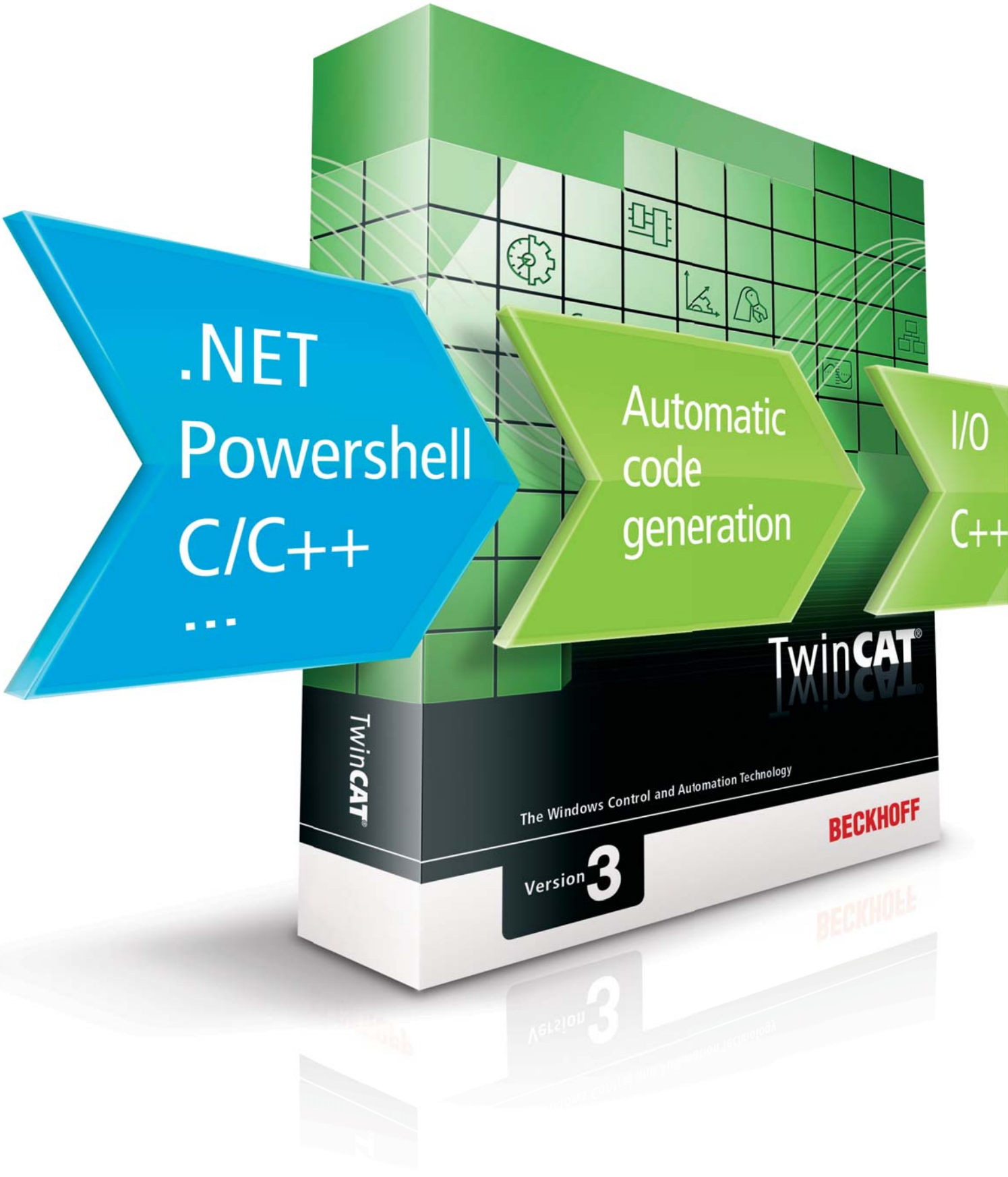

.NET
Powershell
C/C++
...
Automatic
code
generation
I/O
C++
TwinCAT
The Windows Control and Automation Technology
BECKHOFF
Version 3

More efficient engineering with TwinCAT 3

Automation Interface – the open interface for automatic code generation

The Automation Interface within TwinCAT 3 automation software enables complete remote control of the TwinCAT engineering system and automatic generation of programs and configurations. Depending on the degree of automation, manual handling of control projects and any associated errors that go along with it can be reduced or eliminated. The quality of software engineering is increased by automating the project generation, while at the same time saving time and cost.

The automation of machines and plants is becoming increasingly complex and time-consuming. Engineering expenditures are growing in proportion to this, which is reflected, among other things, by the increased costs for the development, configuration and programming a PLC. These costs can be reduced if either the complete software application or parts of it are generated automatically. On top of that, errors in the configuration can be reduced by automated code generation. This is precisely where the TwinCAT Automation Interface makes a dramatic impact: it provides a programming interface using the complete TwinCAT system and virtually all offline and online functions can be remotely-controlled. The error-prone and (in terms of personnel costs) expensive generation of I/O configurations and programs is thus simpler to accomplish.

The TwinCAT Automation Interface

Various interfaces to different tools that simplify the engineering process have already been available for many years in TwinCAT 2. The TwinCAT Automation Interface is one of these disclosed interfaces that is based on the Component Object Model (COM) from Microsoft and – in addition to the standard Microsoft .NET programming languages – also supports modern script languages such as Windows Powershell. This opens up new areas of application and usage scenarios for the TwinCAT Automation Interface.

The TwinCAT Automation Interface is made up of a large number of programming routines that are available to the user in the form of classes and methods. Among other things they enable the loading of TwinCAT projects from a source code database such as the Team Foundation Server. The projects can be modified afterwards and activated on the runtime system. Navigation through a project is intuitive, because the underlying tree structure is the same as in the TwinCAT engineering environment, which facilitates faster familiarization with the Automation Interface.

Configuration nodes can be added, deleted and modified within a project. Appropriate routines are available for this, irrespective of the type of node (PLC,

I/O or Motion). The modification of parameters on a node is simple, since almost all the settings are provided in a generic XML format. The configuration of a node is loaded by means of methods available in the Automation Interface; it can be modified by means of standard XML mechanisms such as those provided, for example, by the .NET framework and saved once again in the TwinCAT configuration. I/O nodes can contain, for example, node addresses or baud rates as parameters, while a PLC node in PLCopen-XML provides the source code (e.g. of a function block) in the corresponding IEC 61131-3 language. Here, too, the source code can be loaded, modified and saved again.

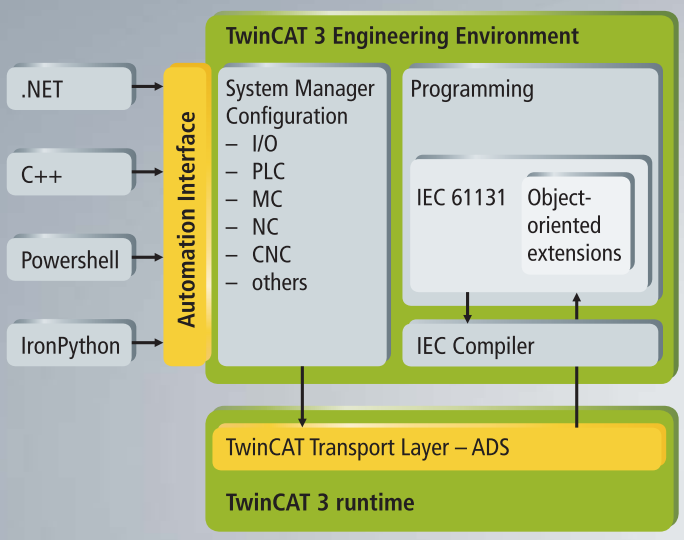
All elements of the TwinCAT configuration are made available to the outside with fine granularity via the Automation Interface. This provides a high degree of flexibility for the customer application, since the Automation Interface – as opposed to the step-by-step generation of the configuration "node for node" – also enables the generation of configurations on the basis of templates. For this, the application could access a template pool, e.g. a source control system, and successively build up the configuration. Templates can be defined in a range from complete projects right down to the node level; i.e. an EtherCAT device from the I/O area, once configured, could be saved in a template file, imported later via the Automation Interface and added to a new configuration.

The generation of a configuration – especially in the case of I/O configurations – is based among other things on hardware addresses, which are of course only available if access to the corresponding I/O device is possible. The Automation Interface provides the option to prepare a configuration offline, i.e. without connected I/O devices, and to add the parameters later on when the I/Os are present. Alternatively, a configuration can also be accomplished with connected I/O devices using the "Scan function" that is familiar from TwinCAT XAE.

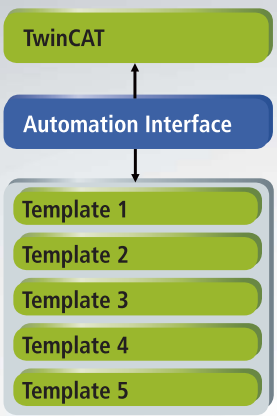
Applications of the TwinCAT Automation Interface

With the aid of the TwinCAT Automation Interface, complete projects can be generated or changed in a standard and flexible way. The application possibilities that result for the customer are extremely varied: They can range from a simple application for the automated addition of version information to PLC function blocks to complex user-defined engineering tools, which only use the regular TwinCAT configuration environment in the background. However, officially available TwinCAT supplement/function products also make use of the Automation Interface: for instance, the TwinCAT ECAD import tool, which uses the Automation Interface internally in order to generate a project from an XML file coming from an ECAD tool.

Users can find a detailed description of the Automation Interface, many solution examples and best practice articles in the Beckhoff Information System. Among other things a downloadable application example is provided that offers the customer a starting point for their own application and thus promises an initial sense of achievement in handling the Automation Interface. The application provides a certain number of different TwinCAT configurations on a graphic interface written in C# using WPF. These can be specified in detail in an XML file. The corresponding configuration is then automatically generated in the background at the push of a button and activated when necessary. The time required for this is just a few seconds – depending on the scope of the configuration to be generated. Such an application offers very high optimization potential for ap-



Architecture of the TwinCAT Automation Interface in TwinCAT 3



Use of templates for generating configurations

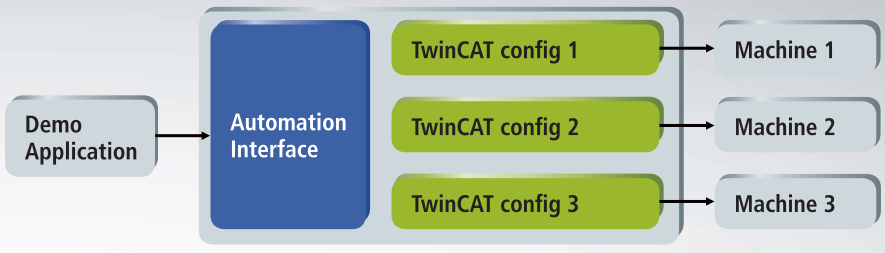
plications where the configuration of a machine must be changed several times a day, for example because tools are to be activated with other I/O systems. A machine operator can then activate the different configurations as required without TwinCAT know-how.

A further demo application shows the possible use of the Automation Interface from a Visual Studio® add-in. The application can be started when a TwinCAT project has been opened. It automatically adds version information (author, version) to all PLC function blocks in the project in the form of constants that can be read later on. Such an application underlines once more the usage possibilities and the flexibility in the integration of applications into the Visual Studio® development environment.

Author: Dr. Josef Papenfort, TwinCAT Product Manager, Beckhoff

Author: Sven Goldstein, TwinCAT Product Manager, Connectivity & Embedded Systems, Beckhoff

Further Information:
www.beckhoff.com/TwinCAT3
http://infosys.beckhoff.com



Structure of the example application