

Forum: [Hack Forums](#)

Thread: [Function to add proxy support for TWebBrowser](#)

tcsn - October 9, 2017, 12:28 pm

The following allows you to use proxy's in your TWebBrowser component when using it in your projects.

Function source code:

```
Code {$DEFINE DELPHI2009_UP}
function SetProxy(Server: String): Boolean;
// Server z.B. '127.0.0.1:8080' oder "
type
  INTERNET_PER_CONN_OPTION = record
    dwOption: DWORD;
    Value: record case Integer of 1: (dwValue: DWORD);
    2: (pszValue: {$IFDEF DELPHI2009_UP}PWideChar{$ELSE}PAnsiChar{$ENDIF});
    3: (ftValue: TFileTime);
  end;
end;

end;
LPINTERNET_PER_CONN_OPTION = ^INTERNET_PER_CONN_OPTION;
INTERNET_PER_CONN_OPTION_LIST = record dwSize: DWORD;
pszConnection:
LPTSTR;
dwOptionCount:
DWORD;
dwOptionError:
DWORD;
pOptions:
LPINTERNET_PER_CONN_OPTION;
end;
LPINTERNET_PER_CONN_OPTION_LIST = ^INTERNET_PER_CONN_OPTION_LIST;

const
  INTERNET_PER_CONN_FLAGS = 1;
  INTERNET_PER_CONN_PROXY_SERVER = 2;
  INTERNET_PER_CONN_PROXY_BYPASS = 3;
  INTERNET_PER_CONN_AUTOCONFIG_URL = 4;
  INTERNET_PER_CONN_AUTODISCOVERY_FLAGS = 5;
  PROXY_TYPE_DIRECT = $00000001;
  PROXY_TYPE_PROXY = $00000002;
  PROXY_TYPE_AUTO_PROXY_URL = $00000004;
  PROXY_TYPE_AUTO_DETECT = $00000008;
  INTERNET_OPTION_REFRESH = 37;
  INTERNET_OPTION_PER_CONNECTION_OPTION = 75;
  INTERNET_OPTION_SETTINGS_CHANGED = 39;

var
```

```
OptionsList: INTERNET_PER_CONN_OPTION_LIST;  
BufSize: DWORD;  
HInternet: Pointer;  
Agent: String;
```

```
begin
```

```
    Result := False;  
    BufSize := SizeOf(OptionsList);  
    OptionsList.dwSize := BufSize;  
    OptionsList.pszConnection := nil;  
    OptionsList.dwOptionCount := 3;  
    OptionsList.pOptions := AllocMem(3 * SizeOf(INTERNET_PER_CONN_OPTION));  
    try
```

```
        if not Assigned(OptionsList.pOptions) then  
            EXIT;
```

```
        OptionsList.pOptions^.dwOption := INTERNET_PER_CONN_FLAGS;  
        OptionsList.pOptions^.Value.dwValue := PROXY_TYPE_DIRECT or  
            PROXY_TYPE_PROXY;
```

```
        inc(OptionsList.pOptions);
```

```
        OptionsList.pOptions^.dwOption := INTERNET_PER_CONN_PROXY_SERVER;  
        OptionsList.pOptions^.Value.pszValue := PChar(Server);
```

```
        inc(OptionsList.pOptions);
```

```
        OptionsList.pOptions^.dwOption := INTERNET_PER_CONN_PROXY_BYPASS;  
        OptionsList.pOptions^.Value.pszValue := 'local';
```

```
        dec(OptionsList.pOptions, 2);
```

```
        Agent := ExtractFileName(Application.ExeName);
```

```
        HInternet := InternetOpen
```

```
            ({$IFDEF DELPHI2009_UP}PWideChar{$ELSE}PAnsiChar{$ENDIF}
```

```
            (Agent), INTERNET_OPEN_TYPE_DIRECT, nil, nil, 0);
```

```
        try
```

```
            Result := InternetSetOption(HInternet,
```

```
            INTERNET_OPTION_PER_CONNECTION_OPTION, @OptionsList, BufSize);
```

```
            InternetSetOption(HInternet, INTERNET_OPTION_REFRESH, nil, 0);
```

```
        finally
```

```
            InternetCloseHandle(HInternet);
```

```
        end;
```

```
    finally
```

```
        FreeMem(OptionsList.pOptions);
```

```
    end;
```

```
end;
```

Now just make sure to call the above function before using `WebBrowser.Navigate` , like this:

```
Code begin
```

```
    SetProxy('ip:port');
```

```
    WebBrowser1.Navigate('https://hackforums.net');
```

```
end;
```

or if you want to load a specific proxy from a specific line located, for example, in a file named `proxy.txt` in your application main dir, use this:

```
Code var
```

```
proxy: TStringList;
```

```
begin
```

```

proxy := TStringList.Create;
proxy.LoadFromFile('proxy.txt');
SetProxy(proxy.Strings[0]); //will load the proxy from the 1st line in the .txt
proxy.Free;
end;
That's it.

```

The component will set the specified proxy, either from the list or directly, in accordance to the route you chose to use, then it will navigate to the specified url. Obviously you can use the function on multiple instances of TWebBrowser, provided they're on your form or self-created. Just attach a new SetProxy(); to each instance and specify the proxy details (ip:port or line from file).

Known issues:

When trying to display, for example, captcha (or more recently, reCaptcha) on a TWebBrowser component in your application, it will most likely fail due to compatibility problems, as the component is using the old IE.

This has an easy fix:

Drop a TButton and name it Registry.
Double click it and add the following code:

```

Code procedure TForm1.RegistryClick(Sender: TObject);
var
Registry: TRegistry;

begin
    Registry := TRegistry.Create;
    Registry.RootKey := HKEY_CURRENT_USER;
    Registry.OpenKey('Software\Microsoft\Internet
Explorer\Main\FeatureControl\FEATURE_BROWSER_EMULATION', TRUE);
    Registry.WriteInteger('YourApplicationExeName', 11001);
    Registry.Free;
    ShowMessage('Registered!');
end;

```

Just replace YourApplicationExeName with the name of your application.

Once you run the application for the first time, click the button named Registry. This will create a registry entry on the current PC (doesn't require Administrator rights) and registers the TWebBrowser component to use the latest IE (11).

Obviously the above code needs to be ran only once per system - since it creates a registry entry you don't need to do it every single time you use the application.

Voila, you're browsing with Edge ?

Bonus:

Instantly empty TWebBrowser cache, cookies, sessions and everything else. Useful to clear everything once you've used it to navigate and so on.

```

Code var
  lpEntryInfo: PInternetCacheEntryInfo;
  hCacheDir: LongWord;
  dwEntrySize: LongWord;
begin
  dwEntrySize := 0;
  FindFirstUrlCacheEntry(nil, TInternetCacheEntryInfo(nil^), dwEntrySize);
  GetMem(lpEntryInfo, dwEntrySize);
  if dwEntrySize > 0 then lpEntryInfo^.dwStructSize := dwEntrySize;
  hCacheDir := FindFirstUrlCacheEntry(nil, lpEntryInfo^, dwEntrySize);
  if hCacheDir <> 0 then begin
    repeat
      // delete cookies
      if ((lpEntryInfo^.CacheEntryType and COOKIE_CACHE_ENTRY) = COOKIE_CACHE_ENTRY)
      // delete history
      or ((lpEntryInfo^.CacheEntryType and URLHISTORY_CACHE_ENTRY) =
URLHISTORY_CACHE_ENTRY)
      // delete "normal" cache
      or ((lpEntryInfo^.CacheEntryType and NORMAL_CACHE_ENTRY) =
NORMAL_CACHE_ENTRY)
      then DeleteUrlCacheEntry(lpEntryInfo^.lpszSourceUrlName);
      FreeMem(lpEntryInfo, dwEntrySize);
      dwEntrySize := 0;
      FindNextUrlCacheEntry(hCacheDir, TInternetCacheEntryInfo(nil^), dwEntrySize);
      GetMem(lpEntryInfo, dwEntrySize);
      if dwEntrySize > 0 then lpEntryInfo^.dwStructSize := dwEntrySize;
    until not FindNextUrlCacheEntry(hCacheDir, lpEntryInfo^, dwEntrySize);
  end;
  FreeMem(lpEntryInfo, dwEntrySize);
  FindCloseUrlCache(hCacheDir);
end;
Use the code in a procedure and call it once you're done.
This is useful for clearing everything.

```

For example: if you run your application and use it to Navigate and log in to say, a Facebook account, then just close the application and relaunch it again, you're still logged in. Similarly, it's independent from the IE installed on your system so clearing that doesn't help.

Using this code once you've completed everything ensures a fresh session with no trace on next launch. Obviously, there are others ways also, this is just one of many. Good luck ?

BitmasterXor - October 9, 2017, 3:13 pm

Very Nice example tcsh :)

I always tend to use the Twebbrowser for simple tasks that do not require plugins and stuff since it does as you also stated "Runs off the old IE browser engine"....

If anyone wants to simplify the use of proxies in a browser component you can simply install the "CEF4Delphi" Library / Delphi VCL web browser component which is based off of "Chromium Browser" it has proxy support right out of the box no source code required simply edit the properties of the Browser component and run your Delphi project.

tcsh - October 9, 2017, 9:40 pm

(Oct 9, 2017 03:13 PM)BitmasterXor Wrote: [\[CITE\]](#) Very Nice example tcsh :)

I always tend to use the Twebbrowser for simple tasks that do not require plugins and stuff since it does as you also stated "Runs off the old IE browser engine"....

If anyone wants to simplify the use of proxies in a browser component you can simply install the "CEF4Delphi" Library / Delphi VCL web browser component which is based off of "Chromium Browser" it has proxy support right out of the box no source code required simply edit the properties of the Browser component and run your Delphi project.

Yup, there are numerous browser components that have integrated proxy support. For some reason though, people still prefer the original TWebBrowser which does not. Depends on the situation, in certain projects this one is better than one derived from Chromium and so forth. (I actually recently had a project that required this and this was my fix)

Should mention this approach does NOT interfere with the system-wide proxy, meaning it does not change proxy settings in your other browsers that are present on the PC

?

BitmasterXor - October 10, 2017, 11:46 am

(Oct 9, 2017 09:40 PM)tcsh Wrote: [\[CITE\]](#) Yup, there are numerous browser components that have integrated proxy support. For some reason though, people still prefer the original TWebBrowser which does not. Depends on the situation, in certain projects this one is better than one derived from Chromium and so forth. (I actually recently had a project that required this and this was my fix)

Should mention this approach does NOT interfere with the system-wide proxy, meaning it does not change proxy settings in your other browsers that are present on the PC

?

Dude that is AMAZING then :) i know exactly what you mean when it comes to messing with the system wide proxy settings it is a very "Primitive" thing to end up having to do so thank you for this share man :) NOW i will Definitely look into this code and probably start using it with the original Twebbrowser component LOL.

Thanks again man, Keep Delphi going Strong.

tcsh - October 10, 2017, 12:10 pm

Yup, that's one of the main things. You can literally drop 10 TWebBrowser's at design time on the form and set a proxy for each. You then run the application and voila, 10 browsers using 10 different proxies simultaneously while your other browsers installed on the pc aren't affected by any of those proxies (i.e they don't interfere).

Good luck with it and have fun ^_^

BitmasterXor - October 10, 2017, 1:05 pm

(Oct 10, 2017 12:10 PM)tcsh Wrote: [\[CITE\]](#) Yup, that's one of the main things. You can literally drop 10 TWebBrowser's at design time on the form and set a proxy for each. You then run the application and voila, 10 browsers using 10 different proxies simultaneously while your other browsers installed on the pc aren't affected by any of those proxies (i.e they don't interfere).

Good luck with it and have fun ^_^

Dude as an idea.... there is a Software called "Mass Video Blaster" which people currently use to market their products on youtube and stuff with.... man using multiple Twebbrowsers with multiple

proxy support would enable you to make your very own version of this type of software lol.... Delphi is so amazing man :)

Xogg - October 11, 2017, 8:14 am

Thread looks nice! But I would just use a Mozilla browser component instead though, only because TWebBrowser is a Shiternet Explorer browser based component.

Otherwise gg, you made it work.

tcsh - October 11, 2017, 8:33 am

(Oct 11, 2017 08:14 AM)Dexuh Wrote: [\[CITE\]](#) Thread looks nice! But I would just use a Mozilla browser component instead though, only because TWebBrowser is a Shiternet Explorer browser based component.

Otherwise gg, you made it work.

You can make it run on the latest (Edge) - see the registry code section I added. There are a lot of people that still use IE/Edge ^_^

Xogg - October 11, 2017, 8:44 am

(Oct 11, 2017 08:33 AM)tcsh Wrote: [\[CITE\]](#) You can make it run on the latest (Edge) - see the registry code section I added. There are a lot of people that still use IE/Edge ^_^
Buttttt edge is only on windows 10.

tcsh - October 11, 2017, 9:34 am

(Oct 11, 2017 08:44 AM)Dexuh Wrote: [\[CITE\]](#) Buttttt edge is only on windows 10.
Pick the IE that works for you. There's a code there (11001 in my above code) that specifies with which version it should be compatible. ?

BitmasterXor - October 11, 2017, 10:50 am

(Oct 11, 2017 08:33 AM)tcsh Wrote: [\[CITE\]](#) You can make it run on the latest (Edge) - see the registry code section I added. There are a lot of people that still use IE/Edge ^_^
This is very true alot of people really do still use IE and Edge browsers. Thanks for the information and the code samples. I will definitely be looking into this.

tcsh - October 11, 2017, 11:03 am

<https://msdn.microsoft.com/library/ee330730.aspx>

Browser emulation codes here, for anyone interested :D