

Forum:[Hack Forums](#)

Thread: [AutoUpdate.cs | Automatically update your application+++](#)

gigajew - March 18, 2018, 7:02 pm

This is a more 'stable' auto-updater. It doesn't rely on timing or batch scripts, it simply renames itself and downloads the new application from the server, then restarts itself using Application.Restart (which still has the cached filename). Remember to reference System.Windows.Forms.dll.

Usage:

Code using System;
using HackForums.gigajew;

```
namespace code_generator
{
    class Program
    {
        static void Main(string[] args)
        {
            AutoUpdate.OnUpdateStatus += (Double percentageDone) => Console.WriteLine("Done: "
+ percentageDone + "%");
            AutoUpdate.Update("https://the.earth.li/~sgtatham/putty/latest/w32/putty.exe");
        }
    }
}
```

glitchgxd - March 19, 2018, 6:39 am

Instead of renaming the current to an ".old", download to a ".download.tmp" first, validate the file after download completion, and then do the swap. That way you cover events where the user might drop connection or something.

Otherwise, looks clean.

Ninja - March 19, 2018, 9:23 am

I have a few remarks on the code:

Why do this:

Code // parse the response

```
HttpWebResponse response = default(HttpWebResponse);
try
{
    response = request.GetResponse() as HttpWebResponse;
}
catch (WebException exception)
{
    response = exception.Response as HttpWebResponse;
}
```

Instead of:

Code using(HttpWebResponse response = request.GetResponse() as HttpWebResponse)

```
{  
    //code  
}
```

- You don't have to set the request.Proxy property to default(WebProxy). Null is a perfectly fine value (and according to some can speed up the requests a little bit as well).
- Why not use the keywords for primitive types? I keep seeing "Double" and "Int32" and stuff like that, but why not just use "double", "int", etc.
- The default WebRequest.Method value is GET, so no need to set it again.
- I see you check if the status is in the "OK spectrum" as you call it. If a web server wants to let you know all went fine it will just use a OK 200 status code. So you can just check for that. This whole line of code gives me aids:
Code if (Math.Floor(((Double)((Double)response.StatusCode) / 100)) == 2.0)- Why use Latin-1 over UTF-8?

gigajew - March 19, 2018, 10:34 am

default(WebProxy) yields null and is technically a more correct way of nullifying values. Also as far as the try-catch goes, if you call WebRequest.GetResponse() and it has another status code than 200 it will throw an exception and you can instead collect the response object from the catch. It's really odd behavior.

Also the getEncoding is not actually used and could be omitted from the code. About 4-5% of websites use it, where others use UTF-8

gigajew - March 19, 2018, 10:40 am

(Mar 19, 2018 10:35 AM)Orka Wrote: [\[CITE\]](#) How come you're forcing a garbage collection as part of the restart? Your file/network handling doesn't allow for trailing unmanaged resources.
It's probably just a mistake. I can't remember

n0\$3ratu\$ - March 19, 2018, 12:44 pm

What if the application has dependencies?
What if those dependencies are placed in different folders?

Ninja - March 19, 2018, 2:24 pm

(Mar 19, 2018 10:34 AM)gigajew Wrote: [\[CITE\]](#) Also as far as the try-catch goes, if you call WebRequest.GetResponse() and it has another status code than 200 it will throw an exception and you can instead collect the response object from the catch. It's really odd behavior.

I know, I never said you should not use a try catch. I was wondering as to why you did not use a using statement. I like to use an extension method that has the try catch in it to return the WebResponse object no matter if an exception was thrown. Then you could just do:

Code using(HttpWebResponse response = request.GetResponse(ignoreException: true) as HttpWebResponse)
{
 //code

}I use this mainly when I am expecting an exception, as some web servers rely heavily on status codes, and return codes that would normally throw an exception on for example incorrect login credentials.

(Mar 19, 2018 10:34 AM)gigajew Wrote: [\[CITE\]](#) default(WebProxy) yields null and is technically a more correct way of nullifying values

How is it technically more correct? The only way I see it being 'better' is when using generics.

Aristocles - March 19, 2018, 3:47 pm

Cool thank you. Ever thought about a smart update process ? Instead of downloading the updated

file as a whole,
maybe you can download only the new byte patterns of the file in hex. I'm not sure I'm being very clear here but
that would make the updating process quicker.

Ninja - March 19, 2018, 5:14 pm

(Mar 19, 2018 03:47 PM)Martin Heidegger Wrote: [\[CITE\]](#) Cool thank you. Ever thought about a smart update process ? Instead of downloading the updated file as a whole,
maybe you can download only the new byte patterns of the file in hex. I'm not sure I'm being very clear here but
that would make the updating process quicker.

If I understand you correctly then it is the same as the "smart" remote desktop thread not too long ago, where a user only wanted to send the changed bytes over. If I recall correctly he used XOR or something. So it is definitely doable.

Aristocles - March 19, 2018, 5:27 pm

(Mar 19, 2018 05:14 PM)Ninja Wrote: [\[CITE\]](#) If I understand you correctly then it is the same as the "smart" remote desktop thread not too long ago, where a user only wanted to send the changed bytes over. If I recall correctly he used XOR or something. So it is definitely doable.

You got me right