

## Creating Smart Tag for Components

---

نحوه ساختن

# Smart Tag

برای کامپوننت های کاربر

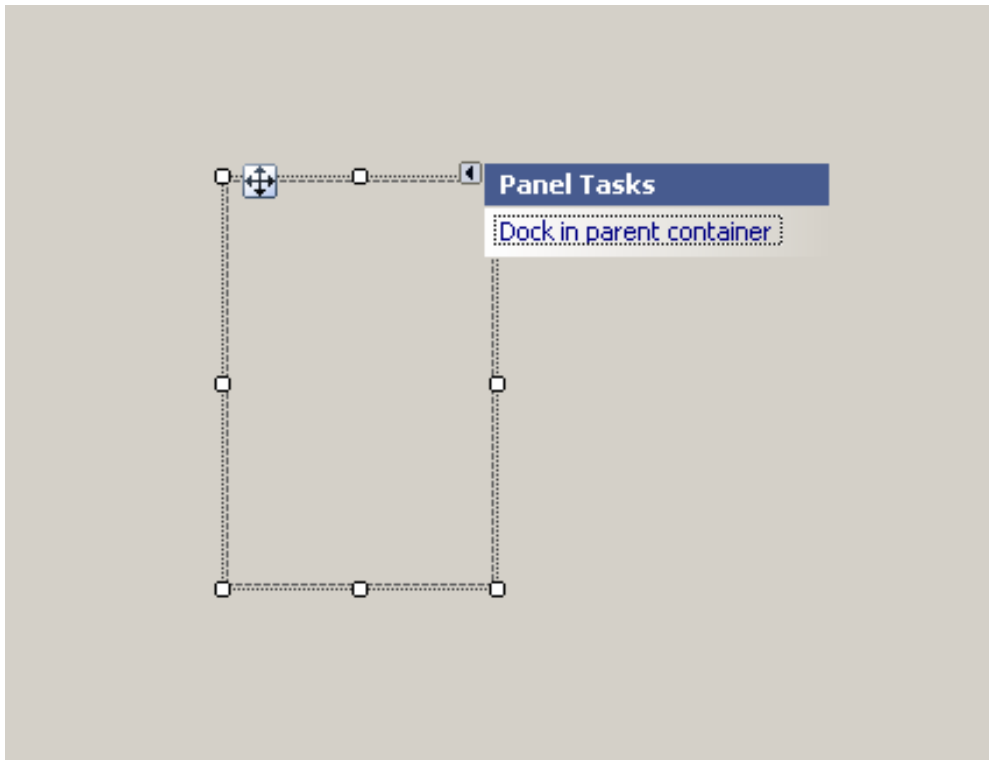
در

**وی بی دات نت**

---

## Smart Tag چیست؟

Smart Tag ها منو هایی هستند که در هنگام طراحی محیط کاربری برنامه روی بعضی از کنترل ها ظاهر می شوند. این گزینه های منویی به برنامه نویس نهایی کمک می کنند تا راحت تر با این کنترل ها کار کند. در واقع Smart Tag ها جهت سهل استفاده کردن یک کامپوننت<sup>1</sup> ساخته شده اند. در شکل روبرو نمایی از Smart Tag مربوط به یک کنترل Panel را مشاهده می فرمایید. این گزینه منویی به برنامه نویس کمک می کند تا بدون مراجعه به پنجره Properties و تغییر در خصوصیت Dock و فقط با کلیک گزینه Dock to parent container<sup>2</sup> خصوصیت Dock کنترل Panel را به All ست کند. این کار باعث می شود تا کنترل تمامی سطح والد<sup>2</sup> خود را بپوشاند. معمولاً برنامه نویس اولیه خصوصیت پر کاربرد یک کنترل یک اعمال هنگام طراحی<sup>3</sup> را در Smart Tag ها قرار می دهد تا دسترسی بدان ها ساده تر صورت بگیرد. در این شکل "Dock to parent container" یک عمل هنگام طراحی است که شرح داده شد. اعمال هنگام طراحی به طور خودکار یک یا چند خصوصیت را به مقداری ست می کنند. چنانچه شما یک برنامه نویس ثانویه<sup>4</sup> هستید به منظور توسعه کامپوننت سهل استفاده و حرفه ای لاطم است تا در مورد رفتار های زمان طراحی نیز تمهیداتی ببینید. به عنوان نمونه نصب Smart Tag روی کنترل از این مثال است. در ادامه نحوه انجام این کار در فضای توسعه دات نت نسخه ۲.۰ (سینتکس VB) توضیح داده خواهد شد.



شکل شماره ۱

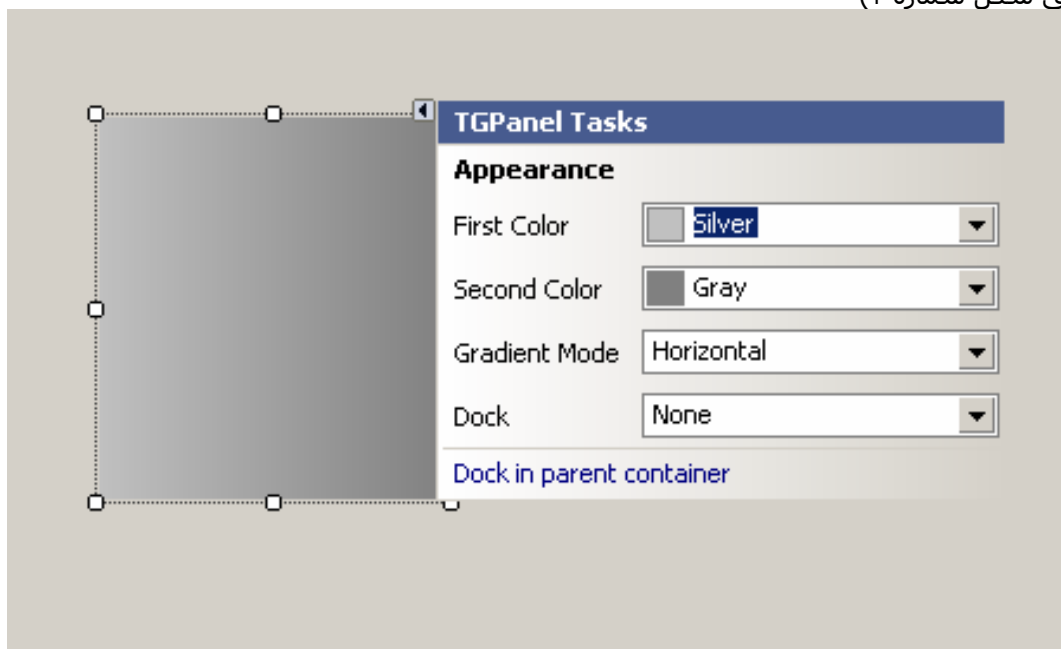
## نحوه سازماندهی مقاله

در این مقاله سعی شده است طبق روش قدم زدن میکروسافت مراحل انجام این کار با یک مثال ساده همراه شود. ما یک کنترل ساده برای ایجاد حالت محو تدریجی یک رنگ و ورود به زنگ دیگر نوشته ایم. این کنترل شامل سه خصوصیت زیر است:

- 
- Component<sup>1</sup>
  - Parent<sup>2</sup>
  - Design-time Methods<sup>3</sup>
  - برنامه نویس کامپوننت<sup>4</sup>

نام خصوصیت	نوع	توضیحات
Color۱	Color	رنگ اول را مشخص می کند
Color۲	Color	رنگ دوم را مشخص می کند
GradientMode	LinearGradientMode	جهت ترسیم را مشخص می کند

ما سعی داریم این سه خصوصیت و خصوصیت Dock را به منوی زمان طراحی کنترل اضافه کنیم. (مطابق شکل شماره ۲)



شکل شماره ۲

ابتدا یک کلاس جدید به پروژه اضافه کنید. سپس کد های مربوط به کنترل مورد نظر (من در این مثال نام کنترل را TGPLPanel نهاده ام) در آن درج کنید. کد های کنترل من به این شکل هستند:

```
Public Class TGPLPanel
    Inherits System.Windows.Forms.Panel

    Dim _Color1 As Color = Color.Silver
    Dim _Color2 As Color = Color.Gray
    Dim _GradientMode As LinearGradientMode =
        LinearGradientMode.Horizontal

    Private Sub TGPLPanel_Resize(ByVal sender As Object, ByVal e As
        System.EventArgs) Handles Me.Resize
        Call PaintMe()
    End Sub

    <System.Diagnostics.DebuggerStepThrough()> _
    Private Sub PaintMe()
        Dim g As Graphics
        Dim bmp As Bitmap
        Dim rc As Rectangle
        Dim brsh As Brush

        If _GradientMode = LinearGradientMode.Vertical Then
            rc = New Rectangle(0, 0, 1, Me.Height)
            bmp = New Bitmap(1, Me.Height)
        Else
```

```

        rc = New Rectangle(0, 0, Me.Width, 1)
        bmp = New Bitmap(Me.Width, 1)
    End If

    brsh = New Drawing2D.LinearGradientBrush(rc, _Color1, _Color2,
    _GradientMode)

    g = Graphics.FromImage(bmp)
    g.FillRectangle(brsh, rc)
    Me.BackgroundImage = bmp

End Sub

Public Property Color1() As Color
    Get
        Return _Color1
    End Get
    Set(ByVal value As Color)
        _Color1 = value
        Call PaintMe()
    End Set
End Property

Public Property Color2() As Color
    Get
        Return _Color2
    End Get
    Set(ByVal value As Color)
        _Color2 = value
        Call PaintMe()
    End Set
End Property

Public Property GradientMode() As LinearGradientMode
    Get
        Return _GradientMode
    End Get
    Set(ByVal value As LinearGradientMode)
        _GradientMode = value
        Call PaintMe()
    End Set
End Property

End Class

```

قبل از ادامه کار فضاهای نامی زیر را در محیط کد وارد کنید

```

Imports System
Imports System.Text
Imports System.Drawing
Imports System.ComponentModel
Imports System.Drawing.Drawing2D
Imports System.ComponentModel.Design

```

تا بدین جا کنترل ما کامل شده و کار می کند اما ما هنوز به هدف خود نرسیده ایم. بنابراین باید اعمال دیگری را نیز صورت دهیم.

**توجه :** برای ادامه کار لازم است تا مرجع System.Design به پروژه اضافه شوند. بدین منظور روی منوی Project کلیک کنید و Add Refrence را انتخاب کنید. سپس مرجع یاد شده را به پروژه اضافه کنید.

در همان فضای اسمی<sup>۵</sup> که کنترل را ساخته اید یک کلاس دیگر که از System.ComponentModel.Design.DesignerActionList ارث بری داشته باشد ایجاد کنید. من نام این کلاس را TGPANELActionList نهاده ام. متغیرهای زیر را به کلاس اضافه کنید:

```
Dim _TGPANEL As TGPANEL
```

```
Dim _DesignerActionUISvc As DesignerActionUIService = Nothing
```

متغیر اول از نوع کنترل ایجاد شده ما و دومی از نوع DesignerActionUIService خواهد بود. در ادامه کاربرد این متغیرها را خواهید دید. متد سازنده ای مانند زیر به این کلاس اضافه کنید:

```
Public Sub New(ByVal _Component As IComponent)
```

```
    MyBase.New(_Component)  
    Me._TGPANEL = _Component
```

```
    Me._DesignerActionUISvc =  
    CType(GetService(GetType(DesignerActionUIService)),  
    DesignerActionUIService)
```

```
End Sub
```

**\_Component** : کامپوننت انتخاب شده در محیط طراحی ویژوال استودیو است. که مقدار آن را در متغیری از نوع کامپوننت مورد نظر که قبلاً ایجاد کرده ایم (**\_TGPANEL**) میریزیم. متغیر **\_DesignerActionUISvc** با مقدار سرویس **DesignerActionUIService** (سرویس منوهای زمان طراحی ویژوال استودیو) پر می شود.

در قدم بعدی روتین زیر را به کلاس اضافه کنید:

```
Private Function GetPropertyByName(ByVal _PropName As String) As  
PropertyDescriptor  
    Dim _Prop As PropertyDescriptor  
    _Prop = TypeDescriptor.GetProperties(_TGPANEL)(_PropName)  
    If _Prop Is Nothing Then  
        Throw New ArgumentException("Property not found!",  
_PropName)  
    Else  
        Return _Prop  
    End If  
End Function
```

این روتین جهت دریافت خصوصیت یک شی در زمان تولید کاربرد دارد. تمام خصوصیت های مورد نظر را که باید در زمان طراحی نمایش داده شوند را به صورت خصوصیت های کلاس ایجاد کنید. با این تفاوت که در هنگام مقدار دهی باید از روتین بالا برای مقدار دهی استفاده شود. مثال:

```
Public Property Color1() As Color  
    Get  
        Return _TGPANEL.Color1  
    End Get  
    Set(ByVal value As Color)  
        GetPropertyByName("Color1").SetValue(_TGPANEL, value)  
    End Set  
End Property
```

در مرحله بعد روتین زیر باید به کلاس اضافه شود:

```
Public Overrides Function GetSortedActionItems() As
DesignerActionItemCollection
    Dim _Items As New DesignerActionItemCollection()

    _Items.Add(New DesignerActionHeaderItem("Appearance"))

    _Items.Add(New DesignerActionPropertyItem("Color1", "First
Color", "Appearance", "Set the first color.))
    _Items.Add(New DesignerActionPropertyItem("Color2", "Second
Color", "Appearance", "Set the second color.))
    _Items.Add(New DesignerActionPropertyItem("GradientMode",
"Gradient Mode", "Appearance", "Set the Gradient Mode"))
    _Items.Add(New DesignerActionPropertyItem("Dock", "Dock",
"Appearance", "Set the Control dock property"))

    Return _Items

    Me._DesignerActionUISvc =
CType(GetService(GetType(DesignerActionUIService)),
DesignerActionUIService)
```

End Function

این تابع از کلاس سازنده (DesignerActionList) به این کلاس به ارث رسیده است. لذا باید به صورت Overrides تعریف گردد.

این تابع لیست آیتم های منوی Smart Tag را بر می گرداند. مقدار برگشتی این تابع از نوع DesignerActionItemCollection است. این نوع بیانگر کلکسیون آیتم های منوی هنگام طراحی است.

توضیحات:

خط زیر یک گروه به لیست منو ها اضافه می کند

```
_Items.Add(New DesignerActionHeaderItem("Appearance"))
```

خط زیر یک آیتم از منو با نام First Color که به خصوصیت Color1 مپ شده و در گروه Appearance

قرار دارد را ایجاد می کند

```
_Items.Add(New DesignerActionPropertyItem("Color1", "First Color",
"Appearance", "Set the first color.))
```

این خط برای مقدار دهی مجدد DesignerActionUISvc نوشته شده است

```
Me._DesignerActionUISvc =
CType(GetService(GetType(DesignerActionUIService)),
DesignerActionUIService)
```

کار این کلاس تمام شده اما ما هنوز به هدف خود رسیده ایم.

در همان فضای اسمی یک کلاس دیگر با کلاس System.Windows.Forms.Design.ControlDesigner ارث بری داشته باید ایجاد کنید.

این کلاس قرار است لیست منو ها را برگشت دهد. بدین منظور یک متغیر به صورت زیر تعریف

کنید:

```
Private _Lists As DesignerActionListCollection
```

این کلاس یک تابع موروثی با نام دارد که ما از آن برای برگشت دادن لیست منو ها استفاده می

کنیم. بدین صورت آن را کد نویسی کنید.

```
Public Overrides ReadOnly Property ActionLists() As
DesignerActionListCollection
    Get
        If _Lists Is Nothing Then
            _Lists = New DesignerActionListCollection()
            _Lists.Add(New TGPanelActionList(Me.Component))
        End If
        Return _Lists
```

```
End Get
End Property
```

من نام این کلاس را TGPaneldesigner گذاشته ام. کار ما تقریباً تمام شده است. خط زیر را که معرف یک کد پیش اجرا است به قبل از عنوان کلاس فوق اضافه کنید

```
<System.Security.Permissions.PermissionSetAttribute(System.Security.P
ermissons.SecurityAction.Demand, Name="FullTrust")> _
```

به خط قبل از کلاس کامپوننت هم این خط را بیفزایید:

```
<Designer(GetType(TGPaneldesigner))> _
```

کار ما تمام شد. اگر کارتان را درست انجام داده باشید می توانید شکل ۲ را در پروژه خود ببینید.

## خلاصه کار

اگر حوصله تعقیب کردن مراحل را نداشتید کارهای زیر را انجام دهید تا نتیجه کار را خودتان ببینید.

- یک سند کد جدید به پروژه خود اضافه کنید.
- مرجع System.Design را به پروژه اضافه کنید
- کد زیر را در سند کد درج کنید و پروژه را بسازید
- سپس کامپوننت TGPaneldesigner را روی فرم قرار دهید و Smart Tag منوی آن را ببینید

کد:

```
Imports System
Imports System.Text
Imports System.Drawing
Imports System.ComponentModel
Imports System.Drawing.Drawing2D
Imports System.ComponentModel.Design
```

```
<Designer(GetType(TGPaneldesigner))> _
Public Class TGPaneldesigner
    Inherits System.Windows.Forms.Panel
```

```
    Dim _Color1 As Color = Color.Silver
    Dim _Color2 As Color = Color.Gray
    Dim _GradientMode As LinearGradientMode =
LinearGradientMode.Horizontal
```

```
    Private Sub TGPaneldesigner_Resize(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Resize
        Call PaintMe()
    End Sub
```

```
<System.Diagnostics.DebuggerStepThrough()> _
Private Sub PaintMe()
    Dim g As Graphics
    Dim bmp As Bitmap
    Dim rc As Rectangle
    Dim brsh As Brush
```

```

    If _GradientMode = LinearGradientMode.Vertical Then
        rc = New Rectangle(0, 0, 1, Me.Height)
        bmp = New Bitmap(1, Me.Height)
    Else
        rc = New Rectangle(0, 0, Me.Width, 1)
        bmp = New Bitmap(Me.Width, 1)
    End If

    brsh = New Drawing2D.LinearGradientBrush(rc, _Color1, _Color2,
    _GradientMode)

    g = Graphics.FromImage(bmp)
    g.FillRectangle(brsh, rc)
    Me.BackgroundImage = bmp

End Sub

Public Property Color1() As Color
    Get
        Return _Color1
    End Get
    Set(ByVal value As Color)
        _Color1 = value
        Call PaintMe()
    End Set
End Property

Public Property Color2() As Color
    Get
        Return _Color2
    End Get
    Set(ByVal value As Color)
        _Color2 = value
        Call PaintMe()
    End Set
End Property

Public Property GradientMode() As LinearGradientMode
    Get
        Return _GradientMode
    End Get
    Set(ByVal value As LinearGradientMode)
        _GradientMode = value
        Call PaintMe()
    End Set
End Property

End Class

<System.Security.Permissions.PermissionSetAttribute(System.Security.P
ermissions.SecurityAction.Demand, Name:"FullTrust")> _
Public Class TGPanelDesigner
    Inherits System.Windows.Forms.Design.ControlDesigner

    Private _Lists As DesignerActionListCollection
    '

    Public Overrides ReadOnly Property ActionLists() As
DesignerActionListCollection
        Get

```

```

        If _Lists Is Nothing Then
            _Lists = New DesignerActionListCollection()
            _Lists.Add(New TGPanelActionList(Me.Component))
        End If
        Return _Lists
    End Get
End Property
End Class

Public Class TGPanelActionList
    Inherits System.ComponentModel.Design.DesignerActionList

    Dim _TGPanel As TGPanel

    Dim _DesignerActionUISvc As DesignerActionUIService = Nothing
    '

    Public Sub New(ByVal _Component As IComponent)

        MyBase.New(_Component)
        Me._TGPanel = _Component

        Me._DesignerActionUISvc =
        CType(GetService(GetType(DesignerActionUIService)),
        DesignerActionUIService)

    End Sub

    Private Function GetPropertyByName(ByVal _PropName As String) As
PropertyDescriptor
        Dim _Prop As PropertyDescriptor
        _Prop = TypeDescriptor.GetProperties(_TGPanel)(_PropName)
        If _Prop Is Nothing Then
            Throw New ArgumentException("Property not found!",
            _PropName)
        Else
            Return _Prop
        End If
    End Function

    Public Property Color1() As Color
        Get
            Return _TGPanel.Color1
        End Get
        Set(ByVal value As Color)
            GetPropertyByName("Color1").SetValue(_TGPanel, value)
        End Set
    End Property

    Public Property Color2() As Color
        Get
            Return _TGPanel.Color2
        End Get
        Set(ByVal value As Color)
            GetPropertyByName("Color2").SetValue(_TGPanel, value)
        End Set
    End Property

    Public Property GradientMode() As LinearGradientMode
        Get
            Return _TGPanel.GradientMode

```

```

        End Get
        Set(ByVal value As LinearGradientMode)
            GetPropertyByName("GradientMode").SetValue(_TGPanel, value)
        End Set
    End Property

    Public Property Dock() As DockStyle
        Get
            Return _TGPanel.Dock
        End Get
        Set(ByVal value As DockStyle)
            GetPropertyByName("Dock").SetValue(_TGPanel, value)
        End Set
    End Property

    Public Overrides Function GetSortedActionItems() As
    DesignerActionItemCollection
        Dim _Items As New DesignerActionItemCollection()

        _Items.Add(New DesignerActionHeaderItem("Appearance"))

        _Items.Add(New DesignerActionPropertyItem("Color1", "First
        Color", "Appearance", "Set the first color.))
        _Items.Add(New DesignerActionPropertyItem("Color2", "Second
        Color", "Appearance", "Set the second color.))
        _Items.Add(New DesignerActionPropertyItem("GradientMode",
        "Gradient Mode", "Appearance", "Set the Gradient Mode"))
        _Items.Add(New DesignerActionPropertyItem("Dock", "Dock",
        "Appearance", "Set the Control dock property"))

        Return _Items

        Me._DesignerActionUISvc =
    CType(GetService(GetType(DesignerActionUIService)),
    DesignerActionUIService)

    End Function
End Class

```

نوشته شده توسط:

<http://BlackDal.BlogFa.com>  
<mailto:Hamed.Masafi@GMail.com>

حامد مصافی