

Forum:[Hack Forums](#)

Thread: [Using Window's URI Scheme.](#)

SubGenius - November 13, 2014, 2:34 am

Hi,

I am going to show you how get advantage of Window's URI scheme so you can easily start your application from a website or pass arguments to it.

It's the same method that torrent clients use to open Magnet links, skype and Battlefield.

Here is an example:

Skills required:

Basic Knowledge of vb.NET

Know how to open the windows registry and how it works.

Some HTML knowledge.

Difficulty level: 3/10?

So, let's start:

First we have to create a key in the registry.

Open the start menu -> type regedit

Create:Code HKEY_CLASSES_ROOT -> yourcommand -> shell -> open -> commandand change the default value of command into this:

Code "C:\yourappsdir\app.exe" "%1" You can create this key in setup or when the user starts the application for first time.

>Have a look [Here](#) on how to create Registry Keys with VB.net

This is how the registry key should look like:

As you probably understood the URI scheme requires an executable location.

This means if your application is portable you should change the location of the registry each time you start your application.

Next, the website files.

Nothing special, some simple HTML tags and a javascript function:

Code:

PHP Code

<?php

<script type='text/javascript'>

```
function SendData(data) {  
    command = 'cCommand:'  
    redirecturl = command+data;  
    window.open(redirecturl,"_self")  
}
```

</script>

```
<input type='text' name='text' id='text' />
<input type='button' value='Send Data' onclick=
'javascript:SendData("-text="+ document.getElementById("text").value);' />
<input type='button' value='Clear Textbox' onclick='javascript:SendData("-clear");' />
```

I have no idea of web programming, searching for sources on the web made it possible.
Upload this to your webserver and you are ready to test it.

Now the application:

First, make your application single instance:

Go to Project Properties -> Application -> check Make single instance application

This will prevent your application from opening multiple times.

Then for this tutorial just add a textbox into your main form,

and this into the code:

Code Imports Microsoft.VisualBasic.ApplicationServices

Public Class frmMain

'Check for Arguments after the application has started.

```
Public Sub app_NextInstance(ByVal sender As Object, ByVal e As
ApplicationServices.StartupNextInstanceEventArgs)
    CheckForArguments(cCollectionToArray(e.CommandLine))
End Sub
```

'Check for Arguments on start up.

```
Private Sub app_StartUp(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
    CheckForArguments(Environment.GetCommandLineArgs)
End Sub
```

'Here you can include whatever u like, it's simple to check for arguments and parse it's data.

```
Sub CheckForArguments(ByVal Arguments As String())
```

```
For Each Argument As String In Arguments
```

```
If Argument.Contains("-text=") Then
```

```
    TextBox1.AppendText(Argument.Split("=")(1) & vbNewLine)
```

```
Elseif Argument.Contains("-clear") Then
```

```
    TextBox1.Clear()
```

```
End If
```

```
Next
```

```
End Sub
```

'Just a simple function that converts a collection of string into an array of strings, nothing to worry

```
Private Function cCollectionToArray(ByVal Collection As
System.Collections.ObjectModel.ReadOnlyCollection(Of String)) As String()
    Dim Array(Collection.Count - 1) As String
    For i As Integer = 0 To (Collection.Count - 1)
        Array(i) = Collection.Item(i)
```

```
Next
Return Array
End Function
End Class
Next,
Go to Project Properties -> Application -> View Application Events
```

and add a new sub:

```
Code Sub App_NextInstance(ByVal sender As Object, ByVal e As
ApplicationServices.StartupNextInstanceEventArgs) Handles Me.StartupNextInstance
    frmMain.app_NextInstance(sender, e)
End Sub
And you are done.
```

If you haven't understand what's going on until now I will explain you:

What we do is register a special(?) command into the registry that allows us to start and parse arguments to our application from websites, or other applications.

You can also use CMD to do the same:
example:

More links:

Download The Project Here: yet unavailable

[How to use command lines](#)

[URI Scheme on MSDN](#)

[How to create registry keys within vb apps](#)

Nikey646 - November 13, 2014, 3:56 am

Just a heads up for anyone who plans on using this, it will only work if you have the Microsoft.VisualBasic namespace imported (Horrible practice, in any language other than VB), and won't work in WPF Based applications (It does not have the SingleInstance option you use)

The alternative to this, is to use the same Registry with NamedPipeStreams (If using a language outside of VB), and Mutex (If using WPF). I've previously posted an example of both, using Threading to make sure reading the new arguments does not freeze the application. For anyone interested: hackforums.net/showthread.php?tid=4470491

Also, just a hint: The registry stuff should be done upon installation, not runtime.

Besides the reliance on the Microsoft.VisualBasic namespace, good tutorial!

Ajumma - November 13, 2014, 10:01 am

This is pretty cool!

I always wanted to run my game from my website kinda like other games out there, and now I can.

SubGenius - November 13, 2014, 6:03 pm

(Nov 13, 2014 04:56 AM)Nikey646 Wrote: [\[CITE\]](#) Just a heads up for anyone who plans on using this, it will only work if you have the Microsoft.VisualBasic namespace imported (Horrible practice, in any language other than VB), and won't work in WPF Based applications (It does not have the SingleInstance option you use)

The alternative to this, is to use the same Registry with NamedPipeStreams (If using a language

outside of VB), and Mutex (If using WPF). I've previously posted an example of both, using Threading to make sure reading the new arguments does not freeze the application. For anyone interested: hackforums.net/showthread.php?tid=4470491

Also, just a hint: The registry stuff should be done upon installation, not runtime.

Besides the reliance on the Microsoft.VisualBasic namespace, good tutorial!

Nice addition.

Yeah, I mentioned you could create the registry key during installation or first run.

All I wanted to do in this tut is to pass the method and the concept.

Earn - November 13, 2014, 11:49 pm

Great tutorial. Finally something unique that I haven't seen posted. Good work.

?Salty? - November 14, 2014, 12:37 am

Damn. Great tutorial, OP.

Nice to see you contributing.

Expect green apples from me, when I can. ?