BECKHOFF

(7.2.2013)

Standard Key Assignment of a Form

To set the default assignments of a form in the menu manager which is used after the first Import of the form, a "public static" method must be defined in the form that returns the return value type "Dictionary <int, IBAFkeyData>". In this Dictionary, the default assignment is stored and requested from the menu manager as needed. Example: Here is entered in the 5 button has a "callMethod" and the button 6 "PlcToggleVarAndFeedbackImage":

```
/// <summary>
/// Gets the default F key default assingment.
/// </summary>
/// <returns></returns>
public static Dictionary<int, IBAFKeyData> GetDefaultFKeyAssingment()
{
    var fkeyAssign = new Dictionary<int, IBAFKeyData>();
    var fkey = new BAFKeyDataStruct
    {
        AccessLevel = 2,
        EventType = "callMethod",
        Data = "CncStart",
        DefaultText = "Start",
        Icon = @"\Bitmap\State\ChStart.ico"
    };
    fkeyAssign.Add(5, fkey);


    fkey = new BAFKeyDataStruct
    {
        AccessLevel = 2,
        EventType = "PlcToggleVarAndFeedbackImage",
        Data = ".PlcAxisEnable",
        Data2 = @"\Bitmap\State\Enable2.ico",
        DefaultText = "Enable",
        Icon = @"\Bitmap\State\Enable.ico"
    };
    fkeyAssign.Add(6, fkey);
         return fkeyAssign;
   }
```

Beckhoff Automation GmbH
Eiserstraße 5
33415 Verl
Germany

Postfach 11 42
33398 Verl

Telefon: +49 (0) 52 46/9 63-0
Fax Zentrale: -149
Fax Vertrieb: -198
Fax Anlagentechnik: -379
Fax Service: -479
E-Mail: info@beckhoff.de
www.beckhoff.de

Geschäftsführer:
Dipl. Phys. Hans Beckhoff
Arnold Beckhoff
Registergericht: Gütersloh HRB 1803
Ust.-Id.-Nr.: DE 126787444
Finanzamt Wiedenbrück
St.-Nr. 347/5819/0016

Kreissparkasse Verl
BLZ 478 535 20
Kto.Nr. 4 000 766
SWIFT WELADED1WDB
IBAN DE114785352
00004000766

Deutsche Bank Gütersloh
BLZ 480 700 43
Kto.Nr. 371 / 7014
SWIFT DEUTDE3B480
IBAN DE854807004
00371701400

BECKHOFF

How do get to the new Form

- The constructor of the form is filled automatically by Unity with the corresponding parameters.
- If the parameter "string InstanceName" exists in the constructor, then the current instance name is passed.
- If the parameter "string ParameterMenuManager" exists in the constructor, then the parameter "Data2" from the key assignment of the calling fkeys is passed.
- Properties are automatically populated when you instantiate the form from the menu manager with instances of the appropriate types if they are available in the general object list. (Here, however, there is no guarantee when the properties are injected)
- All "public void" methods with no or one parameter can use the menu manager via "callMethod" and thus can be assigned to any function keys.
- Example:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Windows.Forms;
using TwinCAT.Cnc;
using TwinCAT.App;
using Beckhoff.App;

public partial class FormTest : Form
    {
        TcAdsServer _adsServer;
        ITCLManager _tclm;
        TcMenu _menu;
        Beckhoff.App.Settings _settings;
       private IBALanguage _language;

        public Beckhoff.App.Settings Settings
        {
            set { _settings = value; }
        }

        public ITcUserAdmin UserManager
        {
            set { _userManager = value; }
        }

        public TcMenu Menu
        {
            set { _menu = value; }
        }

        public ITCLManager Tclm
        {
            set { _tclm = value; }
        }

        public TcAdsServer AdsServer

        {
            set {
                _adsServer = value;
            }
        }

        public void EineMethodeZumAufrufen()
        {
            MessageBox.Show("Methode wurde aufgerufen");
        }
    }
```

```
        }

        public FormTest(IBALanguage language)
        {
            _language = language;
            InitializeComponent();
        }
    }}
```

Prevent a "Hide ()" call by the Menumanager

The HMI is configured as a MDI (Multiple Document Interface) application. All mounted Window (Forms) are displayed as a MDI child window. If another form is called, the Menumanager calls the Hide () method of the "old" Form and creates the "new" form. This process, especially with embedded ActiveX components causes problems:

ActiveX Controls unresponsive, events no longer occur in the Control

With the attribute "BAMenuAttributeNoHide" a class (form) can be marked to not Hide(). Instead the form in the MDI frame is minimized. The attribute is defined in “Beckhoff.App.Core.Interfaces".

Example:

```
/// <summary>
/// The form FormCNC_WPF.
/// </summary>
[BAMenuAttributeNoHide]
public partial class FormCNC_WPF : Form
{
        #region Fields
```

It can be integrated from the menu manager and WPF controls.

To this end, the Assembly "Beckhoff.App.Core.Interface" must be referenced and the control are marked with the attribute "BAWpfMainControl":

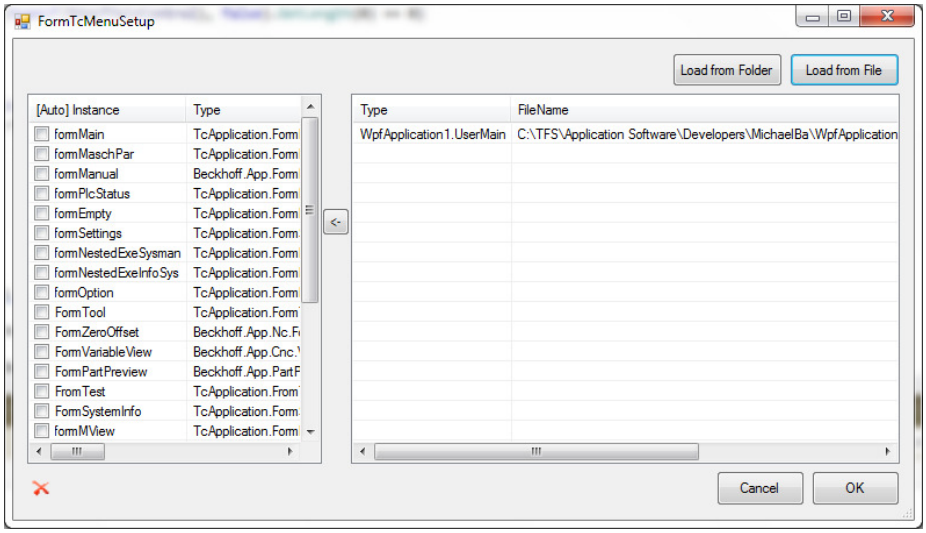
```
using Beckhoff.App.Core.Interfaces;

/// <summary>
/// Interaction logic for UserMain.xaml
/// </summary>
[BAWpfMainControl]
public partial class UserMain : UserControl
{
    public UserMain()
    {
        InitializeComponent();
    }

    public UserMain(IBALogger logger) : this()
    {
        logger.Write("Im Konstruktor des Controls", "WPFCOntrol", LogType.Info);
    }

}
```

The WPF UserContol can then be integrated with the Menumanager.



Again, Unity fills the constructor.

BECKHOFF

Announce the Menumanager from "outside" new event types

- The menu manager can be made known from "outside" new event types.
- To do this in a class a method attribute TcMenuEvent must be defined. As a parameter of the attribute, the event name is specified.
- The following parameters of the method are supported: a string of five parameters or a parameter of type TcMenuEventParameter
- About the method of Menumanagers `AddMenuExtension(Object o);` is the Menumanger then given the class known.

E.g.:

```
using System;
using System.Collections.Generic;
using System.Text;
using Beckhoff.App;

namespace TcApplication
{
    class TcMenuPlcConnect
    {

        [TcMenuEvent("PlcSetVar")]
        public void SetVarInPlc(string plcVar)
        {
            System.Windows.Forms.MessageBox.Show("PlcSetVar aufgerufen mit dem Parameter:
" + plcVar);
        }

        [TcMenuEvent("PlcToggleVar")]
        public void ToggleVarInPlc(string plcVar)
        {
            System.Windows.Forms.MessageBox.Show("PlcToggleVar aufgerufen mit dem
Parameter: " + plcVar);
        }
    }
}

Somewhere in the code:
TcMenuPlcConnect menuPlcConnet = new TcMenuPlcConnect();
menuLocal.AddMenuExtension(menuPlcConnet);
```

BECKHOFF

- To get more information on the calling button the possibility to implement a method with 5 parameters exists (example):

```
        [TcMenuEvent("PlcToggleVarAndFeedback")]
        public void ToggleVarInPlcWithColor(string plcVar, string plcVar2, TwinCAT.App.TcFKey
fkey, int index, TcMenuEvent.EventReason reason)
```

- Significance of the parameter:
  - plcVar: Parameter from Data1
  - plcVar2: Parameter from Data2
  - fKey: Key group that triggered the event
  - index: Key index of Events
  - reason: Reason of Events:
    ```
    KeyDown: Key is pressed
    KeyUp: Key is released
    ElementShow: Button is displayed
    ElementHide: Button is hidden
    ```
- Other Possibilities:

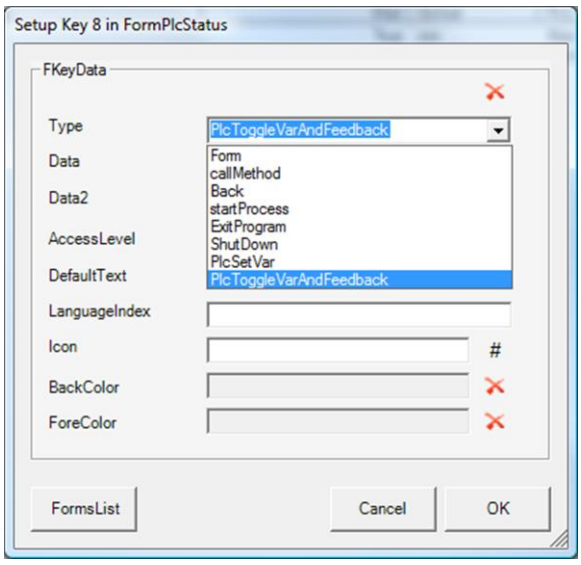
```
[TcMenuEvent("PlcToggleVarAndFeedback")]
public void ToggleVarInPlcWithColor(TcMenuEventParameter param)
    switch (param.Reason)
    {
        case TcMenuEvent.EventReason.KeyUp: ……..
```

The class contains TcMenuEventParameter with the elements described above. The class TcMenuEventParameter additionally contains the parameter

- InstanceName: Instance name of the current form

BECKHOFF

- Now the Event Types "PlcSetVar" and "PlcToggleVarAndFeedback" are known in the menu manager and can be selected. If now a menu button is pressed, the method is called and the parameters from the Data and Data2 field are passed.
  (These two types are the standard already implemented)



BECKHOFF

Remap the Keys from the outside at runtime

To change the keyboard layout at runtime, there is in IBAMenu methods

```
/// <summary>
/// Gets the F key assignment
/// </summary>
/// <param name="token">The token (key index).</param>
/// <returns></returns>
IBAFKeyData GetFKeyAssign(int token);

/// <summary>
/// Sets the F key assignment.
/// </summary>
/// <param name="data">The data.</param>
/// <param name="token">The token.</param>
void SetFKeyAssign(IBAFKeyData data, int token);
```

With GetFKeyAssign, the assignment of a key (identified by their ID Ex 1 = F1, 13 = ALT F1) can be read. In IBAFKeyData all the information of a key are stored. (Color, label, function). With SetFKeyAssign the assignment can be given to the Menumanager. This should be called (in Windows Forms) after the forms Visible has gone to TRUE (Event Visible Changed). Immediately after calling the method, the new values are active.

Example for confirming the ALT-F3 key with the command "start process" and CMD.EXE:

```
var data = _menu.GetFKeyAssign(15);
if (data != null)
{
    data.BackColor = Color.Salmon;
    data.DefaultText = "start CMD";
    data.EventType = "startProcess";
    data.Data = "cmd.exe";
    data.Data2 = "";

    _menu.SetFKeyAssign(data, 15);
}
```

BECKHOFF

Call an „public void“ Methods of Other Forms

Using the methods defined in IBAMenumanager

```
/// <summary>
/// Calls the method of the form.
/// </summary>
/// <param name="formName">Name of the form.</param>
/// <param name="methodName">Name of the method.</param>
/// <param name="parameter">The parameter.</param>
/// <returns></returns>
bool CallMethod(string formName, string methodName, object parameter);

/// <summary>
/// Calls the method of the form.
/// </summary>
/// <param name="form">The form.</param>
/// <param name="methodName">Name of the method.</param>
/// <param name="parameter">The parameter.</param>
/// <returns></returns>
bool CallMethod(Form form, string methodName, object parameter);
```

methods of other already instantiated Forms are able to be called with the proper parameter supplied.

Example:

Here, in the form with the instance name "formCnc" the method "test" with the parameter called type IBAPlcStructure:

```
var form = _menu.GetForm("formCnc");
_menu.CallMethod(form, "Test", this as IBAPlcStructure );
```

Beckhoff Automation GmbH
Eiserstraße 5
33415 Verl
Germany

Postfach 11 42
33398 Verl

Telefon: +49 (0) 52 46/9 63-0
Fax Zentrale: -149
Fax Vertrieb: -198
Fax Anlagentechnik: -379
Fax Service: -479
E-Mail: info@beckhoff.de
www.beckhoff.de

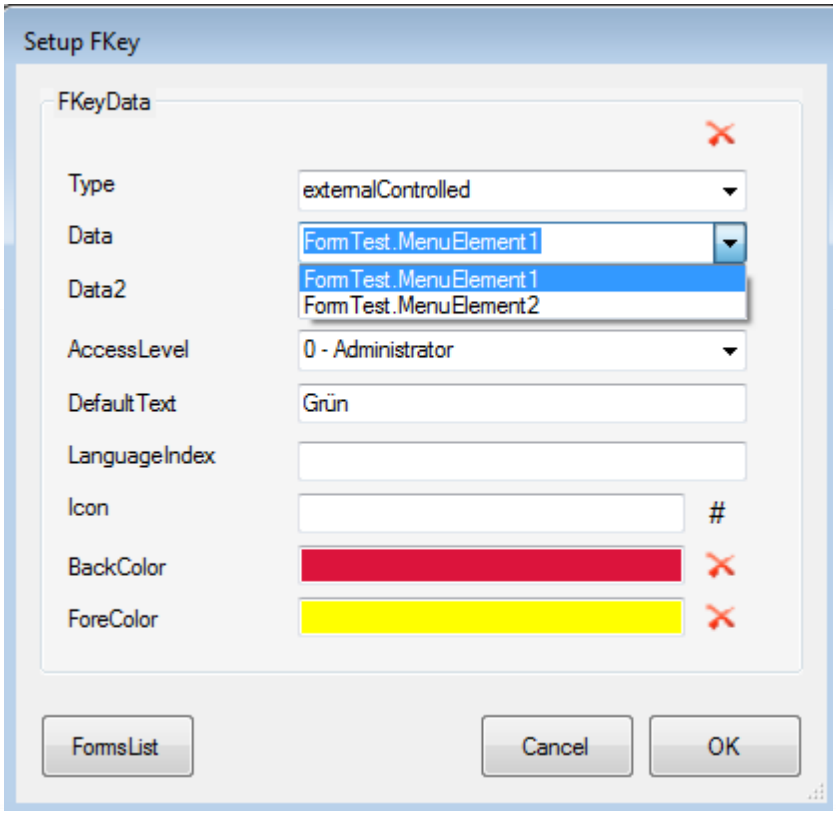
Geschäftsführer:
Dipl. Phys. Hans Beckhoff
Arnold Beckhoff
Registergericht: Gütersloh HRB 1803
Ust.-Id.-Nr.: DE 126787444
Finanzamt Wiedenbrück
St.-Nr. 347/5819/0016

Kreissparkasse Verl
BLZ 478 535 20
Kto.Nr. 4 000 766
SWIFT WELADED1WDB
IBAN DE114785352
00004000766

Deutsche Bank Gütersloh
BLZ 480 700 43
Kto.Nr. 371 / 7014
SWIFT DEUTDE3B480
IBAN DE854807004
00371701400

BECKHOFF

"ExternalControlled" define functions and integrate



In the interfact IBAMenu ther is a method "DefineExternalMenuData" that can be used to access defined external functions of the Menumanager:

```
/// <summary>
/// Publishes the external menu data.
/// </summary>
/// <param name="handle">The handle.</param>
/// <param name="data">The data.</param>
/// <param name="overwrite">if set to <c>true</c> existing data is
overwritten.</param>
/// <returns>
/// true if successful
/// </returns>
 bool DefineExternalMenuData(IBAToken handle, IBAFKeyData data, bool overwrite);
```

BECKHOFF

Example for creating "FormTest.MenuElement1": ("menu" has the Interface IBAMenu)

```
BAFKeyDataStruct _fkeydata
…
…
…
_fkeydata = new BAFKeyDataStruct
{
BackColor = Color.Chartreuse,
DefaultText = "MenuEvent1",
UpEventDelegate = this.UpEventDelegate,
DownEventDelegate = this.DownEventDelegate
};
menu.DefineExternalMenuData(new StringToken("FormTest.MenuElement1"), _fkeydata , false);
…
…
…
private void DownEventDelegate(IBAToken token)
{
; // hier Code für "drücken der Taste" hinzufügen
}
private void UpEventDelegate(IBAToken token)
{
; // hier Code für "lösen der Taste" hinzufügen
}
```

The methods "DownEventDelegate" and "UpEventDelegate" are called by the menu manager upon the corresponding events. At runtime, the assignment at any time be changed can through additional calls to "ReceiveExternalMenuData": (here is a Timer Method "_timerMib_Elapsed" called cyclically:

```
void _timerMib_Elapsed(object sender, System.Timers.ElapsedEventArgs e)
{
this._fkeydata.BackColor =
this._fkeydata.BackColor.Equals(Color.Chartreuse) ? Color.Crimson : Color.Chartreuse;
this._fkeydata.DefaultText =
this._fkeydata.DefaultText.Equals("Grün") ? "Rot" : "Grün";
 _menu.DefineExternalMenuData(new StringToken("FormTest.MenuElement1"), _fkeydata, true);
}
```

Beckhoff Automation GmbH
Eiserstraße 5
33415 Verl
Germany

Postfach 11 42
33398 Verl

Telefon: +49 (0) 52 46/9 63-0
Fax Zentrale: -149
Fax Vertrieb: -198
Fax Anlagentechnik: -379
Fax Service: -479
E-Mail: info@beckhoff.de
www.beckhoff.de

Geschäftsführer:
Dipl. Phys. Hans Beckhoff
Arnold Beckhoff
Registergericht: Gütersloh HRB 1803
Ust.-Id.-Nr.: DE 126787444
Finanzamt Wiedenbrück
St.-Nr. 347/5819/0016

Kreissparkasse Verl
BLZ 478 535 20
Kto.Nr. 4 000 766
SWIFT WELADED1WDB
IBAN DE114785352
00004000766

Deutsche Bank Gütersloh
BLZ 480 700 43
Kto.Nr. 371 / 7014
SWIFT DEUTDE3B480
IBAN DE854807004
00371701400