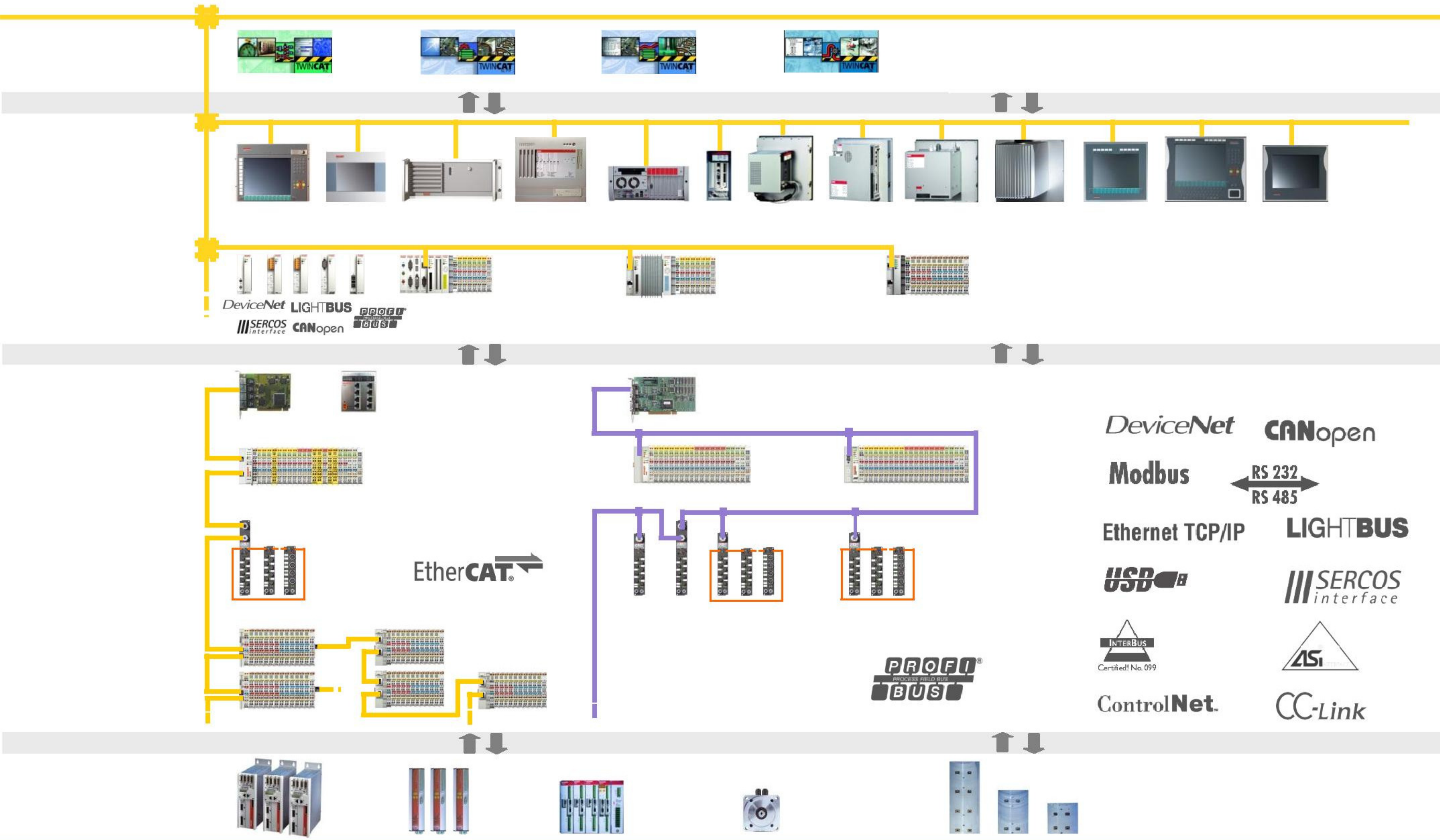
New Automation Technology

Beckhoff TwinCAT CNC Training

System Overview



Agenda

- Presentation of the Beckhoff CNC HMI application base
- Presentation of Main Features
  - User Management
  - Language Switching
  - Settings
  - CNC View
  - NPV Table
  - PLC Machine Data
- Adaptability
  - Parameter Setting via the Menu Manager
  - Integration in Visual Studio

The HMI Application Base (V 2.3.x)

- Beckhoff provides libraries for communication with ADS devices (e.g. PLC or CNC)
- These can be used with the appropriate programming languages and their development environments. These include:
  - Visual Basic
  - C / C++
  - Scripting Languages (VBScript, JScript)
  - VB.NET, C#
  - Java
- The HMI application presented here shows an example of how on this basis, an industry-standard visualization solution for CNC machines can be created.
- The following statements relate to the CNC HMI version 2.3.2!

The HMI Application Base (V 2.3.x)

- Complete visualization solution (not only) for CNC machines
- Free (no official product, no support) used as:
  - Example Application
  - Transitional solution
- Cost-bound in the context of:
  - Application projects
- Developed in C# (Windows Forms)
- Expandable via:
  - Configuration (e.g. key layout, logo, …)
  - Integration of new CIL class libraries

Presentation of the Main Features - Basics

- The CNC HMI (or HMI for short) can be stored with its folder structure at any location either locally or on a remote PC.

Auftrag	24.03.2011 11:32	Dateiordner	
Bitmap	24.03.2011 11:32	Dateiordner	
Customer	29.03.2011 13:30	Dateiordner	
externalReferences	28.03.2011 18:27	Dateiordner	
Log	24.03.2011 11:32	Dateiordner	
Modules	24.03.2011 11:32	Dateiordner	
PLC+TSM	24.03.2011 11:32	Dateiordner	
Plugins	24.03.2011 11:32	Dateiordner	
References	24.03.2011 11:32	Dateiordner	
System	06.05.2011 14:24	Dateiordner	
Beckhoff.App.Shell.Core.config	26.11.2010 15:09	XML Configuration File	6 KB
Beckhoff.App.Shell.Core.exe	24.03.2011 11:16	Anwendung	33 KB

- There is no installation or registration required.
- Start the application via Beckhoff.App.Shell.Core.exe (as pictured)
- After starting the HMI you will be presented with the default form

Presentation of the Main Features - Basics

TwinCAT and PLC Status

Customer Logo

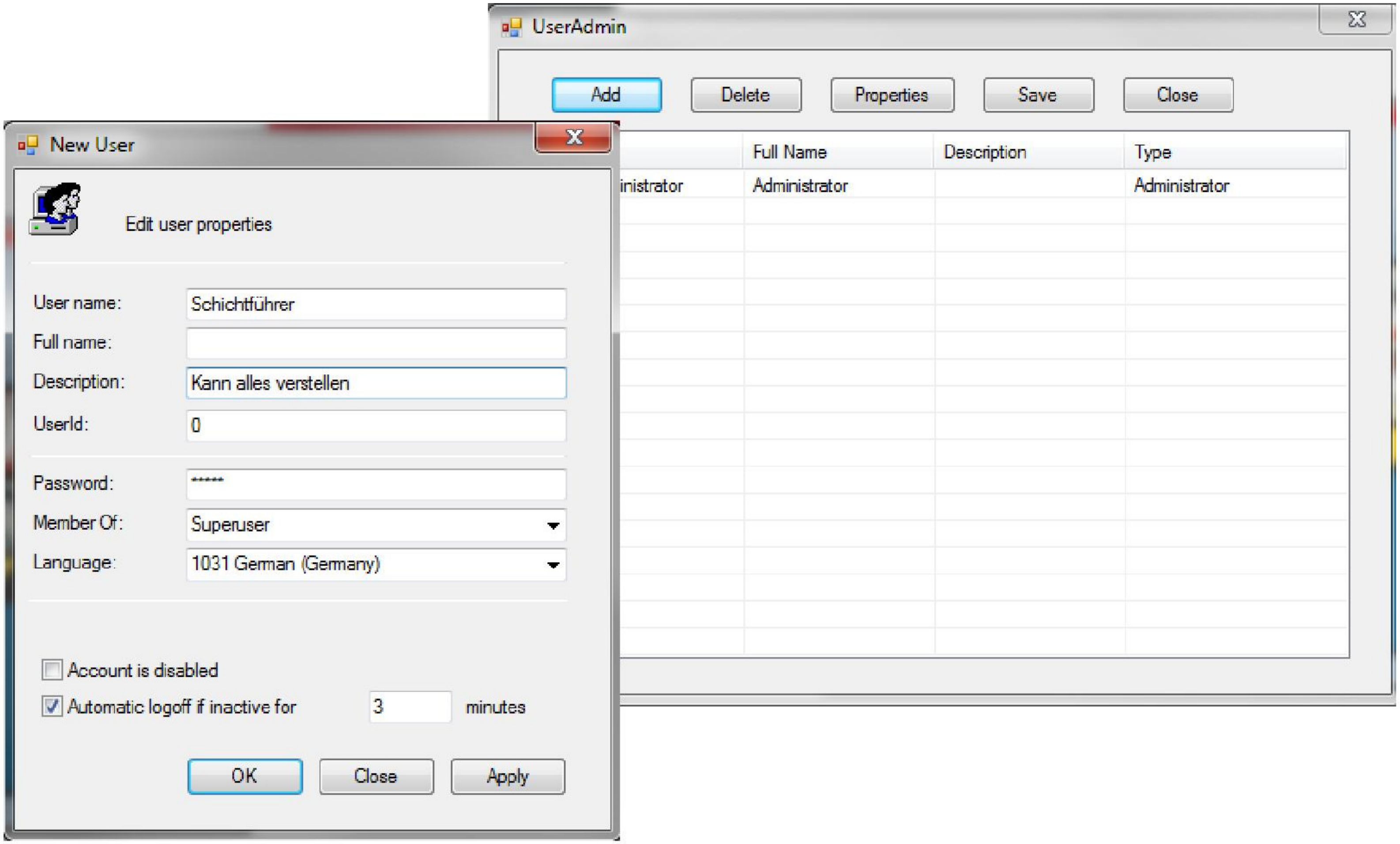
Display Area

Status Bar

Navigation Bar

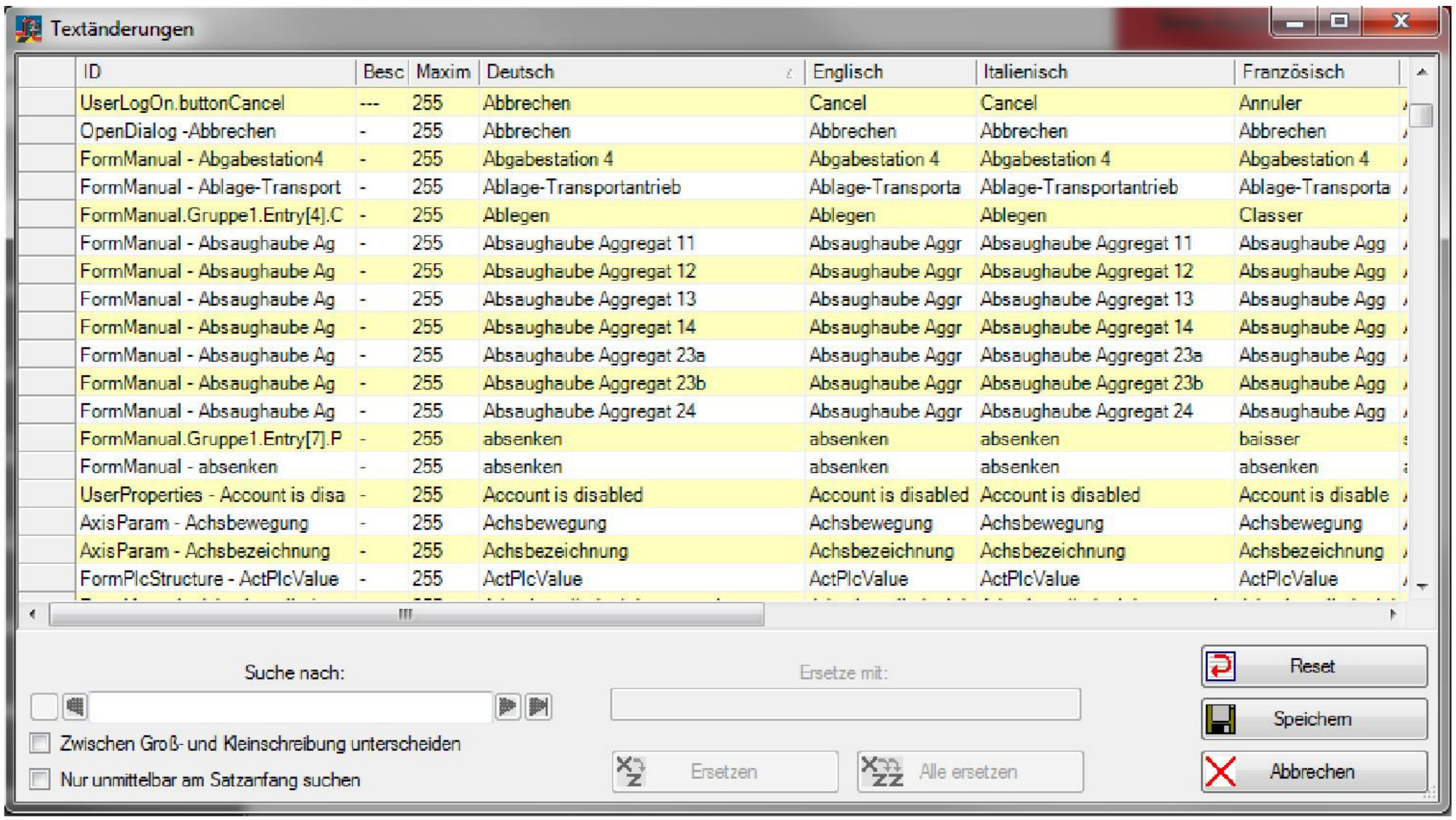
Presentation of the Main Features – User Management

- The HMI includes a flexible user management with an unlimited number of users divided into six different groups – passwords can be configured.



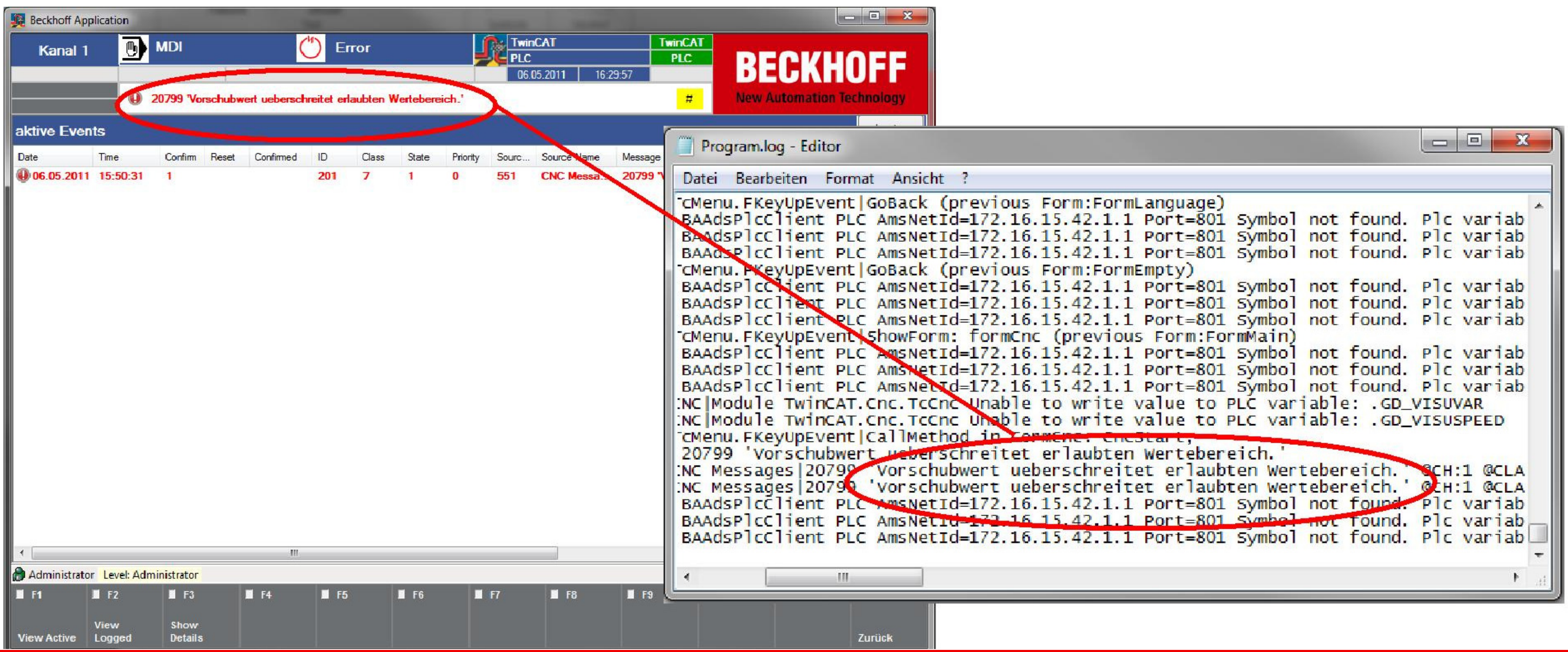
Presentation of Main Features – Language Switching

- Multiple Language Switching is available
- The various translations are stored in an Access database (found under "…\System\Texte.mdb") and can be edited via the HMI or and external program.



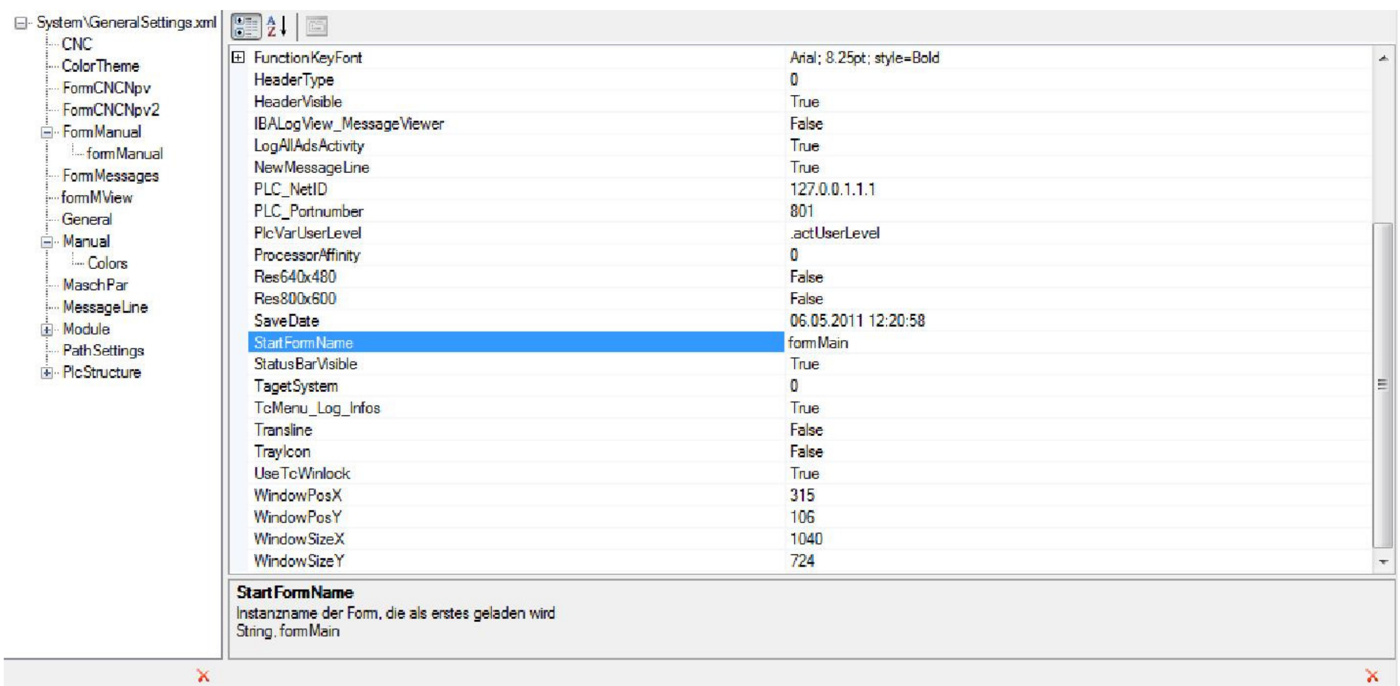
Presentation of the Main Features - Logging

- Major fault and status messages from the HMI, and also from the PLC/CNC are logged in log files (found under “…\Log\Program.log”).
- The open source logging framework “log4net” is used (more information can be found at http://logging.apache.org/log4net/index.html)
- The corresponding configuration file is located at „…\System\log.xml“



Presentation of the Main Features - Settings

- Settings that affect the appearance and behavior of the HMI are stored in the so-called Settings.
- The settings can be modified via the HMI.
- Keep in mind that some settings may require a restart of the HMI in order to be effective.
- The scope of the settings is largely dependent on the type and number of optional features and is usually heavily dependent on the customer.



Introduction of Other Features

- In addition to the previously presented, and fundamentally important for many applications, infrastructure components, the HMI has a number of useful features.
- This can have an application-specific background, or be generally valid.
- The high flexibility of the HMI application framework makes it possible to make such function available via configuration, as well as to remove them entirely
- Two of these components are discussed briefly below:
  - Zero Offset table for CNC
  - PLC machine data management

Introduction of Other Features – Zero Offsets

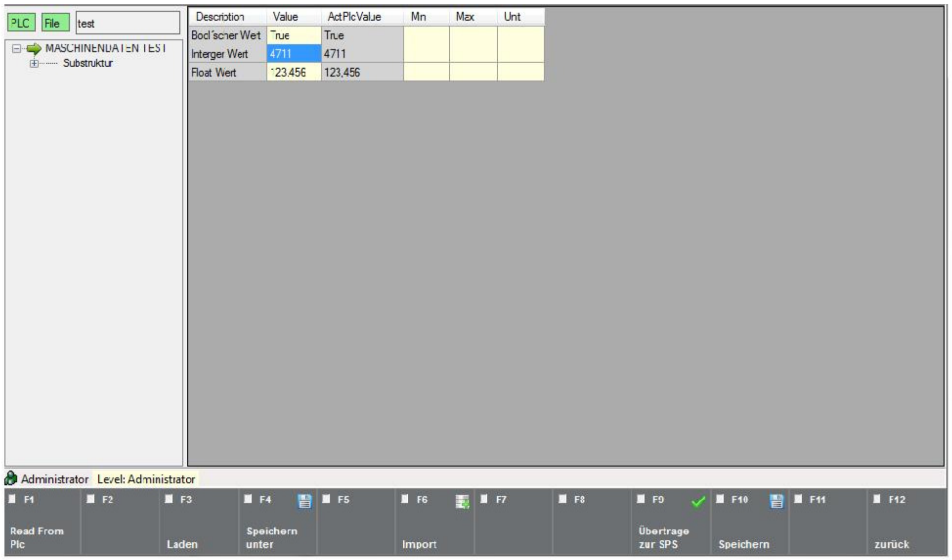
- The zero offset table in the HMI allows you to adjust the zero offset data for the CNC. Without this, the system manager must be used.
- The zero offset data can be stored in an external file, and if necessary (or automatically) loaded.
- NOTE: The adjusted zero offset data are loaded only on startup of the HMI. Therefore, this data is not loaded to the CNC on reboot until the HMI is started and sends the data.

default

Index	X	Y	Z
G54	2.00	12.00	0.00
G55	0.00	0.00	0.00
G56	0.00	0.00	0.00
G57	0.00	0.00	0.00
G58	0.00	0.00	0.00
G59	0.00	0.00	0.00

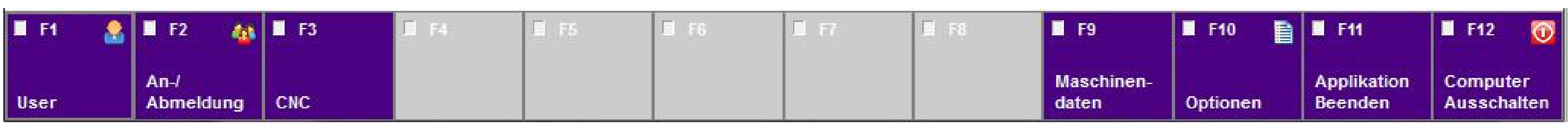
Introduction of Other Features – PLC Machine Data

- Machine data (setting values that are available in the PLC) can be managed on the HMI using the classes from the plugin “Beckhoff.App.PlcStructure.dll”
- The values have to be provided in the PLC with a specifically declared comment to indicate it as loadable and storable machine data.
- More information on this can be found under the documentation “…\Documents\PlcStructure.pdf”



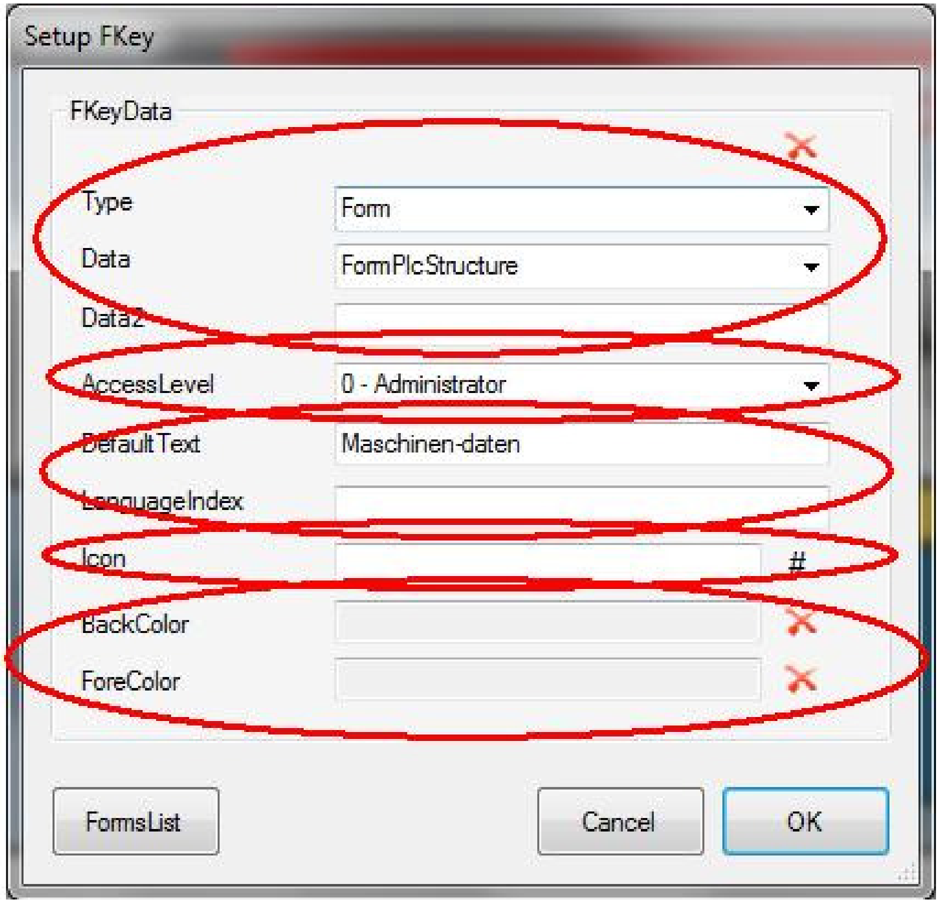
Customization Options – The Menu Manager

- In the Menu Manager is a component that manages the particular assignment of the function keys on the navigation bar.
- The Keys of the HMI are set at runtime:
  - Manually
  - Via Code
- Manual configuration of the Keys is stored in the file “…\System\tcMenu.xml”
- More information about the Menu Manager is located under “…\Documents\MenuManger.pdf”
- The configuration mode is activated by holding Ctrl and Alt and clicking a button. The same actions will also deactivate the configuration mode.



Customization Options – The Menu Manager

- While in configuration mode, a dialog box, with which the behavior of a Key can be set, is displayed by clicking one of the buttons.
- The dialog allows modification of:
  - The function type
  - The access authorization
  - The display text (including language database index)
  - The button icon
  - Foreground and background colors



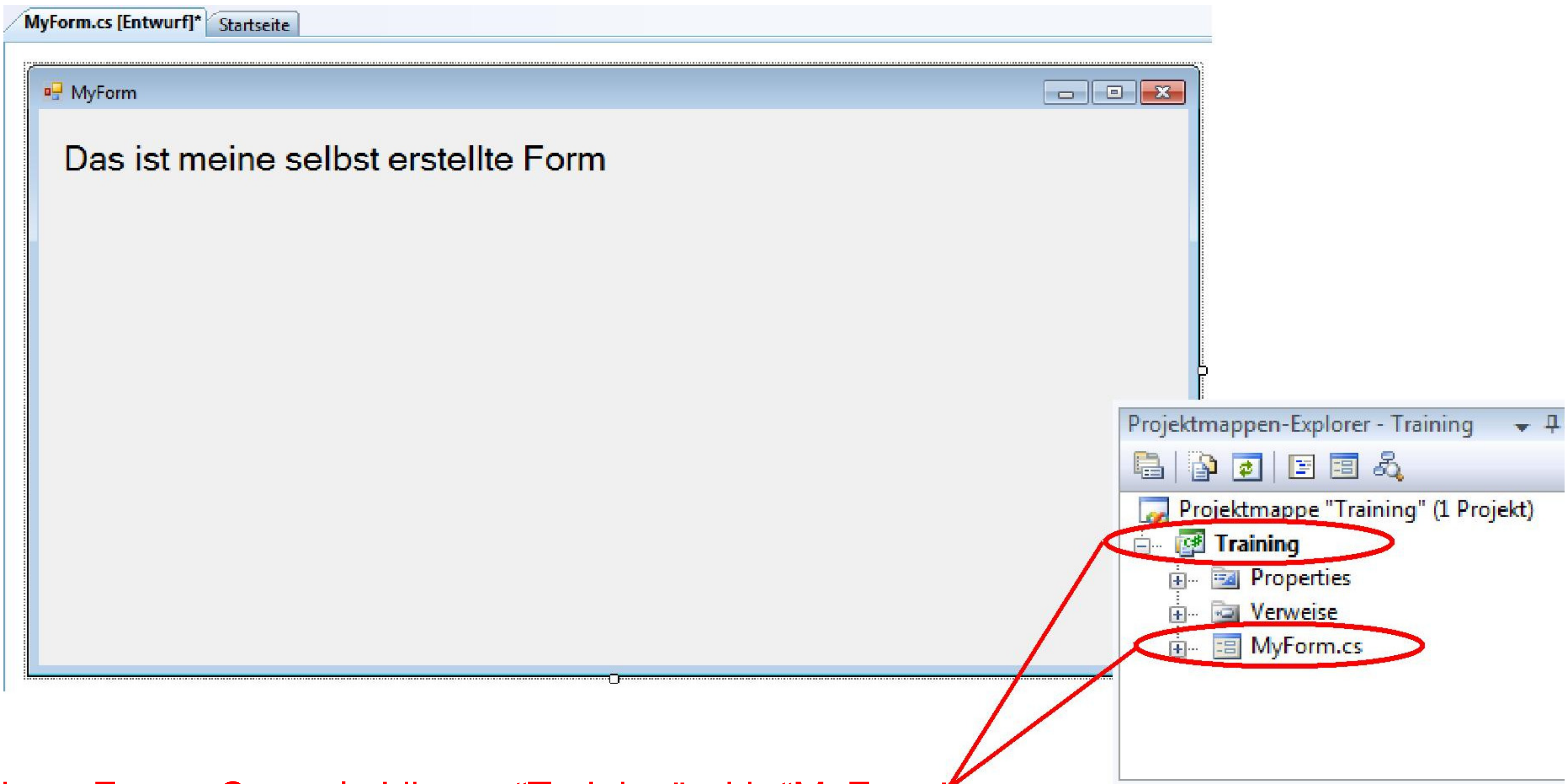
Customization Options – The Menu Manager

The following function types:

Form	Opens the specified in “Data”. The forms available to the Menu Manager must be made known via the “FormsList”.
callMethod	Calls a public void method declared in the current form. This method may accept a maximum of one parameter.
Back	Returns to the calling form.
startProcess	Starts an external application.
ShutDown	Ends the HMI and shuts down the computer.
ExitProgram	Ends the HMI
PlcSetAndResetVar	When pressed this Sets the PLC BOOL variable specified in DATA and Resets the variable when released.
PlcToggleVarAndFeedback	Toggles the PLC BOOL variable specified in DATA in the PLC
PlcToggleVarAndFeedbackImage	Toggles the PLC BOOL variable specified in DATA in the PLC. When the variable is TRUE the image specified in DATA 2 is displayed.

Customization Options – Use of Own CIL Classes

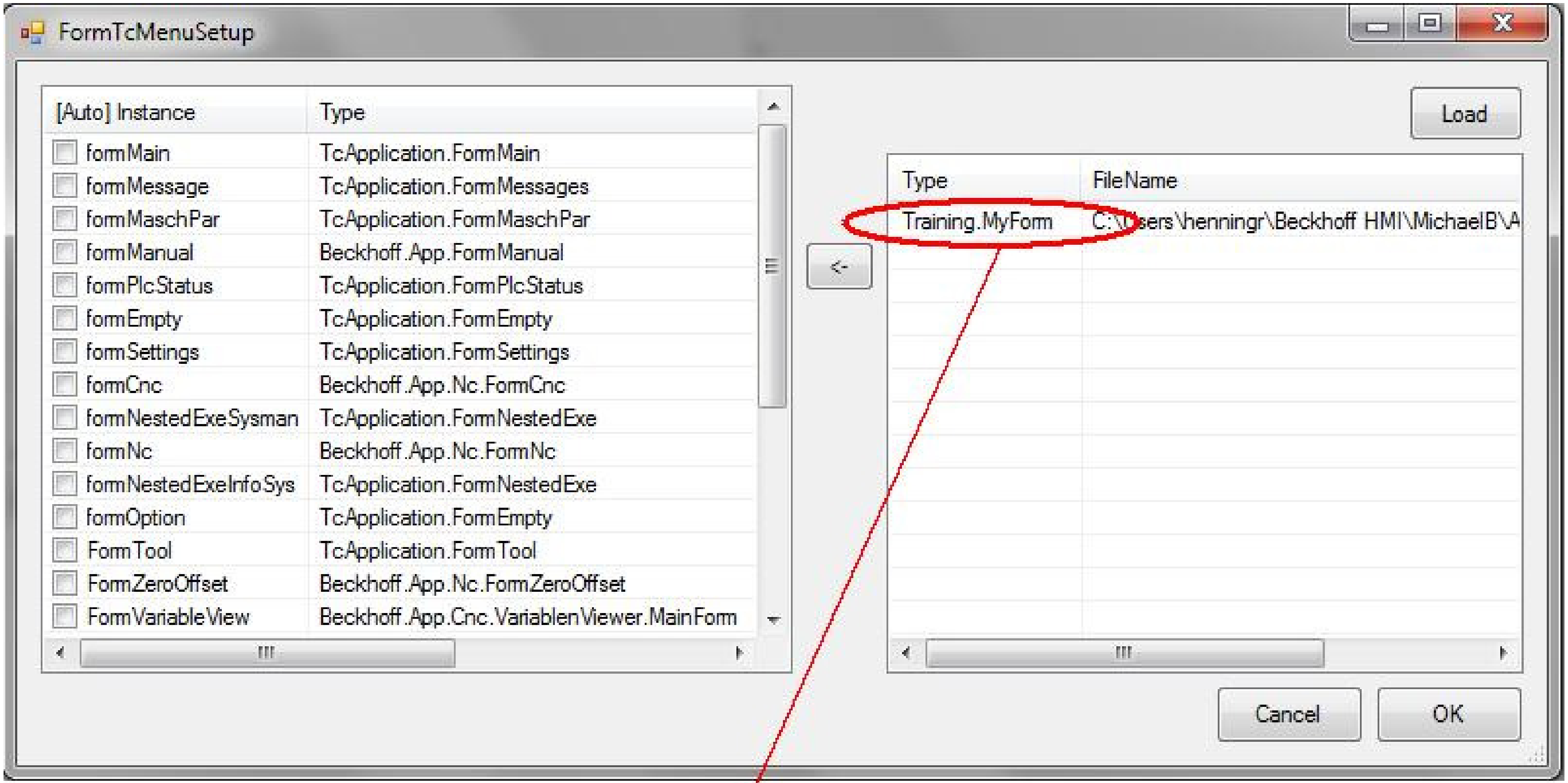
- The flexibility of the Menu Manager allows user developed Windows Forms to be used within the HMI.



Windows Forms Controls Library "Training" with "MyForm"

Customization Options – Use of Own CIL Classes

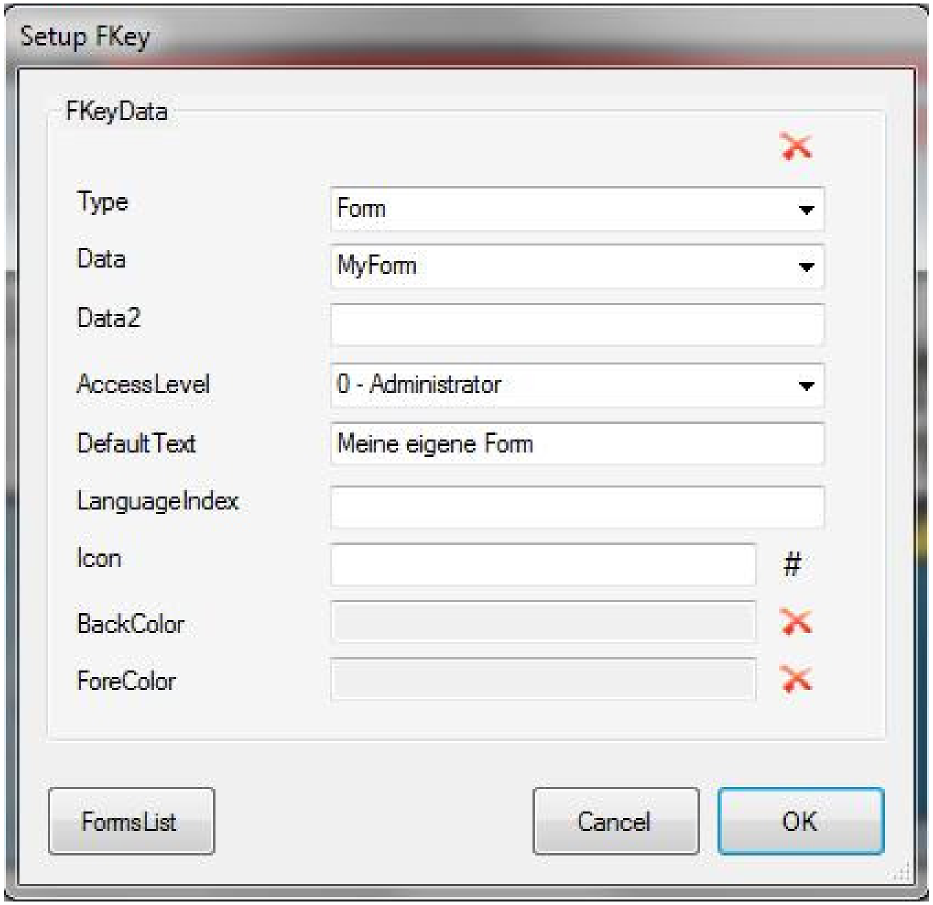
- First, however, are the Menu Manager has to know that the user form is available (“Form List” button in the configuration dialog)



Windows Forms Controls Library “Training” with Form “MyForm”

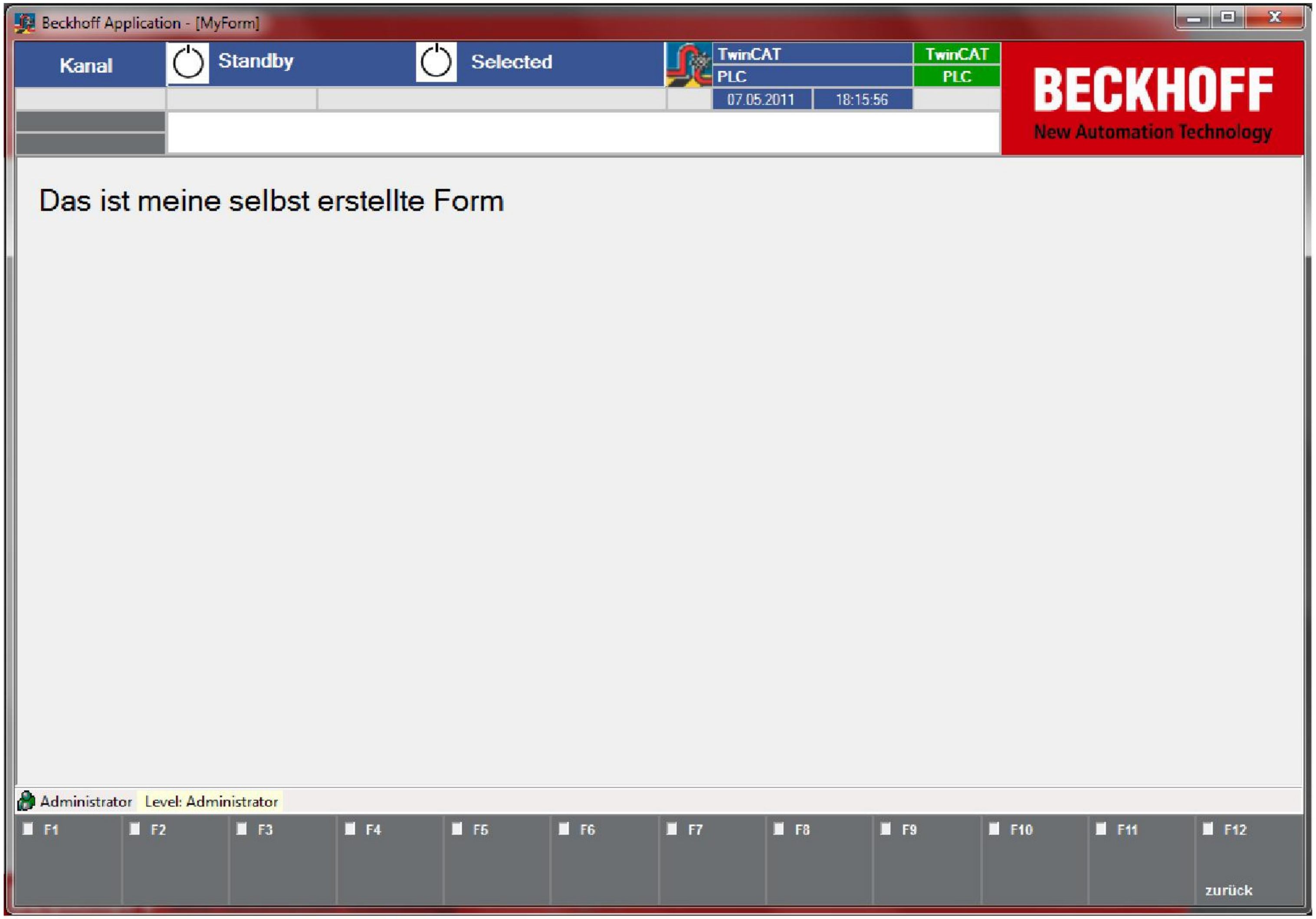
Customization Options – Use of Own CIL Classes

- The “final” configuration…



Customization Options – Use of Own CIL Classes

- … and the user form displayed in the HMI



Customization Options – Use of Own CIL Classes

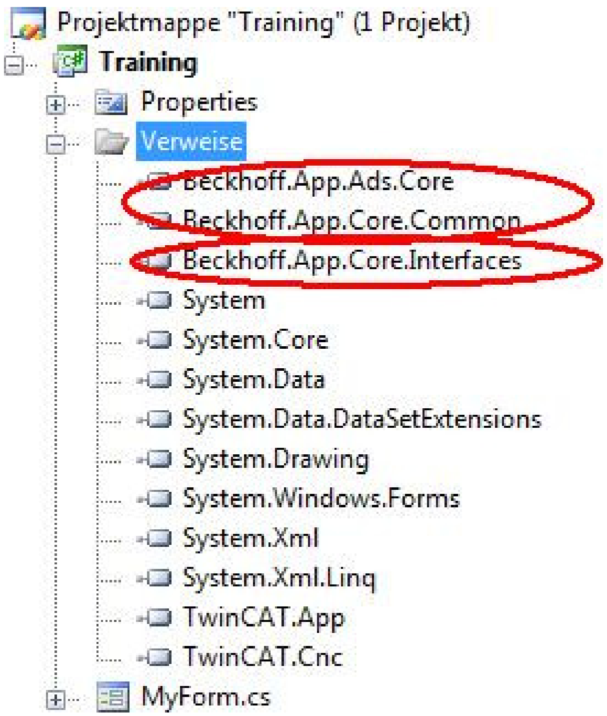
- A user created Windows Form can also access the infrastructure components (logger, language change, settings, ADS, …) of the HMI.
- To do this you must create a parameterized constructor with the needed interfaces.

```
public Form1(IBASettings settings, TcAdsServer adsServerOld): this()
{
```

- The HMI will then pass to the constructor the suitable implementations of these interfaces.
- This mechanism is called “dependency injection” (or DI for short) and is implemented in the HMI via the “Unity Application Block”.
- For more information see:
  - http://unity.codeplex.com
  - http://en.wikipedia.org/wiki/Dependency_Injection

Customization Options – Use of Own CIL Classes

- The usable interfaces are defined in the class libraries:
  - “Beckhoff.App.Ads.Core.dll”
  - “Beckhoff.App.Core.Interfaces.dll”
- These libraries must be referenced in the Windows Forms project.
- Depending on the functionality required more libraries may need to be referenced.



Customization Options – Use of Own CIL Classes

- The most important interfaces are:

IBAAdsServer	Allows use of the ADS communication. FOR EXAMPLE for reading and writing PLC variables (event-driven)
IBALanguage	Provides access to the content of the language database and execution of language switching functionality.
IBALogger	Provides the ability to write messages in to the log file.
IBAMenu	Provides access to the Menu Manager. e.g. for program-controlled settings of the buttons.
IBASettings	Provides read and write access to the application settings.

Customization Options – Use of Own CIL Classes

- The native Windows Forms class that uses the infrastructure of the HMI via DI can also take advantage of the full debugging functionality of the development environment.
- The HMI is not directly executed but instead is executed via the debugger.
- The following project properties need to be changed in Visual Studio.