

## Re: Conversion from VB6 to VB.NET

---

*Source:*

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.vb/2008-04/msg00892.html>

---

- *From:* "Tom Shelton" <[tom\\_shelton@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:tom_shelton@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx)>
  - *Date:* Wed, 16 Apr 2008 00:50:30 -0600
- 

"Mike M" <[Mike M@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:Mike M@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx)> wrote in message <news:922314A3-6E09-4EDD-9EE0-9E66D8717A9B@xxxxxxxxxxxxxxxxxxxx>

Hi,

My conversion seems from VB6 to VB.Net of an interface to an API seem to have gone OK, but when I look at the output from the Debug.Print(" Active server is " & xmlServerActive) statement, xmlServerActive only contains one character(the first one). I suspect it is being truncated. I am pretty much a novice at this sort of thing so any advice would be welcome.

I did a google search, and it sure looks like this api your referencing – actually has a .NET version already, so you might want to check into that. Never the less:

Here is the old VB6 code:

Listening for Servers and System Status using VB

Declare the functions

```
Public Declare Function RCA_listenForServersBstr Lib "clientapi.dll" _
```

```
(ByVal serverActiveCallback As Long, _
```

```
ByVal serverShutdownCallback As Long) As Integer
```

```
Public Declare Function RCA_listenForSystemStatusBstr Lib "clientapi.dll" _
```

```
(ByVal systemStatusCallback As Long, _
```

```
ByVal clusterCreatedCallback As Long, _
```

```
ByVal clusterDeletedCallback As Long, _
```

```
ByVal systemExceptionCallback As Long) As Integer
```

Declare callback functions

```
Public Function serverActiveCallback(ByVal xmlServerActive As String) As Long
```

```
Debug.Print "Active server is " & xmlServerActive
```

```
End Function
```

```
Public Function serverShutdownCallback(ByVal server As String) As Long
```

```
Debug.Print "Shutdown server is " & server
```

```
End Function
```

```
Public Function systemExceptionCallback(ByVal error As String) As Long
```

```
MsgBox "Error : " & error
```

## Re: Conversion from VB6 to VB.NET

```
End Function
Public Function systemStatusCallback(ByVal systemStatus As String) As Long
Debug.Print "System status is " & systemStatus
End Function
Public Function clusterCreatedCallback(ByVal cluster As String) As Long
Debug.Print "Created cluster is " & cluster
End Function
Public Function clusterDeletedCallback(ByVal cluster As String) As Long
Debug.Print "Deleted cluster is " & cluster
End Function
    Set up to listen for servers
ret = RCA_listenForServersBstr(AddressOf serverActiveCallback, _
AddressOf serverShutdownCallback)
    Set up to listen for system status
ret = RCA_listenForSystemStatusBstr(AddressOf systemStatusCallback, _
AddressOf clusterCreatedCallback, _
AddressOf clusterDeletedCallback, _
AddressOf systemExceptionCallback)
```

The new VB.NET (It has some other pieces I have added)  
Option Strict Off

I would turn Option Strict On – pretty much always. There are few cases, such as late binding where it is useful to have it turned off, but I would not make it the general case.

Option Strict On

Imports System.Runtime.InteropServices ' save your self some typeing :)

Option Explicit On  
Module RimagesAPI

Unless your planning to use this structure with VB.NET file IO – I would get rid of the VBFixedString attribute. Without the documentation and seeing the VB6 declaration, I would probably change the char arrays below to just plain old strings. I'm not sure what char set this is using, but I'll assume ansi:

```
<StructLayout (LayoutKind.Sequential, CharSet:=CharSet.Ansi) > _
Public Structure RCA_production_order_description
<MarshalAs(UnmanagedType.ByValTStr, SizeConst:=128)> _
Public orderId As String ' This field cannot be NULL

<MarshalAs(UnmanagedType.ByValTStr, SizeConst:=128)>
Public clientId As String ' This field is optional and can be NULL
....
End Structure
```

Re: Conversion from VB6 to VB.NET

Public Declare Ansi Function RCA\_connect Lib "clientapi.dll" (ByVal clientId As String) As Short

Anyway, I am making some assumptions here that may or may not be correct, depending on this particular API – so I can't guarantee you that this is going to work :)

—

Tom Shelton

.