

I don't "get" how a program can update itself. How can I make my software update?

Asked 12 years, 11 months ago Modified 3 years, 11 months ago Viewed 29k times



Say I make an .exe file and everything is peachy. Wonderful it works.

30



Say I worked on a new feature on the software and I want it to be available for people who already have the older version, how can I make the software find my new version, patch it, and then go about it's business.



I can't seem to wrap my head around the issue.



Thank you.

EDIT: I'm sorry for the confusion, but I was meaning a more code-wise answer. Is there something special in my code I should have to allow updating?

For example, if I want to add a new feature, how can I add a "method" to an already packaged .exe? :S That has me in a swivel.

c# upgrade patch

Share Improve this question

edited Nov 17, 2009 at 12:44

asked Nov 17, 2009 at 12:36

Follow



Sergio Tapia

38.9k 75 180 254

- 4 You can't add a method to an already packaged .exe, .dll, or anything else. You just replace the whole file with a newer version. There are tools that can compare the older/newer versions of the file and generate a "patch" which contains only the changes, but that only helps to decrease the download size and few people today bother with that. The end result is still the same as if you had simply copied the newer file over the old one. – Vilx- Nov 17, 2009 at 14:24

9 Answers

Sorted by:

Highest score (default)



Usually the process is as follows:

18



- the user starts the applicataion
- the application launches an "updater" (another program)
- the updater retrieves from the Internet if a newer version exists



Join Stack Overflow to find the best answer to your technical question, help others answer theirs.

Sign up



- the updater shuts the application down (or better, asks the user to do it) and launches the new installer.
- the installation package do the rest

Of course you can have many variation, but this is the basic way to do it.

On Windows, at least, some software install an updater deamon that is always on and checks for new updates of the software it takes care (GoogleUpdater, for example).

Share Improve this answer Follow

edited Nov 17, 2009 at 13:02

answered Nov 17, 2009 at 12:44



Remo.D

15.8k 6 43 73

How could be incremental? How to do that in Windows, with windows server and windows 7/10 ?

– Cirelli94 Oct 20, 2017 at 8:25

It may be "incremental" in the sense that the download package may contain just the updated files (for example a DLL) rather than the entire set of files (which may include images, sounds, ...). *How to do it depends on your application (language it is written in, ability to setup an "update" server, ...)*

– Remo.D Oct 29, 2017 at 11:31



A popular solution is to actually have a separate process replace the necessary files.

17



Have you ever noticed that Firefox has to restart whenever it updates? Well, that is because a separate process (updater.exe) is downloading the files, closing firefox, replacing the files, and then starting firefox again.



edit of course this assumes you're hosting a list of updates online that the updater program can check (an RSS feed, or even a simple text file).

Share Improve this answer Follow

answered Nov 17, 2009 at 12:38



Matt

42.8k 6 95 100

1 CCleaner [program to clean your system], even don't update itself. By clicking on **Update** it just downloads the new setup. now you have to manually install the software again, which ofcourse uninstall the previous version. I don't know why CCleaner programmers say it is an update ;)

– Rakesh Juyal Nov 17, 2009 at 12:53

Heh, yea, I would say CCleaner takes more of an 'overhaul' approach. – Matt Nov 17, 2009 at 12:59



You need to look at [Click Once Deployment](#).

15



ClickOnce is a deployment technology that enables you to create self-updating Windows-based applications that can be installed and run with minimal user

Join Stack Overflow to find the best answer to your technical question, help others answer theirs.

Sign up





applications deployed with ClickOnce technology if you have developed your projects with Visual Basic and Visual C#.

This creates a downloadable setup exe which you load up to the web. The user either runs this in place or downloads and runs it. This wraps your exe in the code that will check the web location (at a frequency of your choosing) for updates. If it finds one it will install it for the user.

You don't have to make any special changes to your code to enable this. beyond incrementing the version number. Just modify and extend your code as normal, compile, build, package and upload.

One thing to note is though that ClickOnce doesn't support installing applications that require administrative access:

ClickOnce deployment enables non-administrative users to install and grants only those Code Access Security permissions necessary for the application.

So if your application requires admin level access to run or even install then it's not the solution for you.

Share Improve this answer Follow

edited Nov 27, 2018 at 14:22

answered Nov 17, 2009 at 12:39



ChrisF ♦

132k

30

252

323

We've used Click-Once on a number of applications with 1000+ users and it works like a charm.

– Walter Nov 17, 2009 at 13:32

4 I've lost faith in Click-Once. If it works, sure it is nice, but when it fails, boy it is a huge mess. Note that none of the 'big boys' use ClickOnce (Microsoft doesn't even use it!) – Mark Lakata May 1, 2012 at 22:22

4 At this point, I'd recommended rolling your own. you just need a method to download the new files, then replace the current program with the new one. You will find fighting with the horrible bugs in ClickOnce not worth the effort. – Mark Lakata Mar 25, 2013 at 5:02

@MarkLakata: I was so excited about ClickOnce until read your comments. :(– Jack Jan 20, 2015 at 5:36

ClickOnce is not a solution when your app/exe requires Admin privilege. ClickOnce does not support the request execution level 'requireAdministrator' – Tito Nov 27, 2018 at 14:05



One idea - you could have a loader application that does the check for you, then if necessary updates the main app's exe before launching it?

6



Share Improve this answer Follow

answered Nov 17, 2009 at 12:38

Join Stack Overflow to find the best answer to your technical question, help others answer theirs.

Sign up





- 1 Then put some code in the next version of the main programme that does this silently as it runs if it needs to? – [David M](#) Jan 9, 2010 at 18:26
 - 1 I currently use this approach. However users are starting to pin the main application to the taskbar in windows7. Then they run that instead of the loader bypassing the update process. – [srayner](#) Jan 18, 2013 at 12:40
- Ah, good point. Obviously when I answered this originally Windows 7 had only just hit the retail market properly, but an interesting problem. Worthy of a question if someone hasn't already asked it? – [David M](#) Jan 18, 2013 at 14:24



For windows (generally due to windows locking the .exe file while it's being executed)

1



- Software determines it needs updating
- Software runs packaged update executable (at this point Software exits)
- Update downloads required files into some temp folder
- Update applies patches
- Update restarts Software

Share Improve this answer Follow

answered Nov 17, 2009 at 12:39



[Sekhat](#)

4,415

6

42

50



1



Well, the usual way is that your application checks somewhere (usually on the Internet) regularly for new versions. This may be on every startup or once a week or whatever.

There you can simply put for example a text file containing the current version number. If that one would be newer than the version of the installed application you can download an updater (don't forget to sign and check signatures) and install the new version.

Share Improve this answer Follow

answered Nov 17, 2009 at 12:39



[Joey](#)

336k

81

675

676



1



Make your app occasionally make a request over the web to your webserver. Have that request return a file that contains the version number and a link to the newest version of your installer. If that version number is greater than the version number the currently-running program thinks it is, have your program download the new setup file and launch it.

Share Improve this answer Follow

answered Nov 17, 2009 at 12:39



[Cory Petosky](#)

12.2k

2

37

43

Join Stack Overflow to find the best answer to your technical question, help others answer theirs.

[Sign up](#)



Not really what I'm looking for. – [Sergio Tapia](#) Nov 17, 2009 at 12:42

Not if you're using a sensible installer. – [Cory Petosky](#) Nov 17, 2009 at 12:45



You don't modify your current .exe.

1

You launch a separate process that downloads a new version of the file. The new version replaces the old version. No modification needed.



Share Improve this answer Follow

answered Nov 17, 2009 at 13:49



[roomaroo](#)

5,771 4 29 31



From a code point of view I like to have an approach that consists on a simple interface called IRun and another one called IUpdated:

0



```
public interface IRun()
{
    public void Run();
}

public interface IUpdated()
{
    public void Update();
}
```

Now my main application will just load and "run" a library using reflection which in turns launch the real application.

The main application executable will do this steps:

1. Read updater.cfg, connect to the provided location and check for **updates to the updater**.
2. If there are updates download the updated dlls and updater.cfg
3. Using reflection load updater.dll and launch Update() method.

The update method does the same steps for updates to the actual application.

Finally

4. Load "core.dll" and launch the Run() method.

That's of course provided you want to "do it yourself", ClickOnce is actually a very good method too, probably better but gives you less control. The main application executable is simple enough to avoid having to update it itself or should be anyway.



50.2k 11 80 127

I would really enjoy being able to study the C# code and techniques that implement this if they are publicly available anywhere. thanks, – [BillW](#) Nov 17, 2009 at 15:06