Debugging Everything
Code and other everyday things

Custom Protocol Handler (CPH)

By smike19 | June 21, 2019 1 Comment

Microsoft Windows supports registered custom protocols additionally to the common ones such as http, https, ftp, mailto and so on. To register an application to handle a particular URI scheme, new key, along with the appropriate subkeys and values should be added to the Windows registry. The new key The may be added to HKEY_LOCAL_MACHINE\Software\Classes and applied to all users on the local computer or HKEY_CURRENT_USER\Software\Classes that is applied only to the interactive user. Presented below c# console application is doing custom protocol registration for current user, unregistration and may be used for testing.

The code:

```
using System;
using Microsoft.Win32;
using System.Windows.Forms;
using System.Diagnostics;
namespace cph
{
   class Program
   {
      static void Main(string[] args)
      {
```

Search

Recent Posts

snmp++v3 crashes after version upgrade.

Windows Subsystem for Linux. Accessing files.

Windows Subsystem for Linux. Installation Linux Distributions.

Categories

Select Category

Meta

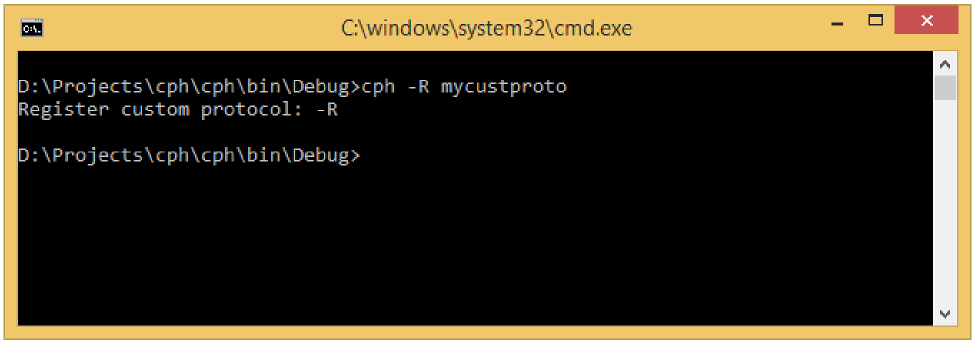
Log in

Entries RSS

Comments RSS

WordPress.org

```
         if(args.Length == 0)
         {
            Console.WriteLine(“Start application with one of following
arguments:\n” +
               “\”-R \” – to register new custom protocol\n” +
               “\”-D \” – to unregister custom protocol\n” +
               “\”-T\” – to test custom protocol”);
            return;
         }
         if(args[0].ToUpper() == “-R”)
         {
            if(args.Length < 2)
            {
               Console.WriteLine(“Custom protocol name is not specified”);
            } else
            {
               Console.WriteLine(“Register custom protocol: ” + args[0]);
               RegistryKey key =
Registry.CurrentUser.OpenSubKey(@”Software\Classes”, true);
               RegistryKey keyCP = key.CreateSubKey(args[1],
RegistryKeyPermissionCheck.ReadWriteSubTree);
               keyCP.SetValue(“”, String.Format(“URL:{0}
Protocol”,args[1]));
               keyCP.SetValue(“URL Protocol”, “”);
               RegistryKey keyIcon = keyCP.CreateSubKey(“DefaultIcon”);
               keyIcon.SetValue(“”,
Process.GetCurrentProcess().ProcessName+”.exe,1″ );
               RegistryKey keyCPshell = keyCP.CreateSubKey(“shell”,
RegistryKeyPermissionCheck.ReadWriteSubTree);
               RegistryKey keyCPopen = keyCPshell.CreateSubKey(“open”,
RegistryKeyPermissionCheck.ReadWriteSubTree);
               RegistryKey keyCPcommand = keyCPopen.CreateSubKey(“command”,
RegistryKeyPermissionCheck.ReadWriteSubTree);
               keyCPcommand.SetValue(“”, “\”” +
System.Reflection.Assembly.GetExecutingAssembly().Location + “\” \”%1\””);
            }
               return;
            }
         if (args[0].ToUpper() == “-D”)
         {
            if (args.Length < 2)
            {
               Console.WriteLine(“Custom protocol name is not specified”);
            }
            else
            {
               Console.WriteLine(“Unregister custom protocol: ” + args[0]);
               try
               {
                  RegistryKey key =
Registry.CurrentUser.OpenSubKey(@”Software\Classes”, true);
                  key.DeleteSubKeyTree(args[1]);
            } catch(Exception ex)
               {
                  Console.WriteLine(ex.Message);
               }
            }
            return;
         }
         if (args[0].ToUpper().Contains(“:-T”))
         {
            Console.WriteLine(“Test custom protocol, argument: ” + args[0]);
            MessageBox.Show(“Close me!”);
            return;
         }
      }
   }
}
```

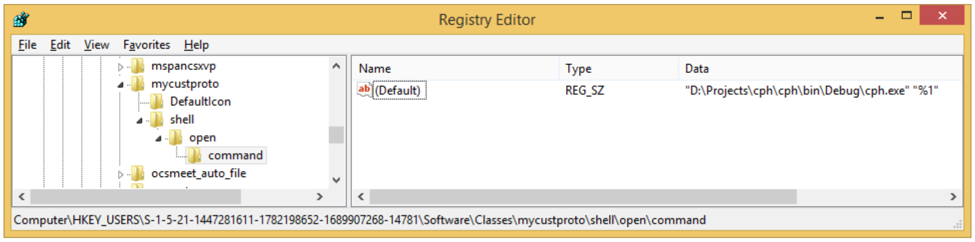


```
D:\Projects\cph\cph\bin\Debug>cph -R mycustproto
Register custom protocol: -R

D:\Projects\cph\cph\bin\Debug>
```

New registry entries looks like:

```
Windows Registry Editor Version 5.00
[HKEY_USERS\S-1-5-21-1447281611-1782198652-1689907268-
14781\Software\Classes\mycustproto]
@=”URL:mycustproto Protocol”
“URL Protocol”=””
[HKEY_USERS\S-1-5-21-1447281611-1782198652-1689907268-
14781\Software\Classes\mycustproto\DefaultIcon]
@=”cph.exe,1″
[HKEY_USERS\S-1-5-21-1447281611-1782198652-1689907268-
14781\Software\Classes\mycustproto\shell]
[HKEY_USERS\S-1-5-21-1447281611-1782198652-1689907268-
14781\Software\Classes\mycustproto\shell\open]
[HKEY_USERS\S-1-5-21-1447281611-1782198652-1689907268-
14781\Software\Classes\mycustproto\shell\open\command]
@=”\”D:\\Projects\\cph\\cph\\bin\\Debug\\cph.exe\”
\”%1\””
```



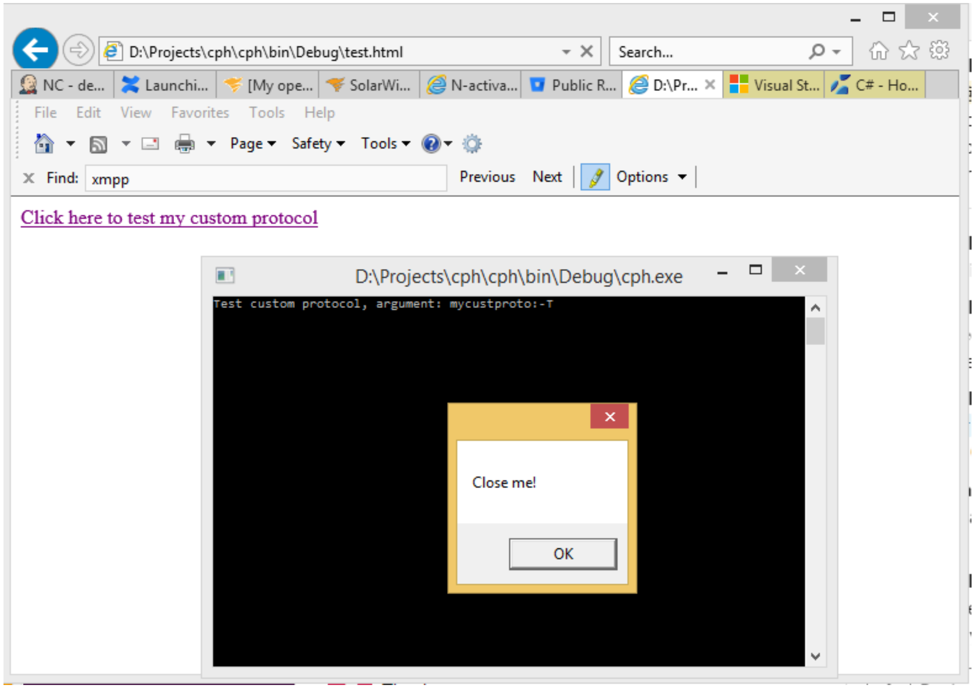
Name	Type	Data
(Default)	REG_SZ	"D:\Projects\cph\cph\bin\Debug\cph.exe" "%1"

Computer\HKEY_USERS\S-1-5-21-1447281611-1782198652-1689907268-14781\Software\Classes\mycustproto\shell\open\command

Step two, using notepad or other text editor create html file:

```
<html>
<body>
<a href=”mycustproto:-T”>Click here to test my custom protocol</a>
</body>
</html>
```

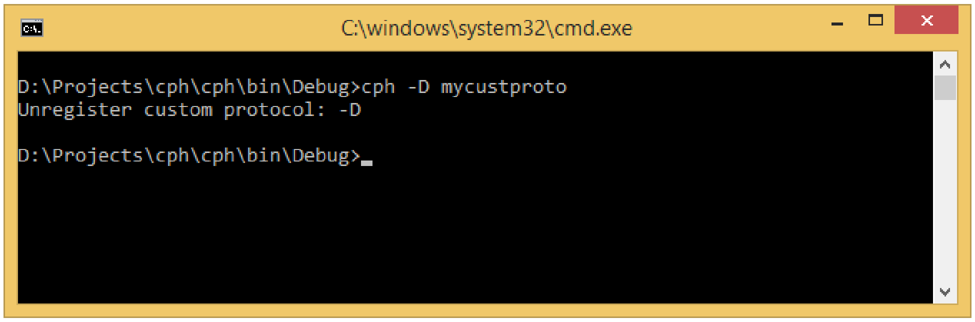
Step three, testing: open html file created in step two in IE, Chrome and Firefox browser and click on “Click here to test my custom protocol” link:



```
Test custom protocol, argument: mycustproto:-T
```

Close me!

Step four, unregister mycustproto custom protocol (removing entries from registry):



```
D:\Projects\cph\cph\bin\Debug>cph -D mycustproto
Unregister custom protocol: -D

D:\Projects\cph\cph\bin\Debug>_
```

To enable/disable warning dialog before opening application related to this CPH, change value of WarnOnOpen to 1 or 0 respectively in Windows registry.

```
Windows Registry Editor Version 5.00

[HKEY_CURRENT_USER\Software\Microsoft\Internet
Explorer\ProtocolExecute\mycustproto]
“WarnOnOpen”=dword:00000001
```

Category: Coding How to Windows

← Old hardware with new OS   How to Increase CPU Usage →

One thought on “Custom Protocol Handler (CPH)”

Sasha
December 11, 2019

The argument of customer protocol handler cannot be more than 2K characters

Reply ↓

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

Post Comment