BECKHOFF

(11.03.2020)

Standard Tastenbelegung einer Form definieren

Um die Standardbelegung einer Form im Menumanager festzulegen, die beim ersten Importieren der Form benutzt wird, muss in der Form eine „public static“ Methode definiert werden, die als Rückgabewert ein „Dictionary<int, IBAFkeyData>“ zurück gibt. In diesem Dictionary wird die Standardbelegung gespeichert und bei Bedarf vom Menumanager angefragt. Beispiel: Hier wird in der Taste 5 ein „callMethod“ eingetragen und in der Taste 6 ein „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: -1 49
Fax Vertrieb: -1 98
Fax Anlagentechnik: -3 79
Fax Service: -4 79
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

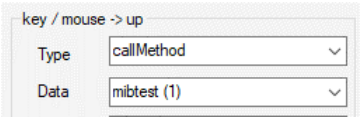
Kreissparkasse Verl
BLZ 478 535 20
Kto.Nr. 4 000 766
SWIFT WELADED1WDB
IBAN DE114785352
00004000766

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

BECKHOFF

Wie kommen Objekte zum neuen Form?

- Der Konstruktor der Form wird per Unity mit den entsprechenden Parametern automatisch gefüllt.
- Ist im Konstruktor der Parameter „String InstanceName“ vorhanden, wird dort der aktuelle Instanzname injiziert.
- Ist im Konstruktor der Parameter „String ParameterMenuManager“ vorhanden, wird dort der Parameter „Data2“ aus der Tastenbelegung des aufrufenden FKeys übergeben.
- Properties werden automatisch beim Instanzieren der Form vom Menumanager mit Instanzen der passsenden Typen gefüllt, falls diese in der allgemeinen Objektliste verfügbar sind. (Hier ist allerdings nicht sichergestellt, wann die Properties injiziert werden)
- Alle „Public void“ Methoden mit keinem oder einem Parameter können über den Menumanager per „callMethod“ aufgerufen werden und somit auf beliebige Funktionstasten gelegt werden.
  Bei Methoden mit einem Parameter wird versucht den übergebenen Parameter (String) auf den Zieltyp zu konvertieren.
  Methoden mit einem Parameter können einen zweiten Parameter vom Typ IBAFkeyData haben. Hier werden dann die Daten des aufrufenden Keys mitgegeben.



- Beispiel:

```
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; }
```

BECKHOFF

```
        }

        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();
        }
    }}
```

Verhindern eine „Hide()“ Aufrufs durch den Menumanager

Die HMI ist als MDI (Multiple Document Interface) Applikation aufgebaut. Alle eingebundenen Fenster (Forms), werden als MDI Child Fenster angezeigt. Wird ein anderes Form aufgerufen, so ruft der Menumanager die Methode Hide() des „alten“ Forms auf und stellt das „neue“ Form dar. Dieses Verfahren kann insbesondere bei eingebetteten ActiveX Komponenten Probleme hervorrufen:

> ActiveX Controls reagieren nicht mehr, Events kommen nicht mehr im Control an

Mit dem Attribut „BAMenuAttributeNoHide“ kann eine Klasse (Form) gekennzeichnet werden, so dass bei einer so gekennzeichneten Klasse kein Hide() mehr aufgerufen wird. Stattdessen wird die Form im MDI Rahmen minimiert. Definiert ist das Attribut in „Beckhoff.App.Core.Interfaces“. Beispiel:

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

BECKHOFF

Form automatisch schließen (Close) statt es zu minimieren

Wird ein anderes Form aufgerufen, so ruft der Menumanager die Methode Hide() des „alten“ Forms auf und stellt das „neue“ Form dar.
Mit dem Attribute `BAMenuAttributeCloseInsteadOfHide` wird das Schließen (Close) des Forms erzwungen statt es zu minimieren.
Dies hat zur Folge, dass jeder weitere Aufruf des Forms einen erneuten Konstrukoraufruf beinhaltet.

Beispiel:

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

Beckhoff Automation GmbH
Eiserstraße 5
33415 Verl
Germany

Postfach 11 42
33398 Verl

Telefon: +49 (0) 52 46/9 63-0
Fax Zentrale: -1 49
Fax Vertrieb: -1 98
Fax Anlagentechnik: -3 79
Fax Service: -4 79
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

Dem Menumanager von „außen“ neue Eventtypen bekannt geben

- Dem Menumanager können von „außen“ neue Event – Typen bekannt gemacht werden.
- Dazu muss in einer Klasse für eine Methode das Attribut `TcMenuEvent` definiert werden. Als Parameter des Attributs wird der Eventname angegeben.
- Folgende Parameter der Methode werden unterstützt: ein String oder fünf Parametern oder einem Parameter vom Typ TcMenuEventParameter
- Über die Methode des Menumanagers
  `AddMenuExtension(Object o);`
  wird dem Menumanger die Klasse dann bekannt gegeben.

Bsp:

```
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);
        }
    }
}

Irgendwo im Code:
TcMenuPlcConnect menuPlcConnet = new TcMenuPlcConnect();
menuLocal.AddMenuExtension(menuPlcConnet);
```

Beckhoff Automation GmbH
Eiserstraße 5
33415 Verl
Germany

Postfach 11 42
33398 Verl

Telefon: +49 (0) 52 46/9 63-0
Fax Zentrale: -1 49
Fax Vertrieb: -1 98
Fax Anlagentechnik: -3 79
Fax Service: -4 79
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

- Um mehr Informationen über die aufrufende Taste zu bekommen, besteht auch die Möglichkeit eine Methode mit 5 Parametern zu implementieren (Beispiel):

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

- Bedeutung der Paramter:
  - plcVar: Parameter aus Data1
  - plcVar2: Parameter aus Data2
  - fKey: Tastengruppe, die das Event ausgelöst hat
  - index: Tastenindex des Events
  - reason: Grund des Events:
    - KeyDown: Taste wurde gedrückt
    - KeyUp: Taste wurde losgelassen
    - ElementShow: Taste wurde angezeigt
    - ElementHide: Taste wurde versteckt

- Weitere Möglichkeit:

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

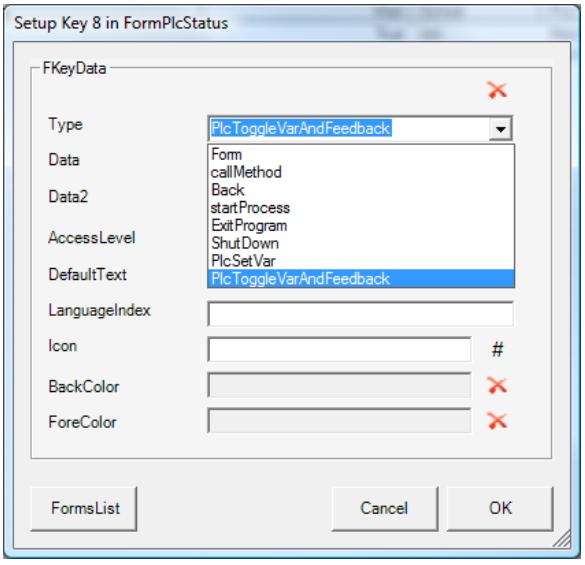
wobei die Klasse TcMenuEventParamter die oben beschriebenen Elemente enthält.

Die Klasse TcMenuEventParameter enthält zusätzlich den Parameter

- InstanceName: Instanzname des aktuellen Form

BECKHOFF

- Nun sind die EventTypen „PlcSetVar“ und „PlcToggleVarAndFeedback“ im Menumanager bekannt und können ausgewählt werden. Wird nun eine Menutaste betätigt, so wird die Methode aufgerufen und die Parameter aus dem Data und Data2 Feld werden übergeben.
  (Diese beiden Typen sind im Standard bereits implementiert)



BECKHOFF

Tastenbelegung von außen zur Laufzeit ändern

Um die Tastenbelegung zur Laufzeit zu ändern, gibt es im IBAMenu die Methoden

```
/// <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);
```

Mit GetFKeyAssign kann die Belegung einer Taste
(identifiziert durch ihre ID Bsp 1 = F1, 13 = ALT F1) ausgelesen werden. In IBAFKeyData sind alle Informationen einer Taste abgelegt. (Farbe, Beschriftung, Funktion).
Mit SetFKeyAssign kann eine eigene Belegung dem Menumanager bekannt gegeben werden. Diese sollte (in Windows Forms) aufgerufen werden, nachdem Visible der Form auf TRUE gegangen ist (Event VisibleChanged).
Unmittelbar nach dem Aufruf der Methode werden die neuen Werte aktiv.

Beispiel zum Belegen der ALT-F3 Taste mit dem Kommando „startProcess“ und 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 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

Beliebige „public void“ Methoden anderer Forms aufrufen

Mit Hilfe der im IBAMenumanager definierten Methoden

```
/// <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);
```

können Methoden von anderen schon instanziierten Forms aufgerufen und mit einem Parameter versorgt werden.

Beispiel:

Hier wird in dem Form mit dem Instanznamen „formCnc“ die Methode „Test“ mit dem Parameter vom Typ IBAPlcStructure aufgerufen:

```
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: -1 49
Fax Vertrieb: -1 98
Fax Anlagentechnik: -3 79
Fax Service: -4 79
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