# Way1

# How to add a Uninstall option in Visual Studio Setup project without writing code

http://www.gogototo.com/Posts/how-to-add-a-uninstall-option-in-visual-studio-setup-project-without-writing-code.html

2008-09-05

Using Visual Studio 2005/2008, you don't need to write any code to add a uninstall option for a Setup project (Yes I know some people can write code to do it)

1. In the Setup Project --> File System windows --> Right Click "File System on Target machine" --> add a Special Folder, select System Folder;
2. Into this system folder Add a file. Browse for msiexec.exe from local System32 folder and add it. Override default properties of this file as follows:

Condition:=Not Installed (make sure you put 'Not Installed' exactly like that, same case and everything), Permanent:=True, System:=True, Transitive:=True, Vital:=False.

1. Create a new shortcut under the 'Users Program Menu', Set Target to the System Folder which you created in the step 1. and point it's at the msiexec.exe. Rename the shortcut to 'Uninstall Your Application'. Set the Arguments property to /x[ProductCode].
2. Build the project, ignore warning about the fact that msiexec should be excluded, DONT exclude it or the setup project wont build.

The 'Not Installed' condition and Permananet:=True ensure that the msiexec.exe is only placed into the system folder as part of the install IF it doesn't aready exist, and it is not removed on an uninstall - therefore it;s pretty safe to ignore that warning and just go for it.

(Based on the description from SlapHead)

# Way2

# [In a Visual Studio setup project, How do I generate an uninstall script?](https://stackoverflow.com/questions/1356160/in-a-visual-studio-setup-project-how-do-i-generate-an-uninstall-script)

<https://stackoverflow.com/questions/1356160/in-a-visual-studio-setup-project-how-do-i-generate-an-uninstall-script>

[Ask Question](https://stackoverflow.com/questions/ask)

Asked 13 years ago

Modified [4 years, 1 month ago](https://stackoverflow.com/questions/1356160/in-a-visual-studio-setup-project-how-do-i-generate-an-uninstall-script?lastactivity)

Viewed 47k times

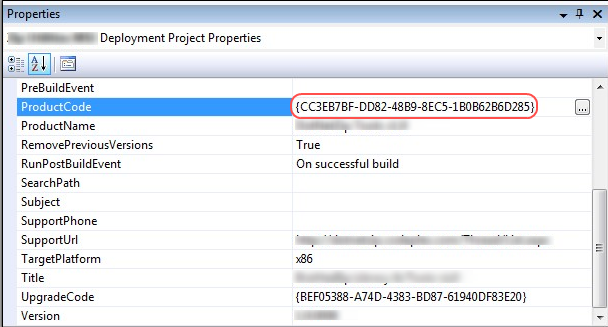
28

31

I have a Visual Studio setup project. Upon installation, it creates an uninstall batch file in the application folder. IF the user wants to uninstall the product, he can go to "Add/Remove Programs", or he can just double-click the uninstall.cmd. The contents are:

%windir%\system32\msiexec /x {CC3EB7BF-DD82-48B9-8EC5-1B0B62B6D285}

The GUID there is the ProductCode from the Setup Project in Visual Studio.



But, in order for upgrades to work properly, I have to increment the Version number, every time I produce a new MSI. And, if I increment the Version number, then I also have to generate a new Guid for the ProductCode. Which means the static uninstall.cmd file needs to change.

How can I dynamically generate a batch file that contains the ProductCode for the, at build time?

8 Answers

Sorted by:



67

I found this solution [here](http://www.gogototo.com/how-to-add-a-uninstall-option-in-visual-studio-setup-project-without-writing-code.html)

"Using Visual Studio 2005/2008, you don’t need to write any code to add a uninstall option for a Setup project (Yes I know some people can write code to do it)

1) In the Setup Project –> File System windows –> Right Click “File System on Target machine” –> add a Special Folder, select System Folder;

2) Into this system folder Add a file. Browse for msiexec.exe from local System32 folder and add it. Override default properties of this file as follows:

Condition:=Not Installed (make sure you put ‘Not Installed’ exactly like that, same case and everything), Permanent:=True, System:=True, Transitive:=True, Vital:=False.

3) Create a new shortcut under the ‘Users Program Menu’, Set Target to the System Folder which you created in the step 1. and point it’s at the msiexec.exe. Rename the shortcut to ‘Uninstall Your Application’. Set the Arguments property to /x{space}[ProductCode].

4) Build the project, ignore warning about the fact that msiexec should be excluded, DONT exclude it or the setup project wont build.

The ‘Not Installed’ condition and Permananet:=True ensure that the msiexec.exe is only placed into the system folder as part of the install IF it doesn’t aready exist, and it is not removed on an uninstall - therefore it;s pretty safe to ignore that warning and just go for it.

(Based on the description from SlapHead)"

[Share](https://stackoverflow.com/a/1367269)

Follow

[edited Mar 22, 2011 at 17:00](https://stackoverflow.com/posts/1367269/revisions)

[[](https://stackoverflow.com/users/210916/mpelletier)](https://stackoverflow.com/users/210916/mpelletier)

[MPelletier](https://stackoverflow.com/users/210916/mpelletier)

**15.8k**1515 gold badges8484 silver badges132132 bronze badges

answered Sep 2, 2009 at 11:55

[[](https://stackoverflow.com/users/131095/ben)](https://stackoverflow.com/users/131095/ben)

[Ben](https://stackoverflow.com/users/131095/ben)

**835**11 gold badge88 silver badges99 bronze badges

* 1

Now that's very handy and much more direct than the approach I took.

– [Cheeso](https://stackoverflow.com/users/48082/cheeso)

[Sep 2, 2009 at 19:59](https://stackoverflow.com/questions/1356160/in-a-visual-studio-setup-project-how-do-i-generate-an-uninstall-script#comment1208123_1367269)

* 8

Awesome, works very well for me on WinXP/Vista & Win7. Worth noting that according to the article linked to, if you use a version of msiexec.exe HIGHER than the target platform there are issues. This is not an issue for me as I'm still intentionally using XP for building - but obviously if you want an installer built on Vista to run on XP that will be an issue.

– [Iain Collins](https://stackoverflow.com/users/28884/iain-collins)

[Nov 24, 2009 at 12:15](https://stackoverflow.com/questions/1356160/in-a-visual-studio-setup-project-how-do-i-generate-an-uninstall-script#comment1676794_1367269)

* Great Post! Exactly what i was looking for.

– [Martin](https://stackoverflow.com/users/559085/martin)

[Dec 8, 2011 at 5:34](https://stackoverflow.com/questions/1356160/in-a-visual-studio-setup-project-how-do-i-generate-an-uninstall-script#comment10411564_1367269)

* 3

Regarding the problem that arises when building the Installer under Vista/7, and the installer tries to overwrite msiexec on older, XP machines; this can be corrected by creating an empty .txt file, and renaming it to "msiexec.exe." Include that file in your installer, rather than the *actual* System32 version. Since the empty, fake file has no version information, it will not try to overwrite the XP msiexec.

– [BTownTKD](https://stackoverflow.com/users/741988/btowntkd)

[May 17, 2012 at 18:39](https://stackoverflow.com/questions/1356160/in-a-visual-studio-setup-project-how-do-i-generate-an-uninstall-script#comment13796288_1367269)

* 2

@NickName I got the same error too. Set the Arguments property to /x{space}[ProductCode] isn't work for me ,but I change the Arguments property to /x [ProductCode],it works.I guess visual studio just can't recognize what's {space} means.by the way ,I tried in VS2005.

– [Rhain](https://stackoverflow.com/users/1118924/rhain)

[May 29, 2015 at 2:56](https://stackoverflow.com/questions/1356160/in-a-visual-studio-setup-project-how-do-i-generate-an-uninstall-script#comment49115251_1367269)

[Show **1** more comment](https://stackoverflow.com/questions/1356160/in-a-visual-studio-setup-project-how-do-i-generate-an-uninstall-script)

10

I just made [this](https://stackoverflow.com/questions/5399881/variables-in-shortcut-arguments-ignored-for-visual-studio-setup-project) work:

Add an uninstall.bat file to your project. The file contents is:

msiexec.exe /x %1

Make a shortcut to that batch file (say in User's Program Menu), specify in the shortcut's properties, next to Arguments: [ProductCode].

[Share](https://stackoverflow.com/a/5764736)

Follow

[edited May 23, 2017 at 12:00](https://stackoverflow.com/posts/5764736/revisions)

[[](https://stackoverflow.com/users/-1/community)](https://stackoverflow.com/users/-1/community)

[Community](https://stackoverflow.com/users/-1/community)Bot

**1**11 silver badge

answered Apr 23, 2011 at 14:26

[[](https://stackoverflow.com/users/210916/mpelletier)](https://stackoverflow.com/users/210916/mpelletier)

[MPelletier](https://stackoverflow.com/users/210916/mpelletier)

**15.8k**1515 gold badges8484 silver badges132132 bronze badges

* 1

You can get rid of the black window by using start msiexec.exe /x %1; but the black window still flashes.

– [moala](https://stackoverflow.com/users/54956/moala)

[Jul 13, 2012 at 15:40](https://stackoverflow.com/questions/1356160/in-a-visual-studio-setup-project-how-do-i-generate-an-uninstall-script#comment15149479_5764736)

* Probably start /B msiexec.exe /x %1.

– [MPelletier](https://stackoverflow.com/users/210916/mpelletier)

[Jul 13, 2012 at 15:58](https://stackoverflow.com/questions/1356160/in-a-visual-studio-setup-project-how-do-i-generate-an-uninstall-script#comment15149955_5764736)

* Try /min, it should at least minimize it.

– [MPelletier](https://stackoverflow.com/users/210916/mpelletier)

[Jul 13, 2012 at 17:48](https://stackoverflow.com/questions/1356160/in-a-visual-studio-setup-project-how-do-i-generate-an-uninstall-script#comment15152671_5764736)

* 4

no, the windows which flashes is not the one of the start command, it is the one of the .bat, so an additional switch is not the solution; the solution is at the vdproj installation project: the ShowCmd property of the shortcut calling from the value vsdscNormal value to the vsdscMinimized value. Solved.

– [moala](https://stackoverflow.com/users/54956/moala)

[Jul 15, 2012 at 15:12](https://stackoverflow.com/questions/1356160/in-a-visual-studio-setup-project-how-do-i-generate-an-uninstall-script#comment15180996_5764736)

[Add a comment](https://stackoverflow.com/questions/1356160/in-a-visual-studio-setup-project-how-do-i-generate-an-uninstall-script)

4

This is the script I wrote that creates an uninstall.cmd. It runs as a custom action during installation.

var fso, ts;

var ForWriting= 2;

fso = new ActiveXObject("Scripting.FileSystemObject");

var parameters = Session.Property("CustomActionData").split("|");

var targetDir = parameters[0];

var productCode = parameters[1];

ts = fso.OpenTextFile(targetDir + "uninstall.cmd", ForWriting, true);

ts.WriteLine("@echo off");

ts.WriteLine("goto START");

ts.WriteLine("=======================================================");

ts.WriteBlankLines(1);

ts.WriteLine(" Uninstall.cmd");

ts.WriteBlankLines(1);

ts.WriteLine("=======================================================");

ts.WriteBlankLines(1);

ts.WriteLine(":START");

ts.WriteLine("@REM The uuid is the 'ProductCode' in the Visual Studio setup project");

ts.WriteLine("%windir%\\system32\\msiexec /x " + productCode);

ts.WriteBlankLines(1);

ts.Close();

The result is a cmd file that always has the current ProductCode in it.

The downside of this is that ?*the script that creates the uninstall.cmd* remains in the installation directory. Not a huge problem but I don't like the rubbish in the install directory. I haven't yet tried to make the "createInstaller.js" self-deleting. That might work.

**EDIT**: yes, making the createInstaller.js self-deleting works fine.

I'm going to accept my own answer!

[Share](https://stackoverflow.com/a/1356485)

Follow

[edited Oct 10, 2009 at 17:50](https://stackoverflow.com/posts/1356485/revisions)

answered Aug 31, 2009 at 8:42

[[](https://stackoverflow.com/users/48082/cheeso)](https://stackoverflow.com/users/48082/cheeso)

[Cheeso](https://stackoverflow.com/users/48082/cheeso)

**186k**9898 gold badges464464 silver badges708708 bronze badges

* 1

Fine solution but damn hacky, I'm wondering about Microsoft, Install-makers I tried before (such inno) was making unistall file automatically.

– [cnd](https://stackoverflow.com/users/238232/cnd)

[May 26, 2011 at 4:58](https://stackoverflow.com/questions/1356160/in-a-visual-studio-setup-project-how-do-i-generate-an-uninstall-script#comment7119325_1356485)

* 1

I know it's been a few years, but I for one would like to know how exactly you went about using this, Cheeso. Are you overriding the OnAfterInstall? If so, how are you inserting Script there?

– [jp2code](https://stackoverflow.com/users/153923/jp2code)

[Jul 27, 2011 at 15:32](https://stackoverflow.com/questions/1356160/in-a-visual-studio-setup-project-how-do-i-generate-an-uninstall-script#comment8140179_1356485)

[Add a comment](https://stackoverflow.com/questions/1356160/in-a-visual-studio-setup-project-how-do-i-generate-an-uninstall-script)

2

a slight variation on the batch file approach. if you don't want to show a command window, use a wscript file. supply [ProductCode] as before in Arguments

<job>

<?job debug="true" error="true" ?>

<script language="JScript">

var arg = [];

for (var i=0; i<WSH.Arguments.length; i++) {

arg.push( WSH.Arguments.Item(i) );

}

if (arg.length>0) {

var productcode = arg[0];

var v = new ActiveXObject("Shell.Application");

v.ShellExecute("msiexec.exe", "/x "+productcode, "", "open", 10);

}

WSH.Quit(0);

</script>

</job>

[Share](https://stackoverflow.com/a/9028895)

Follow

answered Jan 27, 2012 at 4:12

[[](https://stackoverflow.com/users/1153573/locksmith)](https://stackoverflow.com/users/1153573/locksmith)

[locksmith](https://stackoverflow.com/users/1153573/locksmith)

**2,291**11 gold badge1212 silver badges44 bronze badges

[Add a comment](https://stackoverflow.com/questions/1356160/in-a-visual-studio-setup-project-how-do-i-generate-an-uninstall-script)

1

For removing the application I would use the [ProductCode] as a parameter, calling the msiexec from within the application itself - for a detailed guide as to creating the uninstaller please check this blog: <http://endofstream.com/creating-uninstaller-in-a-visual-studio-project/>

[Share](https://stackoverflow.com/a/3603381)

Follow

answered Aug 30, 2010 at 19:04

[[](https://stackoverflow.com/users/435266/peter)](https://stackoverflow.com/users/435266/peter)

[Peter](https://stackoverflow.com/users/435266/peter)

**11**11 bronze badge

[Add a comment](https://stackoverflow.com/questions/1356160/in-a-visual-studio-setup-project-how-do-i-generate-an-uninstall-script)

1

I realize this question has already been answered, but what I did was create a batch file with the following content:

start msiexec.exe /x %1

Using "start" prevents the command prompt from staying open during the uninstallation process and the %1 is replaced by the first argument you pass to the batch file when executing it.

I added the batch file to the application folder and created a menu shortcut that points to the batch file. If you right click the shortcut and select Properties Window. In there, add the following text to the Arguments property:

[ProductCode]

After building and running the installer, you'll have a start menu shortcut that calls the batch file and passes the product code as it was when that version of the application was built. The user will then see a prompt asking if the product should be uninstalled.

The only downside of this solution is that if the user browses to the application folder and double clicks the batch file, the uninstaller will give an error that the install package does not exist.

[Share](https://stackoverflow.com/a/15272583)

Follow

answered Mar 7, 2013 at 13:34

[[](https://stackoverflow.com/users/125001/jela)](https://stackoverflow.com/users/125001/jela)

[Jela](https://stackoverflow.com/users/125001/jela)

**542**11 gold badge66 silver badges2020 bronze badges

[Add a comment](https://stackoverflow.com/questions/1356160/in-a-visual-studio-setup-project-how-do-i-generate-an-uninstall-script)

1

I did a combination of the accepted answer and Locksmith's answer to not see any flashes of command prompt or the command prompt itself. One pro of doing it like this is that you DONT HAVE to create a shortcut and set the arguments for it in order for it to work.

My createUninstaller.js file:

var fso, ts;

var ForWriting= 2;

fso = new ActiveXObject("Scripting.FileSystemObject");

var parameters = Session.Property("CustomActionData").split("|");

var targetDir = parameters[0];

var productCode = parameters[1];

ts = fso.OpenTextFile(targetDir + "Uninstall.js", ForWriting, true);

ts.WriteLine("var v = new ActiveXObject(\"Shell.Application\");");

ts.WriteLine("v.ShellExecute(\"msiexec.exe\", \"/x "+productCode+"\", \"\", \"open\",10);");

ts.Close();

This file is added as a custom action at the commit action 'directory'. In order to actually get to this custom actions: right click your setup project>view>custom actions>right click commit 'directory'>add custom action. After you have to search for the createUninstaller.js file you created and add it.

Now to make the createUninstaller.js read the variables targetDir and productCode you have to  
Right click the createUninstaller.js file in the setup project custom action 'commit' directory and go to properties window. In properties you will see the 'CustomActionData' property. In there you just copy paste [TARGETDIR]|[ProductCode]  
And VOILA! It should now add the Uninstall.js file which will work as soon as you double click it.

[Share](https://stackoverflow.com/a/51509844)

Follow

[edited Jul 25, 2018 at 13:01](https://stackoverflow.com/posts/51509844/revisions)

answered Jul 25, 2018 at 2:05

[[](https://stackoverflow.com/users/6008458/merv)](https://stackoverflow.com/users/6008458/merv)

[Merv](https://stackoverflow.com/users/6008458/merv)

**929**11 gold badge99 silver badges2121 bronze badges

[Add a comment](https://stackoverflow.com/questions/1356160/in-a-visual-studio-setup-project-how-do-i-generate-an-uninstall-script)

0

every time when u generate setup then always change product code. create a uninstaller shortcut and there u will find command line argument passing technique to sort cut. there u will write always "/u product code" product code u need to write here always.

put this code in main method of program.cs file

[STAThread]

static void Main()

{

bool flag = true;

string[] arguements = Environment.GetCommandLineArgs();

foreach (string str in arguements)

{

if ((str.Split('='))[0].ToLower() == "/u")

{

if (MessageBox.Show("Do you want to uninstall job board", "Are you Sure?", MessageBoxButtons.YesNo) == DialogResult.Yes)

{

flag = false;

RemoveRegSettings();

RemoveIniFile();

string guid = str.Split('=')[1];

string path = Environment.GetFolderPath(Environment.SpecialFolder.System);

ProcessStartInfo si = new ProcessStartInfo(path + @"\msiexec.exe", "/i" + guid);

Process.Start(si);

Application.Exit();

return;

}

}

}

//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

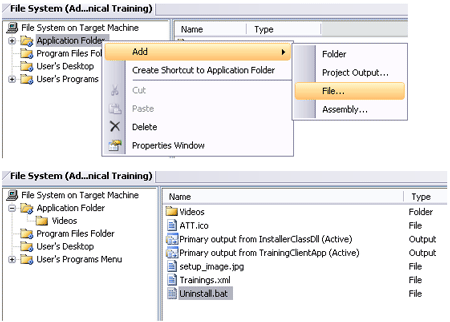
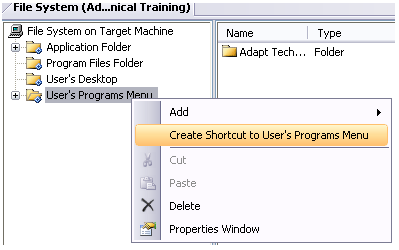
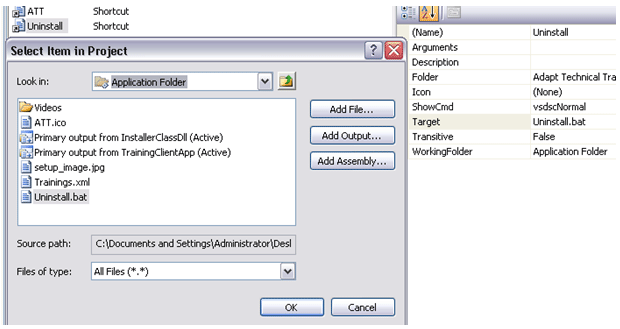
Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

# Way3

<https://www.c-sharpcorner.com/uploadfile/nipuntomar/making-a-uninstall-shortcut-with-visual-studio-setup-project/>

Make a batch file (Uninstall.bat) with the following call  
@echo off  
msiexec /x {xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx}  
(Path depends of your Windows version, check where your system32 folder is located)  
That last part is the product code which you can find in the properties of your setup project.

1. Add the Uninstall.bat in application folder  
     
   right click on Application Folder > Add > File > Uninstall.bat  
     
   
2. Right click on "User's Programs Menu" and select "Create Shortcut to User's Programs Menu".  
     
   
3. Rename it to Uninstall
4. Change the target of Uninstall shortcut to the Uninstall.bat.  
     
   Uninstall shortcut > Properties Window > Target > Application Folder > Uninstall.bat  
     
   
5. Rebuild the project and you are done.