# Creating an Autodesk Inventor Add-in using Visual Studio Vb.net 2022 Part 02.

Get the Inventor application from the Addin.

First, we need to get the Inventor Application from the Addin to communicate with.

```
Imports System.Runtime.InteropServices
Imports Inventor
<progIdAttribute("MyInventorAddIn.StandardAddInServer"),</pre>
GuidAttribute("ad0a3353-0c3b-42a6-87ba-e1a86f4529e7")>
Public Class StandardAddInServer
    Implements Inventor.ApplicationAddInServer
    Private m_inventorApp As Inventor.Application
    Public Sub Activate(AddInSiteObject As ApplicationAddInSite, FirstTime As Boolean)
Implements ApplicationAddInServer.Activate
        MsgBox("My First Add In Activated")
        m_inventorApp = AddInSiteObject.Application
        MsgBox(m_inventorApp.UserName)
    Public Sub Deactivate() Implements ApplicationAddInServer.Deactivate
        Throw New NotImplementedException()
    End Sub
    Public Sub ExecuteCommand(CommandID As Integer) Implements
ApplicationAddInServer.ExecuteCommand
        Throw New NotImplementedException()
    End Sub
    Public ReadOnly Property Automation As Object Implements ApplicationAddInServer.Automation
        Get
            Throw New NotImplementedException()
        End Get
    End Property
End Class
```

Tutorial 250320-01	Autodesk Inventor	Addin	Vb.net	Arthur Knoors

## Create a button definition:

First, we need to create two global properties as below.

Imports System.Runtime.InteropServices

Imports Inventor

<ProgldAttribute("MyInventorAddIn.StandardAddInServer"),

GuidAttribute("ad0a3353-0c3b-42a6-87ba-e1a86f4529e7")>

Public Class StandardAddInServer

Implements Inventor.ApplicationAddInServer

Private m\_inventorApp As Inventor.Application

Private WithEvents m\_sampleButton As ButtonDefinition

Private guid As String = Nothing

Public Sub Activate(AddInSiteObject As ApplicationAddInSite, FirstTime As Boolean) Implements ApplicationAddInServer.Activate

MsgBox("My First Add In Activated")

m\_inventorApp = AddInSiteObject.Application

MsgBox(m\_inventorApp.UserName)

<b>kit</b>				
Tutorial 250320-01	Autodesk Inventor	Addin	Vb.net	Arthur Knoors
Create the GUID f	for the Button defi	nition:		
Dim t As Type = (	GetType(MyInventorAd	din.StandardAddInSe	erver)	
Dim customAttri	ibutes() As Object = t.G	etCustomAttributes(	GetType(GuidAttribute	<mark>), False)</mark>
Dim guidAttribut	e As GuidAttribute = C	Type(customAttribute	es(0), GuidAttribute)	
guid = "{" + guid4	Attribute.Value.ToString	() + ")"		
Catch				
End Try				

N.				
Tutorial 250320-01	Autodesk Inventor	Addin	Vb.net	Arthur Knoors

#### Create the Button Definition selves (KISS):

We create a simple button without bells (KISS) with the code below. We need to use the above create GUID.

m\_sampleButton = m\_inventorApp.CommandManager.ControlDefinitions.AddButtonDefinition("My First Button", "MyFirstButton",Inventor.CommandTypesEnum.kShapeEditCmdType, guid, "My Description", "My Tool Tip Text",,, ButtonDisplayEnum.kDisplayTextInLearningMode)

If FirstTime Then

AddToUserInterface()

End If

End Sub

<b>kit</b>				
Tutorial 250320-01	Autodesk Inventor	Addin	Vb.net	Arthur Knoors

### Add this button to the Ribbon of Autodesk Inventor (ZeroDoc and ToolsTab)

We create a button by adding this code to the class in the Ribbon of Inventor In the Tools tab of ZeroDoc.

You can also choose instead of ZeroDoc, (Part Assembly Drawing, IFeature, Presentation).

Private Sub AddToUserInterface()

Dim r As Ribbon = m\_inventorApp.UserInterfaceManager.Ribbons.Item("ZeroDoc") 'Part Assembly Drawing, Ifeature, Presentation

Dim rt As RibbonTab = r.RibbonTabs.Item("id\_TabTools")

Dim rp As RibbonPanel = rt.RibbonPanels.Add("My First Ribbon Panel", "MyFirstRibbonPanel", guid)

Dim cmd As CommandControl = rp.CommandControls.AddButton(m\_sampleButton)

End Sub

Add the Button Event on execute so the it can be executed:   Mathematical State Research and Researc	Tutorial 250320-01	Autodesk Inventor	Addin	Vb.net	Arthur Knoors
An analysis of the second left start of Contrast As New Workshow Map Frenches on Survivo left transformation and Addin Buttom, Magdin Style, Yesho, French Add in Buttom, Contrast Add in Buttom, Cont	Add the Button Ev	vent on execute so	the it can l	pe executed:	
MagReel, You just pressed your first Addin Butten , MagRee Style, Yeake, "Free Add is Butten" Fuel Sub Public Sub Deactivate() Implements ApplicationAddInServer.Deactivate Throw New NotImplementedException() End Sub Public Sub ExecuteCommand(CommandID As Integer) Implements pplicationAddInServer.ExecuteCommand Throw New NotImplementedException() End Sub Public ReadOnly Property Automation As Object Implements ApplicationAddInServer.Automation Get Throw New NotImplementedException() End Get End Get End Property ind Class	Private Sub m_samp	leButton_OnExecute(C	Context As Nar	neValueMap) Handles r	n_sampleButton.OnEx
Fublic Sub Deactivate() Implements ApplicationAddInServer.Deactivate         Throw New NotImplementedException()         End Sub         Public Sub ExecuteCommand(CommandID As Integer) Implements         ApplicationAddInServer.ExecuteCommand         Throw New NotImplementedException()         End Sub         Public ReadOnly Property Automation As Object Implements ApplicationAddInServer.Automation         Get         Throw New NotImplementedException()         End Get         End Get         End Property         End Class	MsgBox("You just p	oressed your first Addin	Button", Msg	BoxStyle.YesNo, "First A	dd in Button")
Public Sub Deactivate() Implements ApplicationAddInServer.Deactivate Throw New NotImplementedException() End Sub Public Sub ExecuteCommand(CommandID As Integer) Implements ApplicationAddInServer.ExecuteCommand Throw New NotImplementedException() End Sub Public ReadOnly Property Automation As Object Implements ApplicationAddInServer.Automation Get Throw New NotImplementedException() End Get End Property End Class	End Sub				
Public Sub Deactivate() Implements ApplicationAddInServer.Deactivate         Throw New NotImplementedException()         End Sub         Public Sub ExecuteCommand(CommandID As Integer) Implements         ApplicationAddInServer.ExecuteCommand         Throw New NotImplementedException()         End Sub         Public ReadOnly Property Automation As Object Implements ApplicationAddInServer.Automation         Get         Throw New NotImplementedException()         End Get         End Get         End Property         End Class					
Throw New NotImplementedException() End Sub Public Sub ExecuteCommand(CommandID As Integer) Implements ApplicationAddInServer.ExecuteCommand Throw New NotImplementedException() End Sub Public ReadOnly Property Automation As Object Implements ApplicationAddInServer.Automation Get Throw New NotImplementedException() End Get End Property End Class	Public Sub Deactiva	te() Implements Applic	ationAddInSei	ver.Deactivate	
End Sub Public Sub ExecuteCommand(CommandID As Integer) Implements ApplicationAddInServer.ExecuteCommand Throw New NotImplementedException() End Sub Public ReadOnly Property Automation As Object Implements ApplicationAddInServer.Automation Get Throw New NotImplementedException() End Get End Property End Class	Throw New NotIm	olementedException()			
Public Sub ExecuteCommand(CommandID As Integer) Implements ApplicationAddInServer.ExecuteCommand Throw New NotImplementedException() End Sub Public ReadOnly Property Automation As Object Implements ApplicationAddInServer.Automation Get Throw New NotImplementedException() End Get End Property End Class	End Sub				
Public Sub ExecuteCommand(CommandID As Integer) Implements         ApplicationAddInServer.ExecuteCommand         Throw New NotImplementedException()         End Sub         Public ReadOnly Property Automation As Object Implements ApplicationAddInServer.Automation         Get         Throw New NotImplementedException()         End Get         End Get         End Property         End Class					
Throw New NotImplementedException() End Sub Public ReadOnly Property Automation As Object Implements ApplicationAddInServer.Automation Get Throw New NotImplementedException() End Get End Property End Class	Public Sub ExecuteC	Command(CommandIE er.ExecuteCommand	) As Integer) In	nplements	
End Sub Public ReadOnly Property Automation As Object Implements ApplicationAddInServer.Automation Get Throw New NotImplementedException() End Get End Property End Class	Throw New NotIm	olementedException()			
Public ReadOnly Property Automation As Object Implements ApplicationAddInServer.Automation Get Throw New NotImplementedException() End Get End Property	End Sub				
Public ReadOnly Property Automation As Object Implements ApplicationAddInServer.Automation Get Throw New NotImplementedException() End Get End Property End Class					
Get Throw New NotImplementedException() End Get End Property End Class	Public ReadOnly Pro	perty Automation As O	bject Impleme	ents ApplicationAddInSe	erver.Automation
Throw New NotImplementedException() End Get End Property End Class	Get				
End Get End Property End Class	Throw New Noth	mplementedException	()		
End Property End Class	End Get				
End Class	End Property				
	end Class				

Tutorial 250320-01	Autodesk Inventor	Addin	Vb.net	Arthur Knoors

## The complete code:

#### The complete code should look like this:

Imports System.Runtime.InteropServices

Imports Inventor

<ProgldAttribute("MyInventorAddIn.StandardAddInServer"),

GuidAttribute("ad0a3353-0c3b-42a6-87ba-e1a86f4529e7")>

#### Public Class StandardAddInServer

Implements Inventor.ApplicationAddInServer

*Private m\_inventorApp As Inventor.Application* 

Private WithEvents m\_sampleButton As ButtonDefinition

Private guid As String = Nothing

Public Sub Activate(AddInSiteObject As ApplicationAddInSite, FirstTime As Boolean) Implements ApplicationAddInServer.Activate

MsgBox("My First Add In Activated") m\_inventorApp = AddInSiteObject.Application MsgBox(m\_inventorApp.UserName)

```
Try
```

Dim t As Type = GetType(MyInventorAddin.StandardAddInServer) Dim customAttributes() As Object = t.GetCustomAttributes(GetType(GuidAttribute), False) Dim guidAttribute As GuidAttribute = CType(customAttributes(0), GuidAttribute) guid = "{" + guidAttribute.Value.ToString() + "}" Catch

End Try

*m\_sampleButton = m\_inventorApp.CommandManager.ControlDefinitions.AddButtonDefinition("My First Button", "MyFirstButton", Inventor.CommandTypesEnum.kShapeEditCmdType, guid, "My Description", "My Tool Tip Text",,, ButtonDisplayEnum.kDisplayTextInLearningMode)* 

#### If FirstTime Then

•				
Tutorial 250320-01	Autodesk Inventor	Addin	Vb.net	Arthur Knoors
AddToUserInterfac	:e()			
End If				
End Sub				
Private Sub AddToUse	rInterface()			
Dim r As Ribbon = m_ Ifeature, Presentation	_inventorApp.UserIn	terfaceManag	er.Ribbons.Item("ZeroD	oc") 'Part Assembly Drawing,
Dim rt As RibbonTab	= r.RibbonTabs.Item	n("id_TabTools	")	
Dim rp As RibbonPar	nel = rt.RibbonPanel	s.Add("My Firs	t Ribbon Panel", "MyFirs	stRibbonPanel", guid)
Dim cmd As Comma	ndControl = rp.Com	mandControl	s.AddButton(m_sample	Button)
End Sub				
Private Sub m_sample	Button_OnExecute(	Context As Na	meValueMap) Handles i	m_sampleButton.OnExecute
MsgBox("You just pre	essed your first Addi	n Button", Msg	BoxStyle.YesNo, "First A	Add In Button")
End Sub				
Public Sub Deactivate	() Implements Applic	ationAddInSe	rver.Deactivate	
Throw New NotImple	ementedException()			
End Sub				
Public Sub ExecuteCo ApplicationAddInServer.	mmand(Commandll ExecuteCommand	D As Integer) Ir	mplements	
Throw New NotImple	ementedException()			
End Sub				
Public ReadOnly Prope	erty Automation As C	Dbject Implem	ents ApplicationAddInS	erver.Automation
Get				
Throw New NotImp	olementedException	0		
End Get				
End Property				
End Class				

Page 8

Tutorial 250320-01	Autodesk Inventor	Addin	Vb.net	Arthur Knoors

# Testing:

Rebuild again (see tutorial001) and start Autodesk Inventor'

You should receive:



The username from the Inventor Software:

Invontor	Microsoft.VisualBasic.Core	×
TIMEIIIO	А.К.	
	ОК	

And when Inventor is loaded:



When pressing the button you should receive this message:

ntent :as	Team Web	My First Button	•					
	Team Web	My First Ribbon Panel						
	First Add In Button							
	You just pressed your first Addin Button							
		Ja Ne	e					

<b>kit</b>				
Tutorial 250320-01	Autodesk Inventor	Addin	Vb.net	Arthur Knoors

Congrats you are now a step further and created your first add-in with a working button.