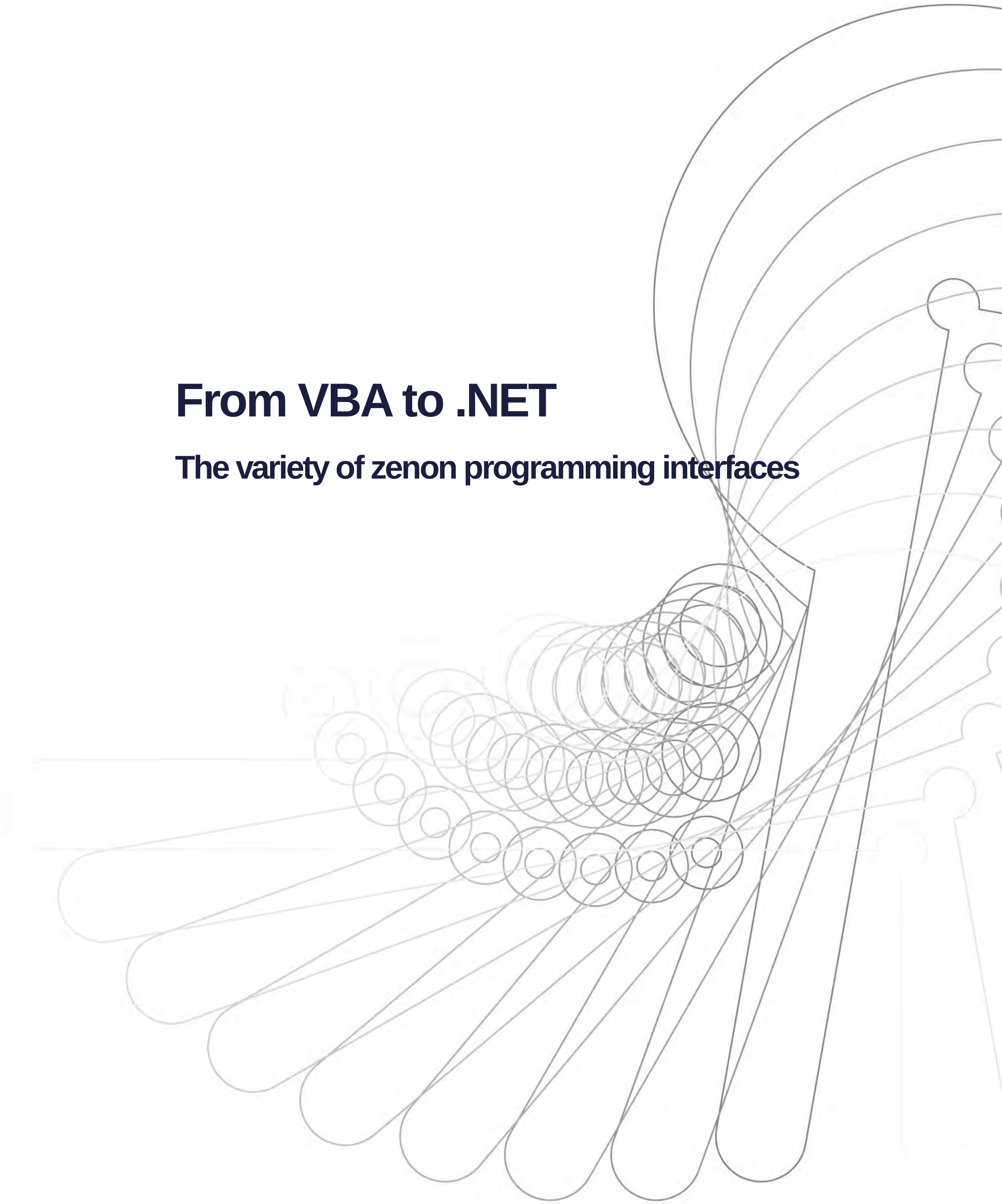
From VBA to .NET

The variety of zenon programming interfaces



Content

From VBA to .NET 1
The zenon Programming Interface.. 3
From expansion to automatic engineering 4
VBA versus VSTA.. 6
.NET provides: 7
Summary.. 9
More on the topic of VBA and VSTA... 10

From VBA to .NET
The variety of zenon programming interfaces

There are many ways to achieve the good design of tailored automation solutions, tailoring them to customers' wishes and adapting them to the most varied of requirements. The possibilities range from direct programming in the PLC using scripts, through to external tools. COPA-DATA has long supported as many ways as possible for customization and follows the philosophy that automators should go their own ways and use their own preferred methods. The variety of interfaces is accordingly large.

VBA has, therefore, already been included as a programming interface since 1999. zenon's process control engine (PCE) offers users the option of programming cyclical processes in VB script or JavaScript. It is particularly suitable for long-lasting functions that run in the background, such as export functions, and it also supports multithreading - in contrast to VBA. Plus, with the integrated IEC 61131-3 SCADA logic, straton, you can also directly access the process and control it from zenon. Now, since version 6.50, zenon also facilitates programming in .NET.

The zenon Programming Interface

The zenon programming interface opens the way to external tools, including .NET. COPA-DATA is not the only manufacturer of automation software with a .NET interface. However, COPA-DATA is one of the first manufacturers at all to have completely integrated VSTA (Visual Studio Tools for Applications) into its product. zenon users therefore benefit from great advantages in terms of reliability and convenience.

VSTA is based on Visual Studio, the professional programming environment from Microsoft, and provides the whole .NET framework in an integrated, professional environment. This comprehensive integration makes it possible to not only allow zenon Runtime systems to be accessed by external .NET applications but for these to be included in the user interface of the automation solution. It also makes it possible to unite two apparently contradictory philosophies: development by the setting of parameters and development by programming. Using zenon, programming is not necessary to create automation projects; they can be created quickly and simply and with a great deal of flexibility just by setting parameters, rendering the projects resistant to errors. At the same time, VSTA permits expansion or redesign of zenon as desired – and thus creates all-round flexibility for developing special solutions.

It is easy to expand existing software with .NET applications. In principle, this can be achieved with any software that is open for external access. However, the important question is: how easily and safely can the programmed expansion be integrated into a process control system or an HMI?

With its powerful programming interface, zenon makes it easy for developers to master the problem areas that always arise when expansions are implemented:

- How can I couple the expansion to individual process variables and, in doing so, ensure that the expansion and control system do not overwrite each other?
- How to react to an alarm in the control system?
- How are user rights coordinated in the HMI and in the expansion?

The zenon programming interface also provides .NET and VBA with zenon-objects and functions and thus enables direct access to zenon functionality. The VSTA integration provides people working on the project with a further essential benefit: automatic engineering.

From expansion to automatic engineering

VBA and VSTA are both well suited to adding special functions or interfaces to a program. zenon also works well in combination with external programs that are written in other languages. In this way, standard features can be expanded with extras, for instance, a special input mask, or very individual user instructions.

COPA-DATA generally aims to include functions that customers desire in regular versions, with the overriding principle being that it is better to set parameters than to have to carry out laborious programming. This is because functional expansions implemented via external programs do not always automatically feature zenon principles such as complete network functionality, integrated redundancy or comprehensive user administration. In addition, they often cause higher maintenance costs, are more liable to errors and sometimes reduce performance.

However, scripting can sometimes be very desirable and, in certain instances, the best way to quick and safe automation. The advantages of scripting are primarily evident in the area of automatic engineering. The comprehensive zenon programming interface has been available in the Editor since version 6 (2003) and has been continually expanded and adapted to deliver new capabilities.

Both Runtime and Editor functionalities are available. The most important Editor functions are the creation of options, process screens and screen elements, templates, alarms, administration of external files, creating variables and data types, and user management, with its stringent access control.

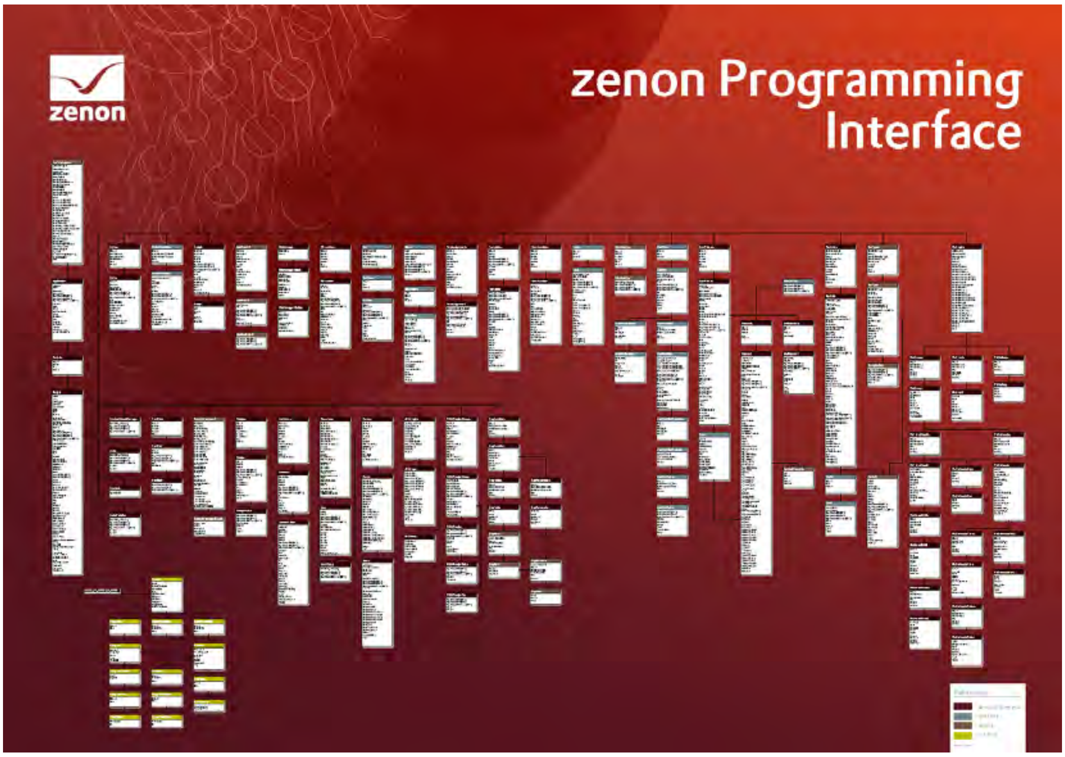


Figure 1: zenon Programming Interface: Overview

zenon Programming Interface as PDF

Customers report savings of up to 98% in engineering through the automatic creation of projects. Engineering becomes quicker using Automatic Engineering, is guaranteed to be free of errors, and makes the automation engineer's life noticeably easier. This is, in part, because standards on the use of wizards can be defined and implemented very easily.

For a long time, VBA was the standard programming language available to those wishing to expand zenon –until .NET was integrated so comprehensively in version 6.50. However, VSTA has not superseded VBA. Both development environments exist alongside each other in zenon. It is left for the user to decide which tool is better suited to their intended purpose.

VBA versus VSTA

VBA can be learnt very easily and is well documented. Its large community of fans ensures that there is considerable literature and quick help for any questions or problems. In particular, VBA is well suited to smaller projects.

Nevertheless, VBA is subject to some restrictions that can become an obstacle, especially with larger projects. A fundamental issue is its deficiencies in the implementation of object-orientation. There is no implementation inheritance at all. Instead of quickly reusing error-free code , more time and money must be invested than is desirable. There is no special syntax nor code highlighting for procedures in the VBA development environment. This deficiency is rectified with the use of the Visual Studio development environment in VSTA.

Security

VBA is single-threaded software. Therefore processor-intensive applications do slow the system down and the multi-core processors now favored cannot contribute to increasing speed. Anyone who wishes for user interfaces and calculations to run in separate threads in order to increase user-friendliness and to optimize performance, will not find any support for this when using VBA.

And, if a VBA program becomes unstable, this affects the host application at the same time. This can quickly become a problem in sensitive automation areas. In addition, the programming already requires increased effort, because there is no global method of handling errors. The Error Handler must be added in each procedure.

Some security conscious developers also see problems with VBA. VBA generally permits access to all COM functionalities. Therefore code can be executed that is not directly related to the host application.

	VBA	VSTA
Easiness to learn	Very simple	Simple
Help available on the internet (documentation, example programs, forums, etc.)	Various	VSTA: Being created C#: A large amount
zenon Forum	Exists	Exists
Object-orientation	Limited	Complete
Runtime access	Full	Full, with the exception of some special zenon events
Editor access	Full	Full
Access from web client	Possible	Not possible
Development environment	Completely integrated into zenon	
Deployment	Completely integrated into zenon	
Reload	Completely integrated into zenon	
Backup	Completely integrated into zenon	
Multithreading	No	Yes
Security	Limited	Good

.NET provides:

For many companies and programmers, .NET has already become accepted as the more future-proof variant in terms of developing expansions. The number of programmers that are well trained in .NET is continually increasing. They find a professional and versatile environment in zenon and its integrated VSTA.

The .NET framework is to become the primary software platform for the development of Windows software and will supersede classic Win32 programming. However, it also requires basic knowledge of object-orientated programming; .NET and C#/Visual Basic.NET.

.NET has considerable advantages in comparison to COM/COM+:

- Object-orientation
- .NET Framework classes
- Platform independence
- Memory administration
- Easy code transfer through simple copying, or installation using wizards
- Configuration by XML instead of registry entries
- Windows Presentation Foundation (VSTA supports the inclusion of WPF controls)
- Threading

Security

The security aspects are shifted completely to the .NET framework with VSTA. VSTA applications run separately from the host application in a separate thread. If the application crashes, the .NET framework offers more options to leave the host application unaffected so it can continue to run stably.

VSTA generally provides considerably more stability than VBA, from conception to execution of the code. Erroneous code is reliably caught by the consistent implementation of exception handling and is therefore unlikely to cause external applications to crash.

Summary

For simple scripts, there is no great performance difference between VSTA and VBA. However, for complex expansions VSTA is generally faster; for example, through quicker access to databases using ADO.NET or better performance by shifting program functions to a background thread.

VBA is easy to learn and is very well suited to small projects. It already has a large community of fans and, therefore, numerous example programs.

VSTA primarily excels through better resource management. Projects that require many objects and a large amount of working memory benefit considerably from VSTA. VSTA is the better choice for new host programs, because it features object-orientation, higher security and a more extensive development environment.

With relation to zenon, and in contrast to external solutions, the integrated VSTA scores points by being easy to use and having no acquisition costs. It is already included in zenon and therefore ready to use on any computer. Plus, it uses all functionality integrated in zenon for backup, deployment and reloading. This means that each time the project is saved, the VSTA project is also saved, VSTA programs are transferred via zenon remote transport and, if a zenon project is reloaded online, the VSTA project is automatically reloaded.

VSTA’s limited compatibility with VBA is no problem for zenon users. They have a free choice anyway. VBA remains a component of zenon and existing macros do not need to be ported to .NET. Therefore, existing projects can continue to use VBA and new projects can be equipped with all VSTA advantages. Both benefit from their complete integration into zenon in comparison to external applications that access zenon.

More on the topic of VBA and VSTA

You can find numerous articles on VBA and VSTA in the COPA-DATA forum and the COPA-DATA magazine, Information Unlimited, examples of which are listed below.

Martin Stark in Information Unlimited 18:

VSTA (.NET) in zenon 6.50 – (R)evolution? (Page 55)

Günther Haslauer in Information Unlimited 17 and 18:

.NET Windows Form Controls in zenon [Part 1] (Page 26)

.NET Windows Form Controls in zenon [Part 2] (Page 56)

VBA series from Robert Ficker in Information Unlimited 13 to 18:

Setting standards for engineering (Standardization) (Page 34)

Reusing existing project parts (XML Importing) (Page 59)

Automatic Engineering with Excel (Creating variables) (Page 46)

Project wizard with file import from database or file (Page 56)

Picture-Filter: A new interface in the zenon object model (Page 59)

Forum on the COPA-DATA website

VSTA forum on the website

VBA forum on the website

COPADATA

do it your way