

Forum:[Hack Forums](#)

Thread: [\[C#/VB\] Simple instance merger class](#)

BahNahNah - November 18, 2015, 1:47 am

Download: [GIST](#)

Simple code that passes the commandline arguments to an existing version of the application (if it exists).

Gif:

Spoiler [\(Click to View\)](#)

Example code:

Code public static void Main(string[] args)

```
{
    SingleInstance.Create(OnNewInstance); //Setting up single instance of the current program

    Console.WriteLine("First instance created");
    Console.ReadLine();
}
```

private static void OnNewInstance(string path, string[] args)

```
{
    Console.WriteLine("New instance created. Path: {0}", path);
    for(int i =0; i < args.Length;i++)
        Console.WriteLine("Arg {0}: {1}", i, args[i]);
}
```

VB:

Code Sub Main()

SingleInstance.Create(AddressOf OnNewInstance)

```
    Console.WriteLine("First instance created")
    Console.ReadLine()
End Sub
```

Private Sub OnNewInstance(ByVal path As String, ByVal args As String())

```
    Console.WriteLine("New instance created. Path: {0}", path)
    For i = 0 To args.Length - 1
        Console.WriteLine("Args {0}: {1}", i, args(i))
    Next
End Sub
```

End Sub

Todo:

-Find a better way to store port.

DisguiserNET - November 18, 2015, 1:52 am

thats a cool idea, but i can't think of a way for me to use it. mind telling me, what you made it for?

BahNahNah - November 18, 2015, 1:54 am

(Nov 18, 2015 02:52 AM)DisguiserNET Wrote: [\[CITE\]](#) thats a cool idea, but i can't think of a way for me to use it. mind telling me, what you made it for?

Associating a file type to your application (just passes the location of the associated file to the existing instance).

WebLaunched programs (like magnet links for torrents).

stratex - November 18, 2015, 1:56 am

(Nov 18, 2015 02:52 AM)DisguiserNET Wrote: [\[CITE\]](#) thats a cool idea, but i can't think of a way for me to use it. mind telling me, what you made it for?

This could probably be used for something like a web browser where instead of opening a new instance, it would open a new tab.

Only3 - November 18, 2015, 7:37 am

this can easy done with process class and Environment.GetCommandLineArgs

Washi - November 18, 2015, 8:41 am

Interesting concept, although why would you use sockets instead of pipes? Wouldn't that be easier as it takes away the need for a port?

<https://msdn.microsoft.com/en-us/library/bb546102%28v=vs.110%29.aspx>

BahNahNah - November 18, 2015, 2:42 pm

(Nov 18, 2015 09:41 AM)TheUnknownProgrammer Wrote: [\[CITE\]](#) Interesting concept, although why would you use sockets instead of pipes? Wouldn't that be easier as it takes away the need for a port?

<https://msdn.microsoft.com/en-us/library/bb546102%28v=vs.110%29.aspx>

I thought pipelines where .net 4?

I try to make my programs .Net 2.0 when it is possible, so its more compatible with other peoples projects.

(Nov 18, 2015 08:37 AM)Only3 Wrote: [\[CITE\]](#) this can easy done with process class and Environment.GetCommandLineArgs
Example?

Only3 - November 18, 2015, 3:08 pm

```
Code if (Args.Length < 2) {  
    Process.Start(new ProcessStartInfo(Application.ExecutablePath, "1"));  
} else if (Args(1) == "1") {  
    // New Instance  
}
```

BahNahNah - November 18, 2015, 3:17 pm

(Nov 18, 2015 04:08 PM)Only3 Wrote: [\[CITE\]](#) Code if (Args.Length < 2) {
 Process.Start(new ProcessStartInfo(Application.ExecutablePath, "1"));
} else if (Args(1) == "1") {
 // New Instance
}

This would not work. Did you even try it?

Tedd - November 18, 2015, 7:11 pm

Ported to vb. I really like this, useful as fuck.

```
SingleInstance.vb:  
Code Imports System.Threading  
Imports Microsoft.Win32  
Imports System.Net.Sockets  
Imports System.Net  
Imports System.Reflection  
Imports System.IO
```

```
''' <summary>  
''' Application instance merger  
''' Made by BahNahNah  
''' uid=2388291  
'''  
''' Ported to VB by Tedd  
''' </summary>
```

```
Public Class SingleInstance
```

```
    Public Delegate Sub MergeInstanceCallback(ByVal path As String, ByVal args As String())
```

```
    Private applicationMutex As Mutex = Nothing
```

```
    Private InstanceKey As String
```

```
    Private PortSave As RegistryKey = Nothing
```

```
    Private InstanceSocket As Socket = Nothing
```

```
    Private SocketBuffer As Byte()
```

```
    Private SenderEndpoint As EndPoint = New IPEndPoint(IPAddress.Loopback, 0)
```

```
    Private MergeInstance As MergeInstanceCallback
```

```
    Public Sub Create(ByVal callback As MergeInstanceCallback)
```

```
        Create(callback, 1048576)
```

```
    End Sub
```

```
    Public Sub Create(ByVal callback As MergeInstanceCallback, ByVal bufferSize As Integer)
```

```
        If callback = Nothing Then Return
```

```
        MergeInstance = callback
```

```
        InstanceKey = Assembly.GetEntryAssembly().GetType().GUID.ToString()
```

```
        PortSave = Registry.CurrentUser.OpenSubKey("AppEvents\Instances", True)
```

```
        If PortSave Is Nothing Then PortSave =
```

```
Registry.CurrentUser.CreateSubKey("AppEvents\Instances")
```

```
        If InstanceKey = Nothing Then Throw New Exception("Invalid InstanceKey")
```

```
        ReDim SocketBuffer(bufferSize)
```

```
        Dim instancePort As Integer = GetCurrentInstancePort()
```

```
        If instancePort = 0 Then
```

```
            CreateNewInstance()
```

```
        Else
```

```
            MergeInstances(instancePort)
```

```
        End If
```

```

End Sub
Private Sub CreateNewInstance()
    If Not applicationMutex Is Nothing Then applicationMutex.ReleaseMutex()

    applicationMutex = New Mutex(True, InstanceKey)

    InstanceSocket = New Socket(AddressFamily.InterNetwork, SocketType.Dgram,
ProtocolType.Udp)
    InstanceSocket.Bind(SenderEndpoint)
    PortSave.SetValue(InstanceKey, DirectCast(InstanceSocket.LocalEndPoint, IPEndPoint).Port)
    InstanceSocket.BeginReceiveFrom(SocketBuffer, 0, SocketBuffer.Length, SocketFlags.None,
SenderEndpoint, AddressOf DataReceived, Nothing)
End Sub
Private Sub MergeInstances(ByVal port As Integer)
    InstanceSocket = New Socket(AddressFamily.InterNetwork, SocketType.Dgram,
ProtocolType.Udp)

    Dim args As String() = Environment.GetCommandLineArgs()

    Using ms As MemoryStream = New MemoryStream()
        Using bw As BinaryWriter = New BinaryWriter(ms)
            bw.Write(args.Length)
            For i = 0 To args.Length - 1
                bw.Write(args(i))
            Next
            InstanceSocket.SendTo(ms.ToArray, New IPEndPoint(IPAddress.Loopback, port))
        End Using
    End Using
    Environment.Exit(0)
End Sub
Private Function GetCurrentInstancePort() As Integer
    Try
        applicationMutex = Mutex.OpenExisting(InstanceKey)
        If PortSave Is Nothing Then Return 0
        Return DirectCast(PortSave.GetValue(InstanceKey, 0), Integer)
    Catch ex As Exception
        Return 0
    End Try
End Function
Private Sub DataReceived(ByVal AR As IAsyncResult)
    Dim bytesRead As Integer = InstanceSocket.EndReceive(AR)

    If bytesRead = 0 Then
        InstanceSocket.BeginReceiveFrom(SocketBuffer, 0, SocketBuffer.Length,
SocketFlags.None, SenderEndpoint, AddressOf DataReceived, Nothing)
        Return
    End If

    Dim data(bytesRead) As Byte
    Buffer.BlockCopy(SocketBuffer, 0, data, 0, data.Length)

    Dim path As String = Nothing
    Dim args() As String = Nothing

```

```

Using ms As MemoryStream = New MemoryStream(data)
Using br As BinaryReader = New BinaryReader(ms)
    ReDim args(br.ReadInt32() - 1)
    path = br.ReadString()
    For i = 0 To args.Length - 1
        args(i) = br.ReadString
    Next
End Using
End Using
MergeInstance(path, args)
InstanceSocket.BeginReceiveFrom(SocketBuffer, 0, SocketBuffer.Length, SocketFlags.None,
SenderEndpoint, AddressOf DataReceived, Nothing)
End Sub
End Class

```

Module1.vb:

Code Module Module1

```
Dim SingleInst As SingleInstance
```

```
Sub Main()
```

```
    SingleInst = New SingleInstance
```

```
    SingleInst.Create(AddressOf OnNewInstance)
```

```
    Console.WriteLine("First instance created")
```

```
    Console.ReadLine()
```

```
End Sub
```

```
Private Sub OnNewInstance(ByVal path As String, ByVal args As String())
```

```
    Console.WriteLine("New instance created. Path: {0}", path)
```

```
    For i = 0 To args.Length - 1
```

```
        Console.WriteLine("Args {0}: {1}", i, args(i))
```

```
    Next
```

```
End Sub
```

End Module

BahNahNah - November 18, 2015, 7:32 pm

(Nov 18, 2015 08:11 PM)Tedd Wrote: [\[CITE\]](#) Ported to vb. I really like this, useful as fuck.

SingleInstance.vb:

-code-

Module1.vb:

-code-

Thanks ;)

I modified it slightly to make it static and added it to the [GIST](#).

Only3 - November 19, 2015, 2:56 am

(Nov 18, 2015 04:17 PM)BahNahNah Wrote: [\[CITE\]](#) This would not work. Did you even try it?
This work. Did you even try it?

BahNahNah - November 19, 2015, 3:00 am

(Nov 19, 2015 03:56 AM)Only3 Wrote: [\[CITE\]](#) This work. Did you even try it?
But it never checks if there is already another process running.

Only3 - November 19, 2015, 3:14 am

bahna open ur brain its easy to do it
also this way u not creating new files all in memory
tip use the power of the args to pass the info

BahNahNah - November 19, 2015, 3:22 am

(Nov 19, 2015 04:14 AM)Only3 Wrote: [\[CITE\]](#) bahna open ur brain its easy to do it
also this way u not creating new files all in memory
tip use the power of the args to pass the info
I tried it.

Spoiler ([Click to View](#))

Only3 - November 19, 2015, 5:26 am

to send arg back to the app u can use sendmessage api

6e756c6c - November 19, 2015, 5:32 am

In Visual Basic you can do this using Application Events (Project Properties -> Application -> "View Application Events"). You just need to override the "OnStartupNextInstance" event and handle the eventArgs. You also need to set "Make single instance application" in Project Properties.

Code Namespace My

Partial Friend Class MyApplication

Protected Overrides Sub OnStartupNextInstance(eventArgs As
ApplicationServices.StartupNextInstanceEventArgs)

'TODO: (Optional) Set eventArgs.BringToFront to true so that the application
activates.

'TODO: Do something with eventArgs.CommandLine

MyBase.OnStartupNextInstance(eventArgs)
End Sub

End Class

End Namespace

Tedd - November 19, 2015, 6:04 am

(Nov 19, 2015 06:32 AM)Aeonhack Wrote: [\[CITE\]](#) In Visual Basic you can do this using Application Events (Project Properties -> Application -> "View Application Events"). You just need to override the "OnStartupNextInstance" event and handle the eventArgs. You also need to set "Make single instance application" in Project Properties.

Code Namespace My

Partial Friend Class MyApplication

```
Protected Overrides Sub OnStartupNextInstance(eventArgs As
ApplicationServices.StartupNextInstanceEventArgs)
    'TODO: (Optional) Set eventArgs.BringToFront to true so that the application
activates.
    'TODO: Do something with eventArgs.CommandLine
```

```
    MyBase.OnStartupNextInstance(eventArgs)
End Sub
```

```
End Class
```

```
End Namespace
```

Fuck I wasted my time porting his code lol I'm dumb. Good share though, thank you.

SIM-1456245 - November 19, 2015, 7:05 am

(Nov 19, 2015 06:32 AM)Aeonhack Wrote: [\[CITE\]](#) In Visual Basic you can do this using Application Events (Project Properties -> Application -> "View Application Events"). You just need to override the "OnStartupNextInstance" event and handle the eventArgs. You also need to set "Make single instance application" in Project Properties.

```
Code Namespace My
```

```
Partial Friend Class MyApplication
```

```
Protected Overrides Sub OnStartupNextInstance(eventArgs As
ApplicationServices.StartupNextInstanceEventArgs)
    'TODO: (Optional) Set eventArgs.BringToFront to true so that the application
activates.
    'TODO: Do something with eventArgs.CommandLine
```

```
    MyBase.OnStartupNextInstance(eventArgs)
End Sub
```

```
End Class
```

```
End Namespace
```

Was just going to post this, I use this for a long time. Because I'm a VB coder, I'm not sure if you have that in C#, if you don't then this might be a good thing, but still I think that there are other workarounds like sendmessage, registry keys or even textfiles. Still good job.

tweety - November 19, 2015, 8:22 pm

(Nov 18, 2015 03:42 PM)BahNahNah Wrote: [\[CITE\]](#) I thought pipelines where .net 4?
I try to make my programs .Net 2.0 when it is possible, so its more compatible with other peoples projects.

Example?

This is the most retarded excuse I've ever heard in my life.

SIM-1456245 - November 20, 2015, 5:50 am

(Nov 19, 2015 09:22 PM)Tweety. Wrote: [\[CITE\]](#) This is the most retarded excuse I've ever heard in

my life.

Honestly it's a good statement, because most of the people here ask for 2.0, because it's HackForums, and most of the people that code stuff here is malware, and malware should be as compatible as possible, so yes, it might be a good idea to try to make classes/libraries 2.0 if you know what it will be used for malware. If it will be used for application that is anything other, should use up-to date frameworks i.e. 4.6

tweety - November 20, 2015, 7:37 am

(Nov 20, 2015 06:50 AM)Electroalek Wrote: [\[CITE\]](#) Honestly it's a good statement, because most of the people here ask for 2.0, because it's HackForums, and most of the people that code stuff here is malware, and malware should be as compatible as possible, so yes, it might be a good idea to try to make classes/libraries 2.0 if you know what it will be used for malware. If it will be used for application that is anything other, should use up-to date frameworks i.e. 4.6

This is the second most retarded thing I've heard in my life.

Washi - November 21, 2015, 5:18 am

(Nov 20, 2015 05:59 PM)Aeonhack Wrote: [\[CITE\]](#) Tweety, one of your major selling points on you're R.A.T Prototype is that it doesn't have .NET dependencies (2.0 specifically) so users have maximum compatibility / install success. As I know you are well aware, .NET 2.0 is still (unfortunately) more common than .NET 4.0 so it makes perfect sense to target 2.0 still. The point being, compatibility is important (as you indicate in your product's thread) and .NET 2.0 is more compatible than 4.0.

?

Is 2.0 still being used that much? I mean, it's over 10 years old. Even windows vista shipped with version 3.0. Besides, the most commonly used operating system is Windows 7 by the time writing this, which has 3.5 pre-installed, and most people buying a new pc would probably be using windows 10 which comes with version 4.6.

I am not saying you're wrong and I know that a lot of people still use windows XP. My point is that to me it sounds rather stupid to rely on ancient versions of .NET framework to get a big audience. I think people should really get their systems up to date if they're still running legacy software. And to be honest, I doubt it is worth targeting such an audience if you're developing an application. But yea that's just my view on that. I am sure a lot of people would disagree >_< !

Sources:

https://en.wikipedia.org/wiki/Usage_share_of_operating_systems#Desktop_and_laptop_computers
https://en.wikipedia.org/wiki/.NET_Framework#Versions

6e756c6c - November 21, 2015, 7:20 am

(Nov 21, 2015 06:18 AM)TheUnknownProgrammer Wrote: [\[CITE\]](#) Is 2.0 still being used that much? I mean, it's over 10 years old. Even windows vista shipped with version 3.0. Besides, the most commonly used operating system is Windows 7 by the time writing this, which has 3.5 pre-installed, and most people buying a new pc would probably be using windows 10 which comes with version 4.6.

I am not saying you're wrong and I know that a lot of people still use windows XP. My point is that to me it sounds rather stupid to rely on ancient versions of .NET framework to get a big audience. I think people should really get their systems up to date if they're still running legacy software. And to be honest, I doubt it is worth targeting such an audience if you're developing an application. But yea that's just my view on that. I am sure a lot of people would disagree >_< !

Sources:

https://en.wikipedia.org/wiki/Usage_share_of_operating_systems#Desktop_and_laptop_computers

https://en.wikipedia.org/wiki/.NET_Framework#Versions

Your link says 30% of Chinese computers still use Windows XP, which do not ship with 3.0 or higher. I don't know how many Chinese people have computers, but I imagine it's a pretty substantial number to simply ignore.

I'm not arguing that people shouldn't move on to higher framework and just force customers to upgrade (that's the route I'm taking with .NET Seal 3). But still, there is obviously still a demand for .NET 2.0 compatible code.

SIM-1456245 - November 21, 2015, 9:09 am

To probably end the discussion, we all know that it's REALLY not needed at all to go down to 2.0, but what we also all know is that saying "This RAT is 2.0", pokes you in the eye. That is called a marketing trick. We all know that most of the people that buy malware are not experienced enough to understand the usage of frameworks, all they know is that they have read that 2.0 is more compatible, and that's it, it sells it to them. That's why this idea of 2.0 compatibility is used only in malware, and not in ANY other application type.

Josh? - July 26, 2016, 6:14 am

Hey BahNahNah, I know this is fairly old, but would this work in a ClickOnce application?

HFDone - July 26, 2016, 12:55 pm

(Jul 26, 2016 06:14 AM)Josh? Wrote: [\[CITE\]](#) Hey BahNahNah, I know this is fairly old, but would this work in a ClickOnce application?

hey J the easy short fast best answer will be trying it your self

Ninja - July 26, 2016, 6:50 pm

It's nice to see that you contributed so many original things to the forums lately. Great job BahNahNah, and for once I mean it.