

Criando um Add-In no Visual Basic 6

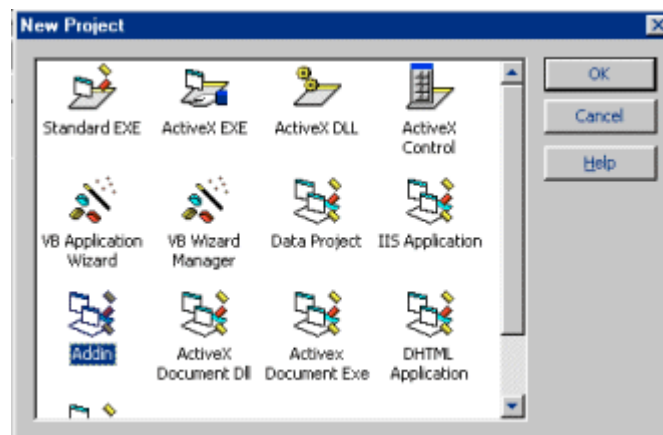
Por [John P. Cunningham](#)

Definindo um diretório, renomeando o formulário e salvando o projeto

Passo 1: Crie um subdiretório na diretório do Visual Basic chamado ConverterAddIn. (Normalmente, a pasta do VB está em C:\Arquivos de Programas\Microsoft Visual Studio\VB98, então sua nova pasta seria "C:\Arquivos de Programas\Microsoft Visual Basic\VB98\ConverterAddIn").

Quando selecionamos Add-In em New Project Type, abrimos o Add-In Designer Template. Ele está localizado no diretório "...VB98\Template\Projects\AddIn.vbp". Se fizermos qualquer alteração e salvá-la, as mudanças seriam salvas nessa pasta e poderiam causar confusão mais tarde. Então criando uma nova pasta, ConverterAddIn, renomeando o formulário para frmConvert e renomeando o projeto para ConverterAddIn, temos que por nosso projeto em sua própria pasta para afastar assim qualquer problema com futuros projetos Add-In e seus designer associados.

Passo 2: Inicie o Visual Basic e no menu File selecione "New Project". Do diálogo New Project, escolha "AddIn" e clique no botão OK.



Caixa de diálogo New Project

O VB remove os formulários existentes e os substitui pelo "ADD-IN Designer Template".

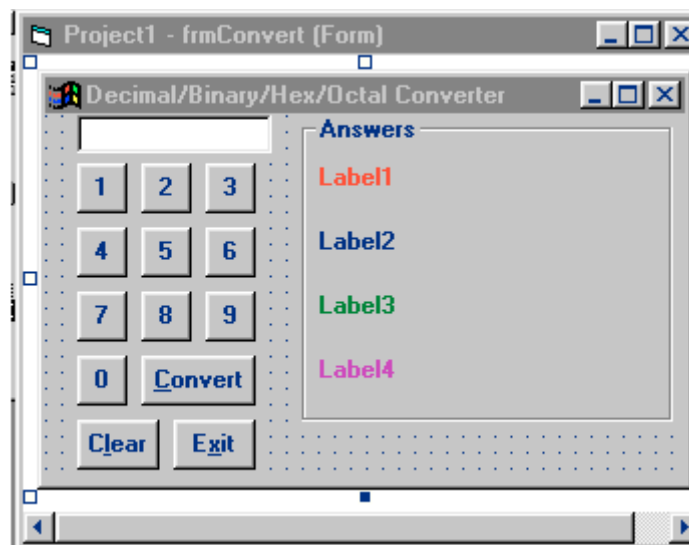
Na janela Project, clique no formulário e no Designer para expandi-los. Dê um clique duplo sobre o formulário para exibi-lo.

Construindo e Codificando o Projeto Conversor de Números

O próximo passo será adicionar uma caixa de texto, quatro etiquetas de texto e outro botão de comando a frmConvert. Primeiramente, altere as captions dos dois botões existentes de "OK" e "Cancel" para "Convert" e "Exit", então adicione o terceiro botão, definindo sua caption para "Clear" e o nomeando como cmdClear. Você pode colocar as etiquetas de texto dentro de um Frame, mas não é necessário.

Depois, adicionaremos os botões de números como uma matriz. Comece colocando outro botão de comando no formulário, defina sua Caption como 0 e seu nome para Btn. Usando o botão direito do mouse, clique no Btn para selecioná-lo e escolha "Copy", depois, clique com o botão direito do mouse novamente agora no formulário e escolha "Paste". Clique em "yes" na caixa de mensagem do VB que aparecerá. O VB responde colocando uma cópia do botão original no canto superior esquerdo do formulário. Modifique a caption desse novo botão para 1, uma olhada na janela de propriedades revela que o nome deste botão é Btn(1) enquanto que o nome do botão original é Btn(0).

Isso era exatamente o que nós queríamos, criar uma matriz de controles para todos os botões de números. Proceda dessa maneira para criar do botão 2 até o 9. No final, você deve ter dez botões de números etiquetados de 0 a 9 e seus nomes serão de Btn(0) até Btn(9). Seu formulário deve parecer agora com o seguinte;



Desenho do formulário

Você notará que eu mudei a Caption do formulário também. Agora estamos prontos para codificar o formulário.

Se você olhar a seção General Declarations do código do formulário, você verá o seguinte:

```
Public VBInstance As VBIDE.VBE
Public Connect As Connect
Option Explicit
```

Estas três linhas de código são parte do Add-In padrão. Estamos mudando a Segunda linha de acordo com o que nosso programa propõe. Altere a Segunda linha para o seguinte:

```
'Altere o nome do designer aqui - na segunda linha
Public Connect As NumConverter
```

Agora adicione o comando "Option Explicit" na seção General Declarations e:

```
Dim Adecimal, digit, h, i, j, k As Integer
Dim jbin As String
```

Depois disso, adicionaremos a subrotina DecBin ao projeto, depois o código abaixo. Não investigaremos em nossos trabalhos esta rotina, basta dizer que converte um número inteiro ou decimal para um número binário.

```
Public Sub DecBin(Adecimal, jbin As String)
'Converte um valor inteiro para string equivalente de dígitos binários
jbin = ""
h = Hex(Adecimal) 'Converte de inteiro para hexadecimal
For i = 1 To Len(h)
    digit = InStr("0123456789ABCDEF", Mid(h, i, 1)) - 1
    j = 8
    k = 4
    Do 'Converte de hexadecimal para binário
        jbin = jbin + Right(Str((digit * j) Mod 2), 1)
        j = j - (j * 2)
        k = k - 1
        If k = 0 Then Exit Do
    Loop While j
Next i
End Sub
```

A seguir é o código que completa os eventos do formulário:

```
Private Sub Btn_Click(Index As Integer)
    Text1 = Text1 & Index
End Sub

Private Sub CancelButton_Click()
    Text1.Text = ""
    'para limpar a memória quando nós desconectamos o Add-In
    Text1.SetFocus
    Frame1.Visible = False
    Connect.Hide
End Sub

Private Sub cmdClear_Click()
    Text1.Text = "" 'Limpa a caixa de texto
    Text1.SetFocus 'para permitir nova conversão
    Frame1.Visible = False
End Sub

Private Sub Form_Load()
    'Centraliza o formulário na tela
    Me.Move (Screen.Width - Me.Width) / 2, (Screen.Height - Me.Height) / 2
End Sub

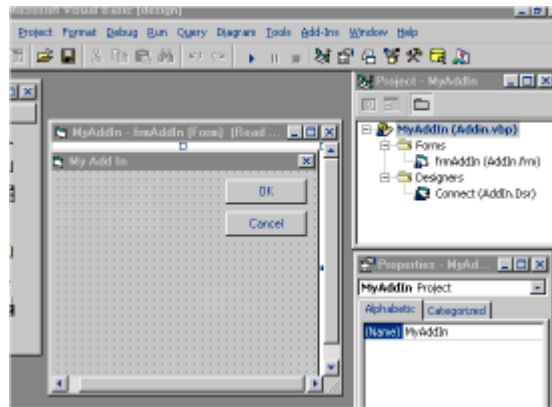
Private Sub OKButton_Click()
    'Comente o código original posto aqui pelo Add-In padrão
    MsgBox "AddIn operation on: " & VBInstance.FullName
    Adecimal = Val(Text1) 'get the number from the TextBox
    Frame1.Visible = True
    'Faz a chamada para conerter de decimal para binário
    Call DecBin(Adecimal, jbin)
```

```

Label1.Caption = "Decimal = " & Text1
Label2.Caption = "Binary = " & jbin 'exibe o resultado
Label3.Caption = "Hex = " & Hex(Adecimal)
Label4.Caption = "Octal = " & Oct (Adecimal)
End Sub

```

Era isso o código dos eventos do projeto, agora a próxima parte.



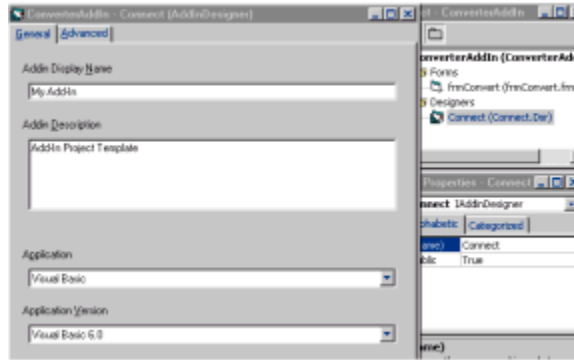
Ampliado!

Se você executar esse programa nesse atual estado dentro do VB, nada de mais aconteceria exceto o aparecimento da janela intermediária. Isso porque os Add-Ins criados com o Add-In padrão são compilados em DLLs (Bibliotecas de vínculos dinâmicos). Se você compilar o projeto nesse ponto, você teria um Add-In adicionado à IDE do VB chamado "MyAddIn". Sempre que "MyAddin" for carregado na janela de Add-Ins da barra de menu principal uma caixa de mensagem apareceria dizendo "AddIn operation on". Esse é um bom começo, mas caixas de mensagens não são de grande ajuda para converter números.

Na janela de propriedades do projeto, acessível pelo menu Project, mude o nome do projeto para ConverterAddIn. Depois de ter feito isso, clique OK e defina o nome do formulário como frmConvert. A seguir, salve o projeto na subpasta "ConverterAddIn" que nós criamos no passo 1. Use "Save Project As" do menu File Menu e salve o projeto "ConverterAddIn".

Faça mudanças no Designer do Add-In

Clicando em "Designers" e então em "Connect(Connect.Dsr)" na Project Window o designer será exibido.



[Blow Up!](#)

Na janela de propriedades, clique em Name e altere-o para "NumConverter", então salve-o clicando nele com o botão direito na janela de projeto e use "Save Connect.Dsr As" e salve-o como "NumConverter" na nossa subpasta ConverterAddIn. Sua janela de projetos e de propriedades deveria aparecer como o mostrado abaixo;



[Blow Up!](#)

O próximo passo é abrir o Designer com um clique com o botão direito do mouse em "NumConverter(NumConverter.Dsr)" dentro da pasta Designers na janela de projeto. Esta abre um menu secundário e queremos selecionar "View Object".

Agora maximize o Designer clicando no botão maximizar no canto superior direito do formulário do designer para mostrar todas as caixas de entrada;



[Blow Up!](#)

Modifique o "Add-In Display Name" de "MyAddIn" para "Conversor de números". Altere "Add-In Description" de "Add-In Project Template" para "Converte um número em base dez para binário, hexadecimal e octal".

Em "Application" deve constar "Visual Basic" e em "Application Version" deve ser "Visual Basic 6.0" como mostrado acima. "Initial Load Behavior" pode ser definido

como Startup e caixa de seleção "Addin is command-line safe" box deve estar desmarcada. Feche o formulário de designer.

Agora abra a janela de código do designer com um clique direito sobre "NumConverter" em Designers na janela de projeto para exibir o menu e selecione "View Code";

Na seção General Section você verá.

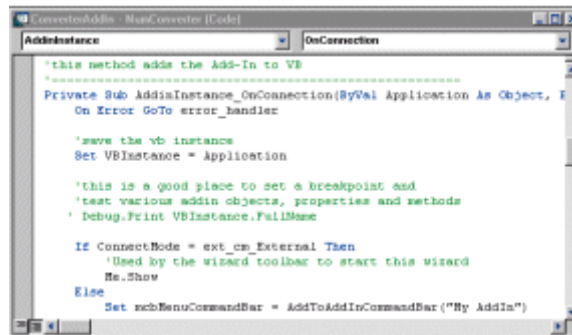
```
Option Explicit
Public FormDisplayed As Boolean
Public VBInstance As VBIDE.VBE
Dim mcbMenuCommandBar As Office.CommandBarControl
Dim mfrmAddIn As New frmAddIn
Public WithEvents MenuHandler As CommandBarEvents
'***Modifique Dim mfrmAddIn As New frmAddin pelo seguinte;***
Dim mfrmConvert As New frmConvert
```

Depois, role abaixo a caixa de combinação.



MsgBox Generator

E abra a sub "AddinInstance - OnConnection".



Blow Up!

Comente a linha "Debug.Print VBInstance.FullName". E cinco linhas abaixo;

```
Set mcbMenuCommandBar = AddToAddInCommandBar ("My AddIn")
```

Modifique essa linha por:

```
Set mcbMenuCommandBar = AddToAddInCommandBar ("ConverterAddIn")
```

Ok, estamos quase terminando. Ao fazer essas alterações, estamos forçando o VB a usar um nome e formulário de Add-In que reflete nosso conversor de números ao invés do nome "MyAdd-In".

Existem apenas algumas mudanças para limpar a casa que faltam e faremos as mudanças finais no código do designer agora. Certifique-se antes de que nós fizemos a seguinte mudança na seção General Declarations do designer:

```
'***Altere Dim mfrmAddIn As New frmAddin para o seguinte;***  
Dim mfrmConvert As New frmConvert
```

Infelizmente, o termo "mfrmAddIn" é ainda empregado no código do nosso designer mais oito vezes e "frmAddIn" sobrou em um lugar. Eles ocorrem nas seguintes sub-rotinas além do comando Dim:

Termo	Mudar para:	Na sub-rotina	Ocorrências
mfrmAddIn	mfrmConvert	General-Show	5
mfrmAddIn	mfrmConvert	General-Hide	1
mfrmAddIn	mfrmConvert	AddinInstance-OnDisconnection	2
frmAddIn	frmConvert	General-Show	1

Você pode fazer as correções necessárias usando o "Find and Replace" (Ctrl H) do VB no menu Edit. (Não se preocupe se você não fez todas as mudanças corretamente porque quando você tentar compilar a Dll, o VB apresentará uma mensagem de erro e então você poderá fazer sua própria correção).

Ok, último passo, salve todo seu trabalho e clique no menu File.

Selecione "Make ConverterAddIn.dll...". Quando o VB terminar de fazer a Dll, clique no menu "Add-Ins" e escolha "Add-In Manager" para ver o "Number Converter" listado em "Available Add-Ins". Selecione-o e clique em Loaded/Unloaded no canto inferior direito, clique cancel para fechar. Agora clique no menu "Add-Ins" novamente e "Number Converter" aparece no menu, selecione-o e o Add-In se abre pronto para usá-lo. Daqui por diante, o "Number Converter Add-In" será ativado no menu add-Ins toda vez que o VB for carregado.

Conclusão

Bem, temos então nosso próprio Add-In! Ele pode ser útil apenas como um copo de chocolate-quente para muita gente. Entretanto, como sempre no VB, as coisas que você pode fazer são limitadas apenas pela sua imaginação. Se você fizer um Add-In legal que você acha que pode beneficiar a mais gente, mande-o para nós que ele pode acabar na página de demos!

Fonte: <http://vb-world.net>

Tradução: JV Informática (jvinf@yahoo.com)