

Forum:[Hack Forums](#)

Thread: [Source - How to Handle/Create URL Protocols](#)

SomeWhiteGuy - July 19, 2012, 7:39 pm

I've made a very simple source that creates a URL Protocol and I've included an app with it that can be used as the application that is opened when the protocol is navigated to.

URL Protocols are stored in the registry, so this would be something you should install when your app is in the installation stage, not after imho.

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

[Video: http://www.youtube.com/watch?v=b_VM67SkVao]

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

[Click meh.](#)

Like always, the project file was created in VS 11. If you do not have it, just open up Module1.vb in NP++ or just regular NP.

I've never worked with the registry before, so if there was an easier way to do all the shit I did, feel free to let me know.

I suggest reading the [security issues](#) pointed out on MSDN about handling these protocols and what can fuck up/make your app vulnerable to malicious shit.

Thanks for downloading and being interested! Have fun!

Eternal Five - July 23, 2012, 3:24 pm

This is very interesting, thanks for sharing!

SomeWhiteGuy - July 23, 2012, 4:29 pm

(Jul 23, 2012 03:24 PM)Eternal Five Wrote: [\[CITE\]](#) This is very interesting, thanks for sharing! Glad that SOMEONE found some use for it, lol.

Thanks for the comment.

Aeonhack Archivist - July 23, 2012, 5:57 pm

Thanks for the share. Ever since the Nexus client (mod installer for TES/Fallout/etc) came out, I have been using something very similar to this. Thanks for the share because this is much more simplified than I had been making it.

Fake. - July 23, 2012, 7:58 pm

If you'd like something a little shorter, I modified your function slightly:

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

```
Code Private Function CreateURLProtocol(protocol As String, displayName As String,
```

```

assemblyName As String, filePath As String) As Boolean
Try
    Dim regKey As Microsoft.Win32.RegistryKey =
Microsoft.Win32.Registry.ClassesRoot.CreateSubKey(protocol)
    With regKey
        'Set top-level values.
        .SetValue(String.Empty, "URL:" & displayName)
        .SetValue("URL Protocol", String.Empty)

        'Create the relevant subkeys.
        .CreateSubKey("URL Protocol")
        .CreateSubKey("DefaultIcon").SetValue(String.Empty, assemblyName & ".exe,1")
        .CreateSubKey("shell")
        .CreateSubKey("shell\open\command").SetValue(String.Empty, String.Format("""{0}"" ""%1""",
filePath))
        .Close()
    End With
    Return True
Catch ex As Exception
    Return False
End Try
End Function

```

There is no need to be declaring several registry keys anyway, you could've just reused the first one considering you never come back to it later on.

Edit: Only just saw the date of the OP >.>