

# Von VBA zu .NET

Die zenon Vielfalt an Programmierschnittstellen

## **Inhalt**

<b>Von VBA zu .NET .....</b>	<b>1</b>
<b>zenon Programming Interface .....</b>	<b>3</b>
<b>Vom Erweitern zum automatischen Engineering.....</b>	<b>4</b>
<b>VBA versus VSTA.....</b>	<b>6</b>
<b>Das bringt .NET: .....</b>	<b>7</b>
<b>Resümee .....</b>	<b>9</b>
<b>Mehr zum Thema VBA und VSTA .....</b>	<b>10</b>

## **Von VBA zu .NET**

### **Die zenon Vielfalt an Programmierschnittstellen**

Automatisierungen anpassen, auf Kundenwünsche zuschneiden, an unterschiedlichste Bedingungen adaptieren – es gibt viele Wege, dies zu erreichen. Von direkter Programmierung in der SPS über Skripte bis zu externen Tools reichen die Möglichkeiten. COPA-DATA unterstützt mit zenon schon lange möglichst viele Wege der Individualisierung und folgt der Philosophie, dass Automatisierer ihren eigenen Wegen und Vorlieben folgen sollen. Entsprechend groß ist die Vielfalt der Schnittstellen:

So ist VBA bereits seit 1999 als Programmierschnittstelle in zenon enthalten. Die Process Control Engine (PCE) bietet Anwendern die Möglichkeit, zyklische Abläufe in VB Script oder Java Script zu programmieren. Sie eignet sich besonders für lang andauernde Funktionen, die im Hintergrund ablaufen, wie etwa Export-Funktionen, und unterstützt im Gegensatz zu VBA auch Multithreading. Mit der integrierten IEC 61131-3 SCADA-Logic straton greift man direkt aus zenon steuernd auf den Prozess zu. Und seit der Version 6.50 beherrscht zenon auch .NET.

### **zenon Programming Interface**

Das zenon Programming Interface öffnet den Weg zu externen Tools, auch zu .NET. COPA-DATA ist zwar nicht der erste Hersteller von Automatisierungssoftware mit einem .NET-Interface. Allerdings hat COPA-DATA als einer der ersten Softwarehersteller überhaupt VSTA (Visual Studio Tools for Applications) vollständig in sein Produkt integriert. zenon Anwender profitieren so von starken Vorteilen bei Sicherheit und Komfort.

VSTA basiert auf der professionellen Programmierumgebung von Microsoft, dem Visual Studio, und stellt das gesamte .NET Framework in einer integrierten, professionellen Umgebung zur Verfügung. Diese tiefgreifende Integration ermöglicht nicht nur den Zugriff auf zenon Laufzeitsysteme von externen .NET-Applikationen oder deren direkte Einbindung in die Oberfläche der Automatisierungslösung. Sie schafft es auch, zwei scheinbar gegensätzliche Philosophien zu vereinen: Parametrieren versus programmieren. Mit zenon werden Automatisierungsprojekte fehlerresistent und schnell parametriert; Programmierung ist im Programm selbst nicht nötig. Gleichzeitig erlaubt VSTA den beliebigen Aus- und Umbau von zenon – und schafft so umfassende Flexibilität bei der Entwicklung von Speziallösungen.

Vorhandene Software durch .NET-Applikationen zu erweitern, ist kein Problem. Das gelingt im Prinzip mit jeder Software, die sich für externe Zugriffe öffnet. Die entscheidende Frage ist aber: Wie einfach und sicher lässt sich die programmierte Erweiterung in ein Prozessleitsystem oder HMI integrieren?

zenon macht es mit seinem Automation Interface Programmierern leicht, die bei Erweiterungen immer wieder auftauchenden Problemstellungen elegant zu meistern:

- Wie kann man die Erweiterung an einzelne Prozessvariablen koppeln und dabei sicher stellen, dass sich Erweiterung und Leitsystem nicht gegenseitig überschreiben?
- Wie wird auf einen Alarm im Leitsystem reagiert?
- Wie werden die Benutzerrechte im HMI und in der Erweiterung koordiniert?

Das zenon Programming Interface stellt .NET aber auch VBA zenon Objekte und Funktionen zur Verfügung und ermöglicht so den direkten Zugriff auf zenon Funktionalitäten. Die VSTA-Integration bringt Projektanten noch einen weiteren elementaren Nutzen: automatisches Engineering.

## **Vom Erweitern zum automatischen Engineering**

VBA und VSTA eignen sich beide gut, um spezielle Funktionen oder Oberflächen zu einem Programm hinzuzufügen. zenon arbeitet auch gut mit externen Programmen, die in anderen Sprachen geschrieben werden, zusammen. So lassen sich Standardfunktionalität mit Extras, wie einer speziellen Eingabemaske, oder sehr individueller Benutzerführung erweitern. COPA-DATA baut in der Regel Kundenwünsche auch in reguläre Versionen ein, ganz nach dem Motto: Lieber einfach parametrieren als aufwändig programmieren. Denn funktionelle Erweiterungen über externe Programme verfügen nicht automatisch über zenon Prinzipien wie volle Netzwerkfähigkeit, integrierte Redundanz oder umfassende Benutzerverwaltung. Sie verursachen zudem oft höheren Wartungsaufwand, sind fehleranfälliger und mindern manchmal die Performance.

Manchmal ist Skripting aber sogar sehr erwünscht und der beste Weg zu schneller und sicherer Automatisierung. Seine Vorteile zeigen sich primär im Bereich Editor-Automatisierung, also beim automatischen Engineering. Das umfangreiche zenon Programming Interface steht im Editor seit Version 6 (2003) zur Verfügung und wurde dafür ständig erweitert und an neue Möglichkeiten angepasst.

Sowohl Runtime- als auch Editor-Funktionalitäten stehen zur Auswahl. Zu den wichtigsten Editorfunktionen zählen die Möglichkeiten, Prozessbilder und Bildelemente zu erstellen, Schablonen, Alarme, das Verwalten von externen Dateien, die Erstellung von Variablen und Typen oder das Benutzer-Management mit seiner strengen Zugriffsverwaltung.



Abbildung 1: zenon Programming Interface: Übersicht

[zenon Programming Interface als PDF](#)

Kunden berichten von Einsparungen von bis zu 98% der Engineeringzeit durch automatisches Erstellen von Projekten. Das Engineering wird mit diesen Methoden schneller, garantiert fehlerfrei – und macht das Leben der Ingenieure spürbar einfacher. Auch, weil sich Standards über den Einsatz von Wizards sehr einfach definieren und implementieren lassen.

Lange war VBA die Standard Programmiersprache zur Erweiterung von zenon – bis mit Version 6.50 .NET in großem Umfang integriert wurde. VSTA hat VBA nicht abgelöst. Beide Entwicklungsumgebungen existieren parallel in zenon. Es bleibt dem Anwender überlassen, sich zu entscheiden, welches Tool für seine Absichten besser geeignet ist.

## **VBA versus VSTA**

VBA ist sehr leicht erlernbar und gut dokumentiert. Seine große Fangemeinde sorgt für umfangreiche Literatur und schnelle Hilfe bei offenen Fragen oder Problemen. Vor allem für kleinere Projekte eignet sich VBA sehr gut.

Allerdings unterliegt VBA auch einigen Einschränkungen, die bei größeren Projekten zum Hindernis werden können. Ein grundlegendes Problem ist seine mangelhaft umgesetzte Objektorientierung. Eine Implementierungsvererbung fehlt komplett. Statt Code schnell und fehlerarm wiederzuverwenden, muss mehr Zeit und Geld in die Entwicklung gesteckt werden als erwünscht. So existiert in der VBA-Entwicklungsumgebung weder eine spezielle Syntax noch Code-Highlighting für Prozeduren. Dieses Manko wird mit der Verwendung der Entwicklungsumgebung des Visual Studios in VSTA behoben.

### **Sicherheit**

VBA ist Single-Threaded-Software. Rechenintensive Anwendungen bremsen damit das System aus und die heute üblichen Mehrkernprozessoren können dann auch nicht zu mehr Tempo beitragen. Wer Benutzeroberfläche und Berechnungen in separaten Threads laufen lassen möchte, um die Benutzerfreundlichkeit zu erhöhen und die Performance zu optimieren, findet bei VBA keine Unterstützung.

Und: Wird ein VBA-Programm instabil, reißt es die Hostapplikation gleich mit. Für sensible Automatisierungsbereiche kann das schnell zum Problem werden. Dazu kommt, dass bereits die Programmierung erhöhten Aufwand verlangt, da keine globale Fehlerbehandlung bereit steht. Der Error Handler muss in jede Prozedur extra eingefügt werden.

Probleme mit VBA sehen auch manche Sicherheitsverantwortlichen in Unternehmen. VBA erlaubt generell den Zugriff auf alle COM-Funktionalitäten. Damit kann auch Code ausgeführt werden, der nicht in unmittelbarem Zusammenhang mit der Hostapplikation steht.

	VBA	VSTA
Erlernbarkeit	Sehr einfach	Einfach
Im Internet verfügbare Hilfen (Dokumentation, Programm-beispiele, Foren, etc.)	Vielfältig	VSTA: Im Entstehen C#: Sehr viel
zenon Forum	Vorhanden	Vorhanden
Objektorientierung	Eingeschränkt	Vollständig
Runtime-Zugriff	Voll	Voll, ausgenommen einige spezielle zenon Events
Editor-Zugriff	Voll	Voll
Zugriff vom Webclient	Möglich	Nicht möglich
Entwicklungsumgebung	Voll in zenon integriert	
Deployment	Voll in zenon integriert	
Reload	Voll in zenon integriert	
Sicherung	Voll in zenon integriert	
Multithreading	Nein	Ja
Sicherheit	Eingeschränkt	Gut

## Das bringt .NET:

Für viele Unternehmen und Programmierer ist .NET bereits die zukunftsichere Variante, wenn es um die Entwicklung von Erweiterungen geht. Die Zahl der in .NET gut ausgebildeten Programmierer wächst beständig. In zenon und seinem integrierten VSTA finden sie eine professionelle und vielfältige Umgebung.

Das .NET-Framework soll die primäre Softwareplattform für die Entwicklung von Windows-Software werden und die klassische Win32-Programmierung ablösen. Sie verlangt aber auch grundlegende Kenntnisse zur objektorientierten

Programmierung, zu .NET und zu C#/Visual Basic.NET. Im Vergleich zu COM/COM+ bringt .NET wesentliche Vorteile mit sich:

- Objektorientierung
- .NET Framework-Klassen
- Plattformunabhängigkeit
- Speicherverwaltung
- einfache Weitergabe durch simples Kopieren oder Installation mit Wizards
- Konfiguration per XML statt Registryeinträge
- WPF (VSTA unterstützt das einbinden von WPF-Controls)
- Threading

## **Sicherheit**

Die Sicherheitsaspekte werden bei VSTA komplett auf das .NET-Framework ausgelagert. VSTA-Applikationen laufen getrennt von der Host-Applikation in einem separaten Thread. Sollte die Anwendung abstürzen, bietet das .NET Framework erweiterte Möglichkeiten um die Hostapplikation davon unberührt und stabil weiterlaufen zu lassen.

VSTA bietet insgesamt eine deutlich höhere Stabilität als VBA, von der Konzeption bis zur Ausführung des Codes. Fehlerhafter Code wird durch die konsequente Umsetzung des Exception Handlings zuverlässig abgefangen und kann so keine Abstürze externer Anwendungen hervorrufen.

## Resümee

Für einfache Skripte unterscheiden sich VSTA und VBA in der Performance nicht wesentlich. Bei komplexeren Erweiterungen ist VSTA in der Regel schneller, beispielsweise durch schnelleren Zugriff auf Datenbanken durch ADO.NET oder besserer Performance durch Auslagern von Programm-Funktionen in einen Hintergrund-Thread.

VBA ist leicht erlernbar und eignet sich sehr gut für kleine Projekte. Es verfügt bereits über eine große Fangemeinde und damit unzählige Programmbeispiele.

VSTA punktet vor allem mit besserem Ressourcenmanagement. Projekte, die viele Objekte und eine große Menge Arbeitsspeicher benötigen, profitieren deutlich von VSTA. Für neue Hostprogramme ist VSTA die bessere Wahl, da es mit Objektorientierung, höherer Sicherheit und einer umfangreicheren Entwicklungsumgebung aufwartet.

Gegenüber externen Lösungen punktet das integrierte VSTA durch einfaches Handling und Null Euro Anschaffungskosten. Es ist bereits in zenon enthalten und damit auf jedem zenon Rechner einsatzbereit. Vor allem aber nutzt es alle in zenon integrierten Funktionalitäten für Sicherung, Deployment und Reload. Das heißt: Mit jeder Projektsicherung wird auch das VSTA-Projekt gesichert, mit dem zenon Remote Transport werden auch VSTA-Programme übertragen und wenn ein zenon Projekt online nachgeladen wird, wird auch das VSTA-Projekt automatisch nachgeladen.

Die eingeschränkte Kompatibilität von VSTA zu VBA ist für zenon Anwender kein Problem. Sie haben ohnehin freie Wahl. VBA bleibt weiterhin Bestandteil von zenon und bestehende Makros müssen nicht in .NET portiert werden. So können bestehende Projekte weiter auf VBA setzen und neue Projekte mit allen VSTA-Vorteilen ausgestattet werden. Beide profitieren von ihrer Integration in zenon gegenüber externen Anwendungen, die auf zenon zugreifen.

## Mehr zum Thema VBA und VSTA

Im COPA-DATA Forum und dem COPA-DATA Magazin Information Unlimited finden Sie umfangreiche Artikel zu VBA und VSTA. Zum Beispiel:

Martin Stark in Information Unlimited 18:

[VSTA \(.NET\) in zenon 6.50 – die \(R\)evolution?](#) (Seite 55)

Günther Haslauer in Information Unlimited 17 und 18:

[.NET Windows Form Controls in zenon \[Teil 1\]](#) (Seite 26)

[.NET Windows Form Controls in zenon \[Teil 2\]](#) (Seite 56)

VBA-Serien von Robert Ficker in Information Unlimited 13 bis 18:

[Standards für die Projektierung setzen \(Standardisierung\)](#) (Seite 34)

[Bestehende Projektteile wieder verwenden \(XML Import\)](#) (Seite 59)

[Automatisch projektieren mit Excel \(Variablen anlegen\)](#) (Seite 46)

[Projekt Wizard mit Datenimport aus DB oder Datei](#) (Seite 56)

[Picture Filter: Eine neue Schnittstelle im zenon Objektmodell](#) (Seite 59)

Forum auf der COPA-DATA-Website

[VSTA Forum auf der Website](#)

[VBA Forum auf der Website](#)



© 2010 Ing. Punzenberger COPA-DATA GmbH

All rights reserved.

Distribution and/or reproduction of this document or parts thereof in any form is permitted solely with the written permission of the COPA-DATA company. The technical data contained herein has been provided solely for informational purposes and is not legally binding. Subject to change, technical or otherwise. zenon<sup>®</sup> and straton<sup>®</sup> are both trademarks registered by Ing. Punzenberger COPA-DATA GmbH. All other brands or product names are trademarks or registered trademarks of the respective owner and have not been specifically earmarked.