

## Beschreibung der blue office® Add-In – Möglichkeiten

### Inhaltsverzeichnis

1.	Benutzersteuerelemente .....	2
1.1	Verweise .....	3
1.2	Klassen .....	3
1.3	Benutzersteuerelemente .....	3
1.4	Klasse: AddIn .....	4
1.5	Benutzersteuerelement: Adresse_Control.....	5
2.	Menupunkt im Programmregister .....	7
2.1	Verweise .....	8
2.2	Klassen .....	8
2.3	Klasse: AddIn .....	9
2.4	Klasse: AddInTabMenu.....	10
3.	Menupunkt in der Applikation zum Ausführen einer Funktion .....	12
3.1	Verweise .....	13
3.2	Klassen .....	13
3.3	Klasse: AddIn .....	14
3.4	Klasse: AddInMenu.....	15
4.	Menupunkt in der Applikation zum Öffnen eines neuen Registers .....	16
4.1	Verweise .....	17
4.2	Klassen .....	17
4.3	Benutzersteuerelemente .....	17
4.4	Klasse: AddIn .....	18
4.5	Klasse: AddInMenu.....	19
4.6	Benutzersteuerelement: Applikation_Control .....	21

## 1. Benutzersteuerelemente

Für den Form-Designer der Adressverwaltung, Artikelverwaltung, Projektverwaltung, Belegkopf usw.

## 1.1 Verweise

- ➔ [blueoffice.common](#) (blueoffice.common.dll)
- ➔ [blueoffice.menu](#) (blueoffice.menu.dll)
- ➔ [blueoffice.DAL](#) (blueoffice.DAL.dll)  
(DAL = **D**ata **A**ccess **L**ayer,  
Wird nur benötigt, wenn auf die aktuell gewählte  
Adresse, Artikel, usw. zugegriffen werden soll)

## 1.2 Klassen

- ➔ [AddIn.vb](#) (z.B.)  
Interfaces:
  - [blueoffice.common.AddIn.IAddInWithLicence](#)  
(Schnittstelle für AddIns mit individuellem oder universal-Lizenzschlüssel)  
oder:  
[blueoffice.common.AddIn.IAddIn](#)  
(Schnittstelle für AddIns ohne Lizenzschlüssel (Lizenzfile)  
→ Bitte diese Schnittstelle nur als Integrator verwenden, damit AddIns  
getestet werden können. Fertig erstellte Kunden AddIns haben  
üblicherweise einen Lizenzschlüssel, weshalb dafür die erst erwähnte  
Schnittstelle verwendet werden muss

## 1.3 Benutzersteuerelemente

- ➔ [Adresse\\_Control.vb](#) (z.B.)  
Interfaces :
  - [blueoffice.common.AddIn.IAddInControl](#)
  - [blueoffice.menu.IFrameControl](#)  
oder:  
[blueoffice.menu.IFrameControlMenu](#)  
(nur wenn Contextmenueingaben abgefragt oder ausgelöst werden sollen)

## 1.4 Klasse: AddIn

```
Public Class AddIn

    Implements blueoffice.common.AddIn.IAddInWithLicence

    Public Function GetInfo() As blueoffice.common.AddIn.AddInInfo _
    Implements blueoffice.common.AddIn.IAddIn.GetInfo

        Dim RetVal As New blueoffice.common.AddIn.AddInInfo
        RetVal.CompanyName = "Ihr Firmenname AG"
        RetVal.Name = "Mein AddIn"

        RetVal.AddToList(GetType(Adresse_Control), _
        blueoffice.common.AddIn.AddInShared.AddInDesitnation.ADR, _
        "Mein Adress-Benutzersteuerelement", _
        "Zeigt (z.B.) Informationen in der Adressverwaltung an")

        Return RetVal

    End Function

End Class
```

## 1.5 Benutzersteuerelement: Adresse\_Control

```

Public Class Adresse_Control

    Implements blueoffice.common.AddIn.IAddInControl
    Implements blueoffice.menu.IFrameControl

    #Region "Interface IAddInControl"

    Public Function Description() As String _
    Implements blueoffice.common.AddIn.IAddInControl.Description

        '** Beschreibung des AddIns
        Return "Zeigt (z.B.) Informationen in der Adressverwaltung an
(Adresse_Control)"
    End Function

    Private Function IAddInControl_Name() As String _
    Implements blueoffice.common.AddIn.IAddInControl.Name

        '** Name des AddIns
        Return "Mein Adress-Benutzersteuerelement (Adresse_Control)"

    End Function

    #End Region

    #Region "Interface"

    Public Property CurrentObject As Object _
    Implements blueoffice.menu.IFrameControl.CurrentObject
    Get
        Return Nothing
    End Get
    Set(value As Object)

        '** Hier erhalten Sie die Aktuell gewählte Adresse, Artikel, usw.
        '** als DAL-Objekt

        If value IsNot Nothing Then

            If TypeOf (value) Is blueoffice.DAL.A_ADRESSEN Then

                Dim tADR As blueoffice.DAL.A_ADRESSEN =
                DirectCast(value, blueoffice.DAL.A_ADRESSEN)

                If tADR IsNot Nothing AndAlso tADR.HasData Then

                    Dim ADR_ID As Integer = 0
                    ADR_ID = tADR.ADR_ID

                End If

            End If

        End If

    End Set
    End Property
    Public Event ObjectChanged As
blueoffice.menu.IFrameControl.ObjectChangedEventHandler Implements
blueoffice.menu.IFrameControl.ObjectChanged

```

```

Public Event PerformAction As
blueoffice.menu.IFrameControl.PerformActionEventHandler Implements
blueoffice.menu.IFrameControl.PerformAction

Public Sub Action(Action As blueoffice.menu.FrameControlAction, _
Optional ActionValue As Object = Nothing) _
Implements blueoffice.menu.IFrameControl.Action

End Sub

Public Sub Clear() Implements blueoffice.menu.IFrameControl.Clear
'** Hier könne Sie Ihre Felder oder Listen Leeren
'** Diese Methode wird z.B. Aufgerufen wenn kein Datensatz
'** vorhanden ist
End Sub

Public Sub ControlClosing(ByRef Cancel As Boolean) _
Implements blueoffice.menu.IFrameControl.ControlClosing
'** Diese Methode wird vor dem Schliessen
'** (z.B. der Adressverwaltung) aufgerufen.
'** Sie können das schliessen mit: Cancel = True verhindern.
'** z.B. wenn nötige Eingaben noch fehlen
'** oder Ihre Daten noch nicht gespeichert sind.
End Sub

Public Sub InitializeControl() _
Implements blueoffice.menu.IFrameControl.InitializeControl
'** Hier können Sie Ihre Felder und Listen Initialisieren
'** Diese Methode wird beim starten
'** (z.B. der Adressverwaltung) 1x aufgerufen
End Sub

Public Sub Save(ByRef Cancel As Boolean) _
Implements blueoffice.menu.IFrameControl.Save
'** Diese Methode wird beim Speichern
'** (z.B. der Adresse) aufgerufen
'** Sie können hier falls nötig Ihre Daten speichern
'** oder mit: Cancel = True das Speichern abbrechen
'** wenn z.B. nötige Eingaben noch fehlen.
End Sub

Public Function GetControl() As System.Windows.Forms.Control _
Implements blueoffice.menu.IFrameControl.GetControl

'** Hier geben Sie blue office das AddIn zurück
'** Dies ist in der Regel 'Me'
'** (in diesem Beispiel für 'Adresse_Control')
'** könnte aber auch ein anderes sein
Return Me

End Function

Public Function GetSelected() As blueoffice.menu.SelectedObject _
Implements blueoffice.menu.IFrameControl.GetSelected
Return Nothing
End Function

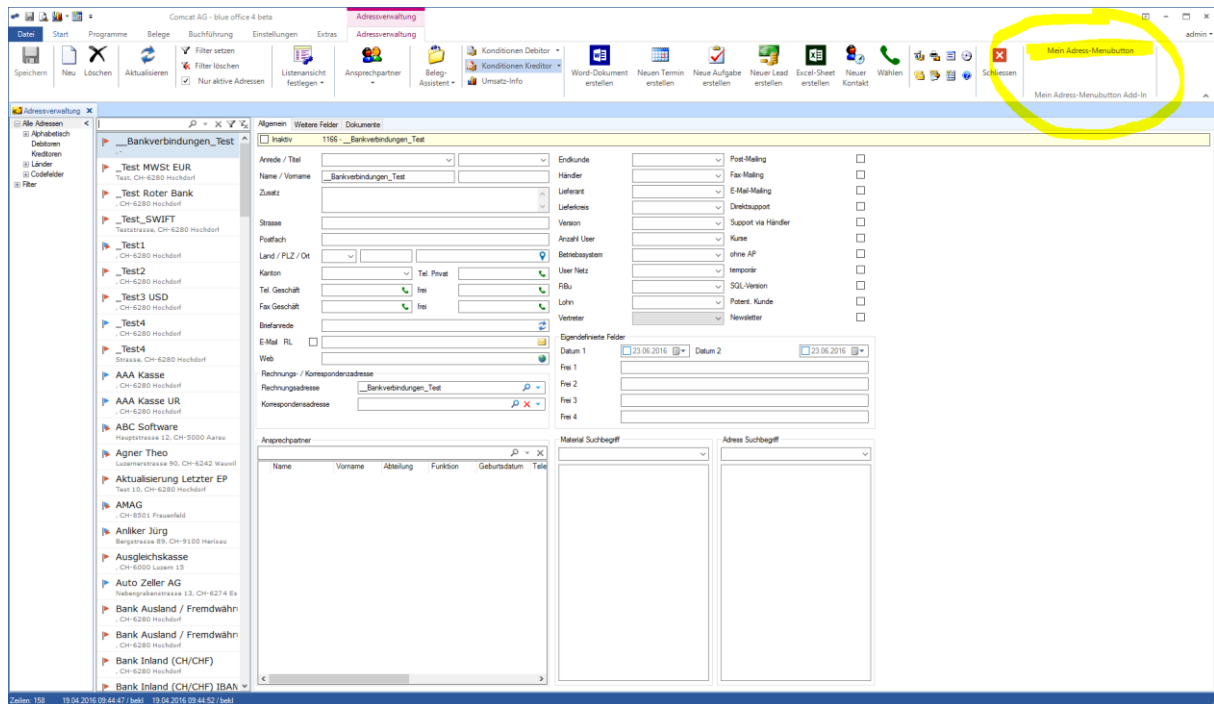
#End Region

End Class

```

## 2. Menüpunkt im Programmregister

Menüpunkt für Adressverwaltung, Artikelverwaltung, Projektverwaltung, Belegkopf usw.



## 2.1 Verweise

- ➔ [blueoffice.common](#) (blueoffice.common.dll)
- ➔ [blueoffice.menu](#) (blueoffice.menu.dll)
- ➔ [blueoffice.DAL](#) (blueoffice.DAL.dll)  
(DAL = **D**ata **A**ccess **L**ayer,  
Wird nur benötigt, wenn auf die aktuell gewählte  
Adresse, Artikel, usw. zugegriffen werden soll)

## 2.2 Klassen

- ➔ [AddIn.vb](#) (z.B.)  
Interfaces:
  - [blueoffice.common.AddIn.IAddInWithLicence](#)  
(Schnittstelle für AddIns mit individuellem oder universal-Lizenzschlüssel)  
oder:  
[blueoffice.common.AddIn.IAddIn](#)  
(Schnittstelle für AddIns ohne Lizenzschlüssel (Lizenzfile)  
→ Bitte diese Schnittstelle nur als Integrator verwenden, damit AddIns  
getestet werden können. Fertig erstellte Kunden AddIns haben  
üblicherweise einen Lizenzschlüssel, weshalb dafür die erst erwähnte  
Schnittstelle verwendet werden muss
- ➔ [AddInTabMenu.vb](#) (z.B.)  
Interfaces:
  - [blueoffice.common.AddIn.IAddInTabMenu](#)



## 2.3 Klasse: AddIn

```
Public Class AddIn

    Implements blueoffice.common.AddIn.IAddInWithLicence

    Public Function GetInfo() As blueoffice.common.AddIn.AddInInfo _
    Implements blueoffice.common.AddIn.IAddIn.GetInfo

        Dim RetVal As New blueoffice.common.AddIn.AddInInfo
        RetVal.CompanyName = "Ihr Firmenname AG"
        RetVal.Name = "Mein AddIn"

        RetVal.AddToList(GetType(AddInTabMenu), _
        blueoffice.common.AddIn.AddInShared.AddInDesitnation.ADR, _
        "Mein Adress-Menubutton 1", _
        "Zeigt (z.B.) eine MessageBox in der Adressverwaltung an")

        Return RetVal

    End Function

End Class
```

## 2.4 Klasse: AddInTabMenu

```
Public Class AddInTabMenu

    Implements blueoffice.common.AddIn.IAddInTabMenu

    Private _CurrentObject As blueoffice.DAL.A_ADRESSEN
    Private _CurrentID As Integer

    #Region "Interface AddInTabMenu"

    Public WriteOnly Property CurrentObject As Object _
    Implements blueoffice.common.AddIn.IAddInTabMenu.CurrentObject

        Set(value As Object)

            If value IsNot Nothing Then

                '** Hier erhalten Sie die Aktuell gewählte Adresse,
                '** Artikel, usw. als DAL-Objekt
                _CurrentID = 0

                If TypeOf (value) Is blueoffice.DAL.A_ADRESSEN Then

                    _CurrentObject = DirectCast(value, _
                                                blueoffice.DAL.A_ADRESSEN)

                    If _CurrentObject IsNot Nothing _
                        AndAlso _CurrentObject.HasData Then

                        _CurrentID = _CurrentObject.ADR_ID
                    End If
                End If
            End If

        End Set

    End Property

    Public Sub Load(ContextualMenu As blueoffice.menu.Contextual, _
        tabInfo As blueoffice.menu.TabInfo) _
    Implements blueoffice.common.AddIn.IAddInTabMenu.Load

        ContextualMenu.AddTool("IhrFirmennameAG.AdresseTool.Menubutton", _
            blueoffice.menu.MenuFrame.ToolType.Button, "", "", _
            "Mein Adress-Menubutton")

        Dim grp As blueoffice.menu.RibbonGroup
        grp = ContextualMenu.AddRibbonGroup(0, _
            "IhrFirmennameAG.AdresseGroup.Gruppe1", _
            "Mein Adress-Menubutton Add-In")

        grp.AddTool("IhrFirmennameAG.AdresseTool.Menubutton")

    End Sub

    Public Sub MenuBeforeDropdown(sender As Object, Key As String, _
        MenuTool As blueoffice.menu.ToolBase, ByRef Cancel As Boolean) _
    Implements blueoffice.common.AddIn.IAddInTabMenu.MenuBeforeDropdown

    End Sub
```

```

Public Sub MenuClick(Sender As Object, Key As String, _
MenuTool As blueoffice.menu.ToolBase, ByRef IsHandled As Boolean) _
Implements blueoffice.common.AddIn.IAddInTabMenu.MenuClick

    If Key = "IhrFirmennameAG.AdresseTool.Menubutton" Then

        MsgBox(_CurrentObject.ADR_Sortierung, _
            vbInformation + vbOKOnly, "")

        IsHandled = True
    End If

End Sub

Public Sub Unload() _
Implements blueoffice.common.AddIn.IAddInTabMenu.Unload

End Sub

Public Function Description() As String _
Implements blueoffice.common.AddIn.IAddInTabMenu.Description
    '** Beschreibung des AddIns
    Return "Zeigt (z.B.) eine Messagebox in der Adressverwaltung an
(AddInTabMenu)"
End Function

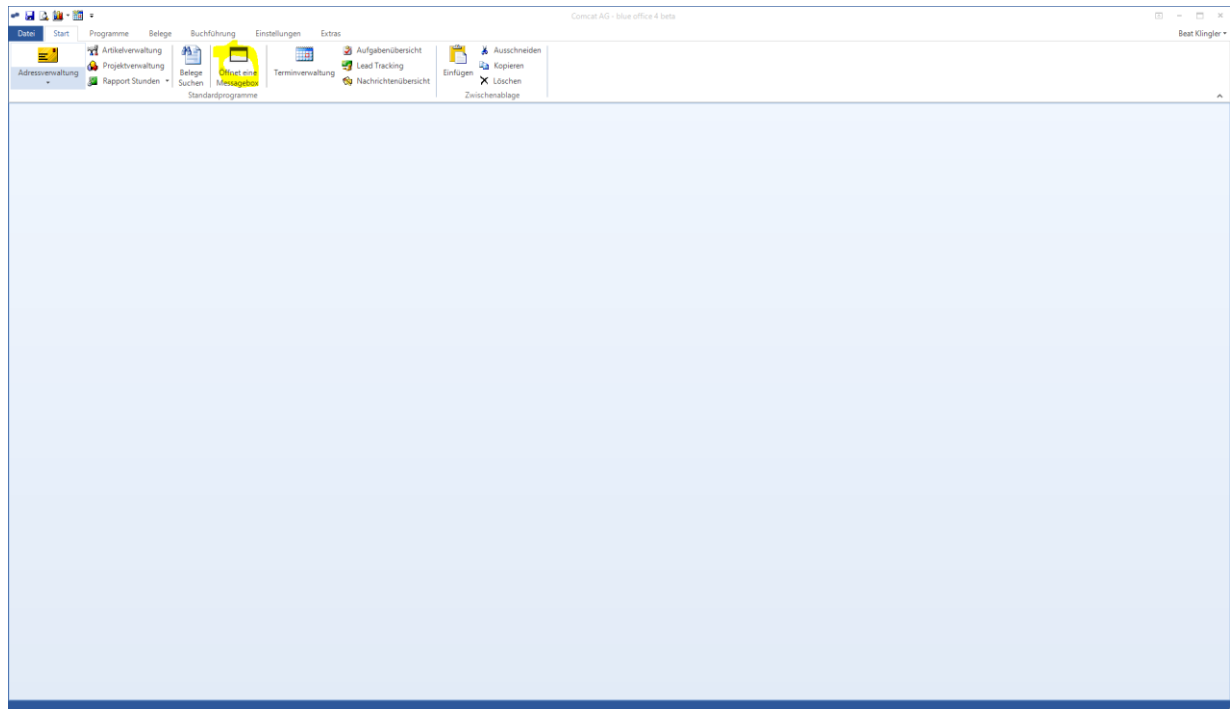
Public Function Name() As String _
Implements blueoffice.common.AddIn.IAddInTabMenu.Name
    '** Name des AddIns
    Return "Mein Adress-Menubutton (AddInTabMenu)"
End Function

#End Region

End Class

```

### 3. Menupunkt in der Applikation zum Ausführen einer Funktion



### 3.1 Verweise

- ➔ [blueoffice.common](#) (blueoffice.common.dll)
- ➔ [blueoffice.menu](#) (blueoffice.menu.dll)

### 3.2 Klassen

- ➔ [AddIn.vb](#) (z.B.)

Interfaces:

- [blueoffice.common.AddIn.IAddInWithLicence](#)  
(Schnittstelle für AddIns mit individuellem oder universal-Lizenzschlüssel)  
oder:  
[blueoffice.common.AddIn.IAddIn](#)  
(Schnittstelle für AddIns ohne Lizenzschlüssel (Lizenzfile)  
➔ Bitte diese Schnittstelle nur als Integrator verwenden, damit AddIns  
getestet werden können. Fertig erstellte Kunden AddIns haben  
üblicherweise einen Lizenzschlüssel, weshalb dafür die erst erwähnte  
Schnittstelle verwendet werden muss

- ➔ [AddInMenu.vb](#) (z.B.)

Interfaces:

- [blueoffice.common.AddIn.IAddInMenu](#)

### 3.3 Klasse: AddIn

```
Public Class AddIn

    Implements blueoffice.common.AddIn.IAddInWithLicence

    Public Function GetInfo() As blueoffice.common.AddIn.AddInInfo _
    Implements blueoffice.common.AddIn.IAddIn.GetInfo

        Dim RetVal As New blueoffice.common.AddIn.AddInInfo
        RetVal.CompanyName = "Ihr Firmenname AG"
        RetVal.Name = "Mein AddIn"

        RetVal.AddToList(GetType(AddInMenu), _
            "Mein Applikation-Menubutton", _
            "Zeigt (z.B.) eine MessageBox in der Applikation an")

        Return RetVal

    End Function

End Class
```

### 3.4 Klasse: AddInMenu

```
Public Class AddInMenu

    Implements blueoffice.common.AddIn.IAddInMenu

    #Region "Interface AddInMenu"

    Public Sub Load(MainMenu As blueoffice.menu.MenuFrame) _
        Implements blueoffice.common.AddIn.IAddInMenu.Load

        MainMenu.RootTools.AddTool( _
            "IhrFirmennameAG.ApplikationTool.Menubutton1", _
            blueoffice.menu.MenuFrame.ToolType.Button, "EXE", "EXE", _
            "Öffnet eine Messagebox", "")

        With MainMenu.GetRibbonGroup(0, 0)
            With .AddTool("IhrFirmennameAG.ApplikationTool.Menubutton1", 5)
                .InstanceProps.IsFirstInGroup = True
                .InstanceProps.PreferredSizeOnRibbon =
blueoffice.menu.MenuFrame.ToolSize.Large
            End With
        End With

    End Sub

    Public Sub MenuBeforeDropdown(sender As Object, Key As String, _
        MenuTool As blueoffice.menu.ToolBase, ByRef Cancel As Boolean) _
        Implements blueoffice.common.AddIn.IAddInMenu.MenuBeforeDropdown

    End Sub

    Public Sub MenuClick(Sender As Object, Key As String, _
        MenuTool As blueoffice.menu.ToolBase, ByRef IsHandled As Boolean) _
        Implements blueoffice.common.AddIn.IAddInMenu.MenuClick
        If Key = "IhrFirmennameAG.ApplikationTool.Menubutton1" Then
            MsgBox("Dies ist die Messagebox", vbInformation + vbOKOnly, "")
            IsHandled = True
        End If
    End Sub

    Public Sub Unload() Implements blueoffice.common.AddIn.IAddInMenu.Unload

    End Sub

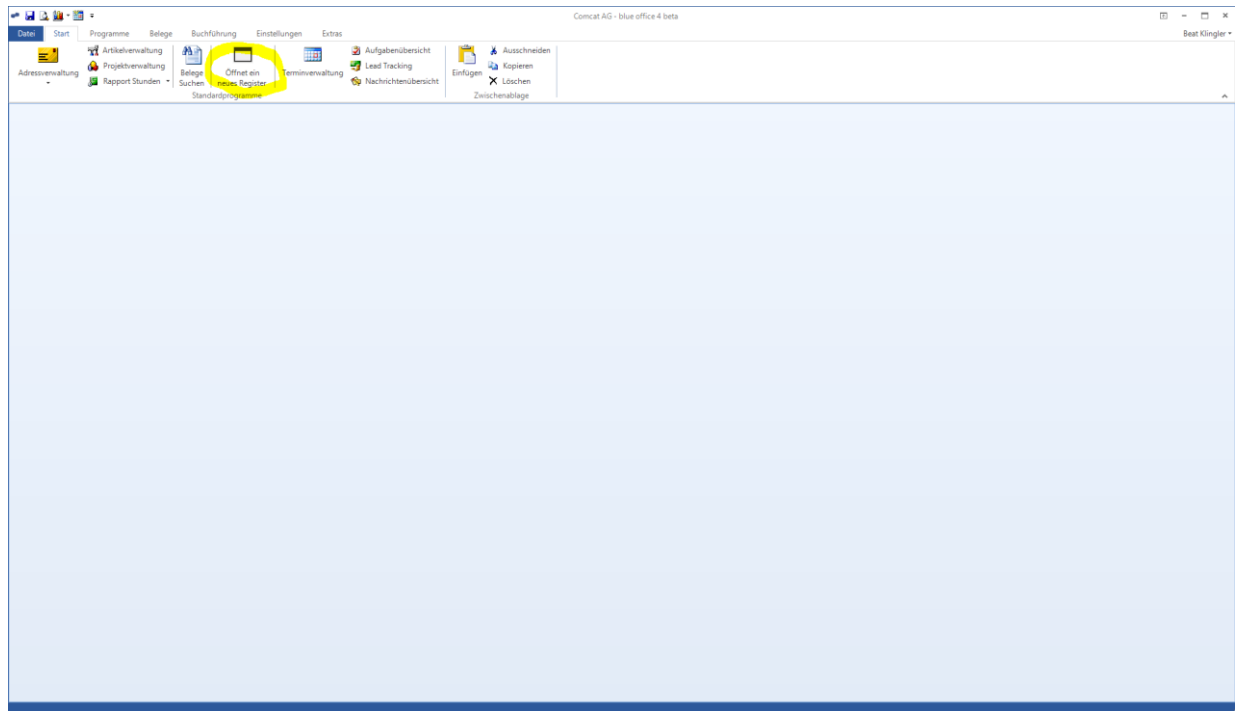
    Public Function Description() As String _
        Implements blueoffice.common.AddIn.IAddInMenu.Description
        ''' Beschreibung des AddIns
        Return "Zeigt (z.B.) eine Messagebox in der Applikation an
(AddInMenu)"
    End Function

    Public Function Name() As String _
        Implements blueoffice.common.AddIn.IAddInMenu.Name
        ''' Name des AddIns
        Return "Mein Applikation-Menubutton 1 (AddInMenu)"
    End Function

    #End Region

End Class
```

#### 4. Menüpunkt in der Applikation zum Öffnen eines neuen Registers





#### 4.1 Verweise

- ➔ [blueoffice.common](#) (blueoffice.common.dll)
- ➔ [blueoffice.menu](#) (blueoffice.menu.dll)

#### 4.2 Klassen

- ➔ [AddIn.vb](#) (z.B.)

Interfaces:

- [blueoffice.common.AddIn.IAddInWithLicence](#)  
(Schnittstelle für AddIns mit individuellem oder universal-Lizenzschlüssel)  
oder:  
[blueoffice.common.AddIn.IAddIn](#)  
(Schnittstelle für AddIns ohne Lizenzschlüssel (Lizenzfile)  
➔ Bitte diese Schnittstelle nur als Integrator verwenden, damit AddIns getestet werden können. Fertig erstellte Kunden AddIns haben üblicherweise einen Lizenzschlüssel, weshalb dafür die erst erwähnte Schnittstelle verwendet werden muss

- ➔ [AddInMenu.vb](#) (z.B.)

Interfaces:

- [blueoffice.common.AddIn.IAddInMenu](#)

#### 4.3 Benutzersteuerelemente

- ➔ [Applikation\\_Control.vb](#) (z.B.)

Interfaces :

- [blueoffice.menu. IFrameTabMenu](#)

#### 4.4 Klasse: AddIn

```
Public Class AddIn

    Implements blueoffice.common.AddIn.IAddInWithLicence

    Public Function GetInfo() As blueoffice.common.AddIn.AddInInfo _
    Implements blueoffice.common.AddIn.IAddIn.GetInfo

        Dim RetVal As New blueoffice.common.AddIn.AddInInfo
        RetVal.CompanyName = "Ihr Firmenname AG"
        RetVal.Name = "Mein AddIn"

        RetVal.AddToList(GetType(AddInMenu), _
            "Mein Applikation-Menubutton 2", _
            "Zeigt (z.B.) ein neues Register in der Applikation an")

        Return RetVal

    End Function

End Class
```

## 4.5 Klasse: AddInMenu

```
Public Class AddInMenu

    Implements blueoffice.common.AddIn.IAddInMenu

    #Region "Interface AddInMenu"

    Public Sub Load(MainMenu As blueoffice.menu.MenuFrame) _
        Implements blueoffice.common.AddIn.IAddInMenu.Load

        MainMenu.RootTools.AddTool( _
            "IhrFirmennameAG.ApplikationTool.Menubutton2", _
            blueoffice.menu.MenuFrame.ToolType.Button, "EXE", "EXE", _
            " Öffnet ein neues Register", "")

        With MainMenu.GetRibbonGroup(0, 0)
            With .AddTool("IhrFirmennameAG.ApplikationTool.Menubutton2", 5)
                .InstanceProps.IsFirstInGroup = True
                .InstanceProps.PreferredSizeOnRibbon =
blueoffice.menu.MenuFrame.ToolSize.Large
            End With
        End With

    End Sub

    Public Sub MenuBeforeDropdown(sender As Object, Key As String, _
        MenuTool As blueoffice.menu.ToolBase, ByRef Cancel As Boolean) _
        Implements blueoffice.common.AddIn.IAddInMenu.MenuBeforeDropdown

    End Sub

    Public Sub MenuClick(Sender As Object, Key As String, _
        MenuTool As blueoffice.menu.ToolBase, ByRef IsHandled As Boolean) _
        Implements blueoffice.common.AddIn.IAddInMenu.MenuClick
        If Key = "IhrFirmennameAG.ApplikationTool.Menubutton2" Then

            If blueoffice.menu.AppFrameShared.IsTabOpen( _
                "IhrFirmennameAG.ApplikationControl", True) = False Then

                Dim tCtl As blueoffice.menu.IFrameTabMenu = New
Applikation_Control

                blueoffice.menu.AppFrameShared.OpenTab( _
                    "IhrFirmennameAG.ApplikationControl", _
                    "Neues Register", "", "EXE", tCtl)

            End If

            IsHandled = True

        End If
    End Sub

    Public Sub Unload() Implements blueoffice.common.AddIn.IAddInMenu.Unload

    End Sub

End Class
```

```
Public Function Description() As String _
Implements blueoffice.common.AddIn.IAddInMenu.Description
    '** Beschreibung des AddIns
    Return "Zeigt (z.B.) ein neues Register in der Applikation an
(AddInMenu)"
End Function

Public Function Name() As String _
Implements blueoffice.common.AddIn.IAddInMenu.Name
    '** Name des AddIns
    Return "Mein Applikation-Menubutton 2 (AddInMenu)"
End Function

#End Region

End Class
```

## 4.6 Benutzersteuerelement: Applikation\_Control

```
Public Class Applikation_Control

    Implements blueoffice.menu.IFrameTabMenu

    Dim _TabInfo As blueoffice.menu.TabInfo

    #Region "Interface"

    Public Property ContextualMenu As blueoffice.menu.Contextual _
    Implements blueoffice.menu.IFrameTabMenu.ContextualMenu

    Public Property MainMenu As blueoffice.menu.MenuFrame _
    Implements blueoffice.menu.IFrameTabMenu.MainMenu

    Public Sub Action(Action As blueoffice.menu.FrameControlAction, _
    Optional ActionValue As Object = Nothing) _
    Implements blueoffice.menu.IFrameTab.Action

    End Sub

    Public Sub InitializeContextualMenu() _
    Implements blueoffice.menu.IFrameTabMenu.InitializeContextualMenu

        '** Hier wird die Toolbar des neuen Registers erstellt

        ContextualMenu.CreateEmptyContextualMenu("Neues Register")
        ContextualMenu.AddRibbonTab("T1", "Neues Register")

        ContextualMenu.RootTools.AddTool( _
        "IhrFirmennameAG.ApplikationTool.Menubutton3", _
        blueoffice.menu.MenuFrame.ToolType.Button, "EXE", "EXE", _
        "Öffnet eine Messagebox", "")

        With ContextualMenu.AddRibbonGroup("T1", "G1", "")
            .PreferredToolSize = blueoffice.menu.MenuFrame.ToolSize.Large
            .AddTool("IhrFirmennameAG.ApplikationTool.Menubutton3")
        End With

        With ContextualMenu.AddRibbonGroup("T1", "G2", "")
            .PreferredToolSize = blueoffice.menu.MenuFrame.ToolSize.Large
            .AddTool("mnuD_Schliessen")
        End With

    End Sub

    Public Sub MenuBeforeDropdown(sender As Object, Key As String, _
    MenuTool As blueoffice.menu.ToolBase, ByRef Cancel As Boolean) _
    Implements blueoffice.menu.IFrameTabMenu.MenuBeforeDropdown

    End Sub
```

```

Public Sub MenuClick(Sender As Object, Key As String, _
MenuTool As blueoffice.menu.ToolBase, ByRef IsHandled As Boolean) _
Implements blueoffice.menu.IFrameTabMenu.MenuClick

    If Key = "IhrFirmennameAG.ApplikationTool.Menubutton3" Then
        MsgBox("Dies ist die Messagebox", vbInformation + vbOKOnly, "")
        IsHandled = True
    End If

End Sub

Public Sub TabActivated() _
Implements blueoffice.menu.IFrameTab.TabActivated
    '** Hier wird die Toolbar des neuen Registers angezeigt
    ContextualMenu.ShowTab()
End Sub

Public Sub TabClosed() Implements blueoffice.menu.IFrameTab.TabClosed
    '** Diese Methode wird nach dem Schliessen des Registers aufgerufen.
End Sub

Public Sub TabClosing(ByRef Cancel As Boolean) _
Implements blueoffice.menu.IFrameTab.TabClosing

    '** Diese Methode wird vor dem Schliessen des Registers aufgerufen.
    '** Sie können das schliessen mit: Cancel = True verhindern.
    '** z.B. wenn nötige Eingaben noch fehlen
    '** oder Ihre Daten noch nicht gespeichert sind.

End Sub

Public Sub TabLoad(tabInfo As blueoffice.menu.TabInfo) _
Implements blueoffice.menu.IFrameTab.TabLoad
    _TabInfo = tabInfo
End Sub

Public Sub TabShow() Implements blueoffice.menu.IFrameTab.TabShow
    Me.Refresh()
End Sub

Public Function GetSelected() As blueoffice.menu.SelectedObject _
Implements blueoffice.menu.IFrameTab.GetSelected
    Return Nothing
End Function

Public Function GetTabControl() As System.Windows.Forms.Control _
Implements blueoffice.menu.IFrameTab.GetTabControl
    '** Hier geben Sie blue office das AddIn zurück
    '** Dies ist in der Regel 'Me' (in diesem Beispiel für
    '** 'Application_Control') könnte aber auch ein anderes sein.
    Return Me
End Function

Public Function TabInfo() As blueoffice.menu.TabInfo _
Implements blueoffice.menu.IFrameTab.TabInfo
    Return _TabInfo
End Function

#End Region

End Class

```