

Building VMware Visual Basic Applications



I am a huge fan of the website run-virtual.com, on November the 16th 2006 Richard Garthagen yelled [Yippie!!](#) in one of his posts and that was for a reason. Richard was able to talk to VirtualCenter 2.0.1 via SOAP with his own Visual Basic application. On November the 21th he published his first application with the name VmotionInfo. That was the eye opener for me. I immediately visited the vmware.com website to get my hands on as much SDK info as possible. Unfortunately the VB example folder was empty. During the next months I was struggling with C# and Google-ling like crazy but my VB application could not communicate with VirtualCenter. In the beginning of April 2007 I attended the TSX in Nice and got up early to visit the V13 Perl SDK lab hosted by Jeremy van Doorn.

Afterwards I [interviewed](#) Jeremy and asked him if VMware is going to release the SDK examples for Visual Basic. His answer was that it will be released this year? This year I thought? I want to start writing my apps right now. This week I had a breakthrough. I only slept for a few ours last night but it paid off. My Visual Basic application is communicating with VirtualCenter and "logging on" takes an eye blink. Yippie!!! I only have till Friday to work on my first app because after next weekend I am delivering the V13 Install and Configure training in The Hague and Eindhoven. So I have decided that I am going to share my invention.

This document describes how to build a VB Application that can communicate with Virtual Center 2.0.1. It's written by Eric Sloof (esloof@ntpro.nl). As ever my documents are released as is and without warranty.

The first thing you have to do is get your hand on a free copy of the Microsoft Visual Basic 2005 Express Edition. You can download and register a copy at the following location: <http://msdn.microsoft.com/vstudio/express/downloads/>

To communicate with Virtual Center you need the following Dynamic Link Library's.

VIClient.dll
VimService.dll
VimService.XmlSerializers.dll

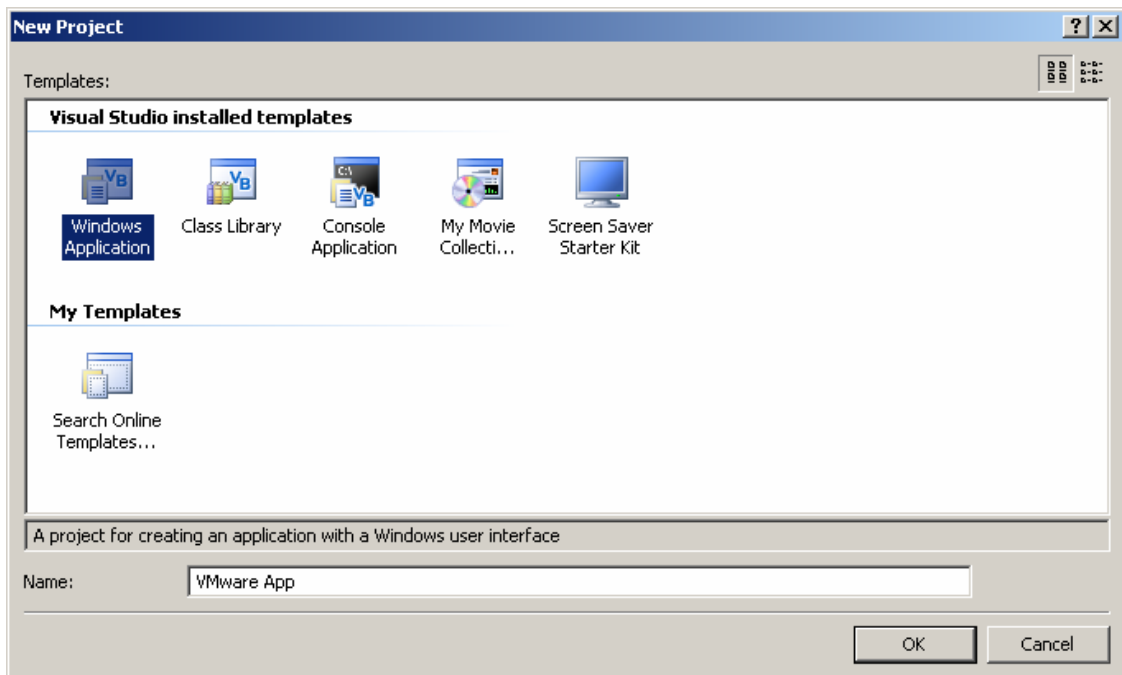
There is an issue with building these DLL files. So what I did is borrowing the pre-build files from Rob Baumstark (thanks Rob). He has written a handy tool to manage the MAC addresses of the virtual network cards. Visit the following article on Run-Virtual.com and download the ZIP file. When the download is ready, extract the three DLL files and place them in your empty project folder.

http://www.run-virtual.com/?page_id=173

Ok, we have everything we need, lets start the Microsoft Visual Basic 2005 Express Edition.

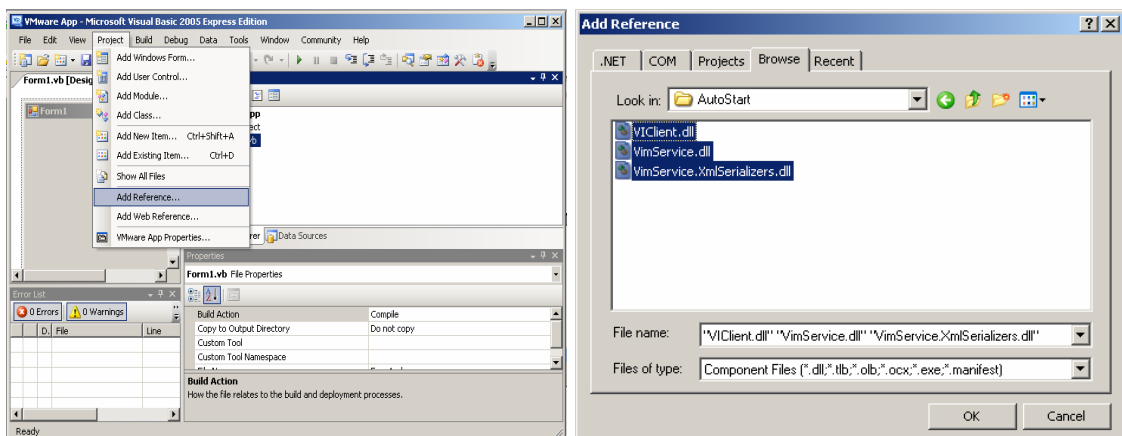


Choose the menu option <File> <New> <Project> and give your project a name.



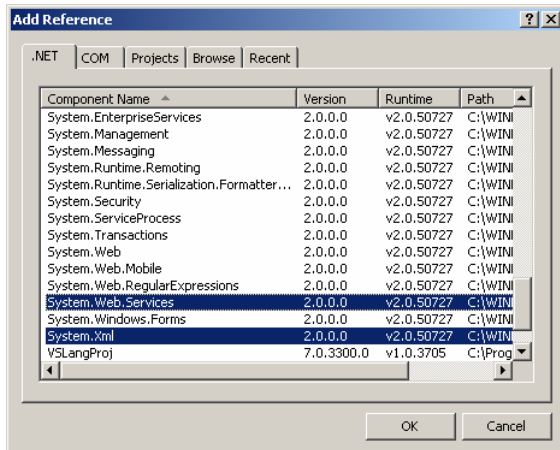
Click on the <OK> button.

Choose the <Project> <Add> <reference> menu option.



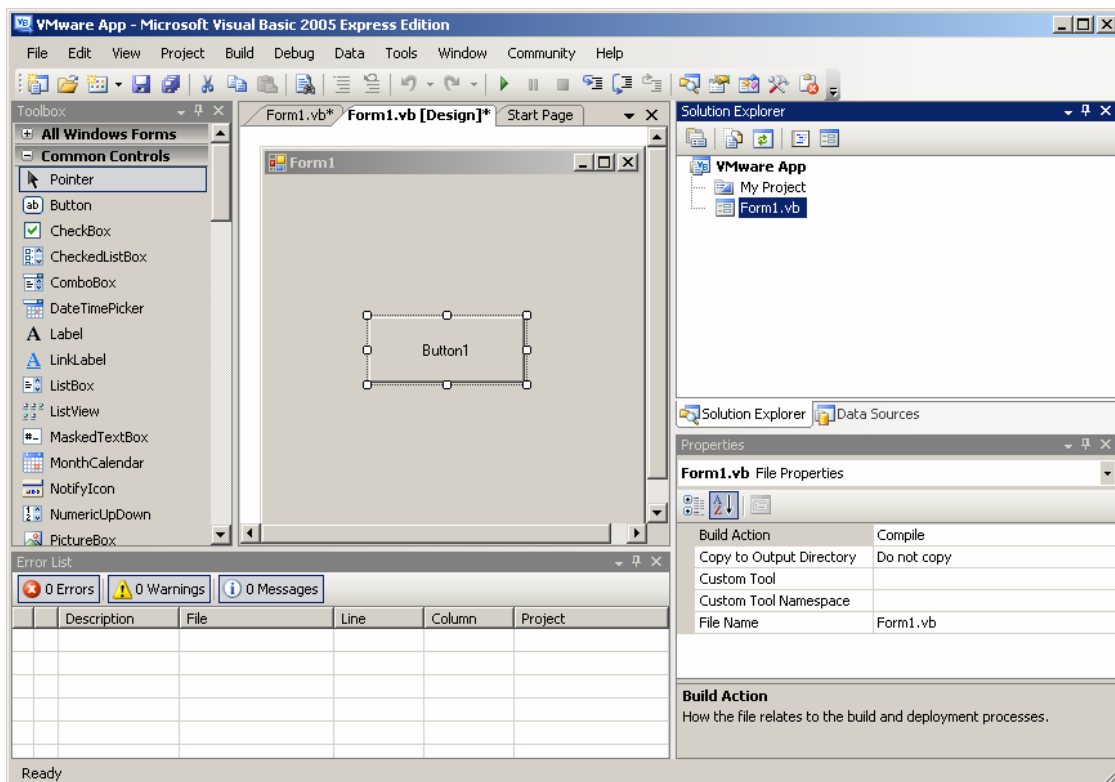
Select the three DLL files and Click on the <OK> button.

Choose the <Project> <Add> <Reference> menu option again and select the tab <.NET>. Add the System.Web.Services and System.Xml reference in the VB project.



On the VMware forum there is a discussion about an error during SOAP call with an article from [vtorch](#). I used the C# port to VB as the basis for my project (thanks vtorch).

You need to draw a new <Button>.



And paste the following code in your project. Double click the form, delete the existing code and just paste the next code into your project. Then change the server name user and password sentence.

```
sc.Connect("https://VitrualCenterServer/sdk", "Adminuser", "Password")
```

```

Imports System
Imports System.Web.Services.Protocols
Imports VimService

Public Class Form1

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Dim sc As New SimpleClient()
        sc.CreateServiceRef("ServiceInstance")

        sc.Connect("https://server/sdk", "Adminuser", "Password")
        sc.Disconnect()
    End Sub

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

    End Sub
End Class

Public Class SimpleClient
    Protected _service As VimService
    Protected _sic As ServiceContent
    Protected _svcRef As ManagedObjectReference
    Protected _propCol As ManagedObjectReference
    Protected _rootFolder As ManagedObjectReference
    Public Sub CreateServiceRef(ByVal svcRefVal As String)
        System.Net.ServicePointManager.CertificatePolicy = New
CertPolicy()
        _svcRef = New ManagedObjectReference()
        _svcRef.type = "ServiceInstance"
        _svcRef.Value = svcRefVal
    End Sub
    Public Sub Connect(ByVal url As String, ByVal username As String,
ByVal password As String)
        If _service IsNot Nothing Then
            Disconnect()
        End If
        _service = New VimService()
        _service.Url = url
        _service.CookieContainer = New System.Net.CookieContainer()
        _sic = _service.RetrieveServiceContent(_svcRef)
        _propCol = _sic.propertyCollector
        _rootFolder = _sic.rootFolder

        MsgBox(_sic.about.build)
        MsgBox(_sic.about.name)
        MsgBox(_sic.about.osType)
        MsgBox(_sic.setting.Value)
        MsgBox(_sic.rootFolder.Value)

        If _sic.sessionManager IsNot Nothing Then
            _service.Login(_sic.sessionManager, username, password,
Nothing)
        End If
    End Sub
End Class

```

```

Public Sub Disconnect()
    If _service IsNot Nothing Then
        _service.Logout(_sic.sessionManager)
        _service.Dispose()
        _service = Nothing
        _sic = Nothing
    End If
End Sub

<STAThread()> _
Public Shared Sub Main(ByVal args As String())
    If args Is Nothing OrElse args.Length < 2 Then
        MsgBox("Usage : SimpleClient <webserviceurl> <username>
<password>")
    End If
    Dim now1 As DateTime = DateTime.Now
    Dim sc As New SimpleClient()
    Try
        sc.CreateServiceRef("ServiceInstance")
        sc.Connect(args(0), args(1), args(2))
        sc.Disconnect()
        Dim now2 As DateTime = DateTime.Now
        MsgBox("Start Time : " + now1.ToLongDateString() + " " +
now1.ToLongTimeString())
        MsgBox("End Time : " + now2.ToLongDateString() + " " +
now2.ToLongTimeString())
        Console.Out.WriteLine("Done")
    Catch se As SoapException
        MsgBox("Caught SoapException - " + " Actor : " +
se.Actor.ToString() + " Code : " + se.Code.ToString() + " Detail XML :
" + se.Detail.OuterXml)
    Catch e As Exception
        MsgBox("Caught Exception : " + " Name : " +
e.[GetType]().Name + " Message : " + e.Message + " Trace : " +
e.StackTrace.ToString())
    End Try
End Sub

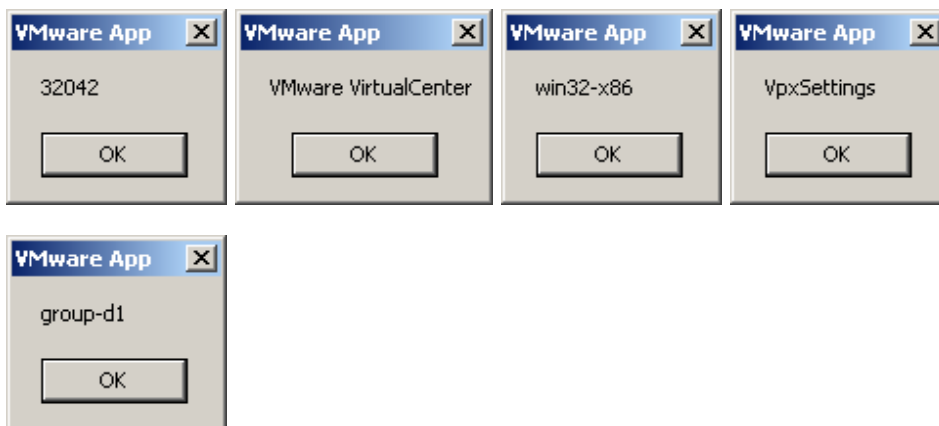
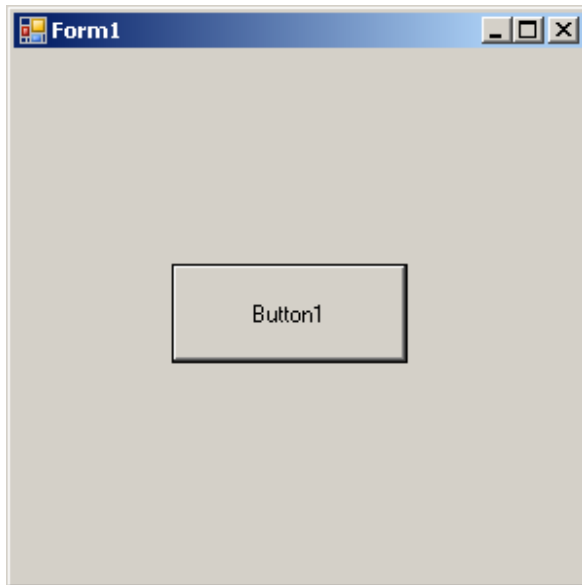
End Class

Public Class CertPolicy
    Implements Net.ICertificatePolicy
    Public Function CheckValidationResult(ByVal srvPoint As
System.Net.ServicePoint, ByVal certificate As
System.Security.Cryptography.X509Certificates.X509Certificate, ByVal
request As System.Net.WebRequest, ByVal certificateProblem As Integer)
As Boolean Implements
System.Net.ICertificatePolicy.CheckValidationResult
        Return True
    End Function
End Class

```

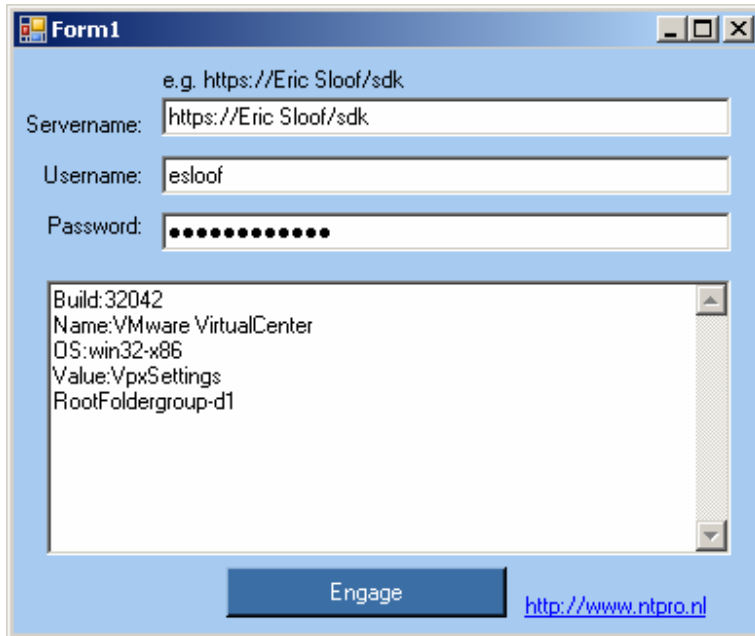
Let's see if it works.....

Press F5 wait and you should see this.



Yippy !!!

I know the code is a bit crappy but I depend on you to improve it. My colleague Edwin Akker created some textboxes so you can fill in the server name, username and password. When it's ready you can find it on my website.



The screenshot shows a Windows application window titled "Form1". It contains three input fields: "Servername:" with the example text "e.g. https://Eric Sloof/sdk" and the entered value "https://Eric Sloof/sdk"; "Username:" with the value "esloof"; and "Password:" with a masked field of ten dots. Below these is a text area containing system information: "Build:32042", "Name:VMware VirtualCenter", "OS:win32-x86", "Value:VpxSettings", and "RootFoldergroup-d1". At the bottom, there is an "Engage" button and a URL "http://www.ntpro.nl".

You can build and publish your exe now, please share as much code as possible. When you have trouble with the .net runtime you can download a copy here :

<http://www.microsoft.com/downloads/info.aspx?na=90&p=&SrcDisplayLang=en&SrcCategoryId=&SrcFamilyId=0856EACB-4362-4B0D-8EDD-AAB15C5E04F5&u=http%3a%2f%2fdownload.microsoft.com%2fdownload%2f5%2f6%2f7%2f567758a3-759e-473e-bf8f-52154438565a%2fdotnetfx.exe>

I want to take this opportunity to thank some people.

The credits are going to Richard Garsthagen, Jeremy van Doorn, vtorch, Rob Baumstark ,Edwin Akker and Esther Sloof.

Greetings from Eric Sloof

Eric Sloof is a VMware Certified Instructor based in the Netherlands, he is working as an IT specialist for over 15 years and is running the www.ntpro.nl website.

NTPRO.NL
Eric Sloof
Biezenhaak 11
3356 CR Papendrecht
The Netherlands

Telefoon +31 786412628
Mobiel +31621265027
Fax +31786155248
E-mail esloof@ntpro.nl

