

Make your voice heard. [Take the 2019 Developer Survey now](#)

How add context menu item to Windows Explorer for folders

[Ask Question](#)

71



63

I have found out how to add right-click context menu items to files on Windows Explorer, by adding keys to the registry. I.e. I can right-click on a file in Explorer and run a custom app against that file.

I would like to do the same for a folder and have not found a way to do that (yet). I see articles on creating/writing custom context menu handlers, but I would rather not go there.

I have found an article [here](#) on how to add cascading context menu items to the Desktop and to the "Computer" in Explorer, but this does not work for any folder.

I would like to be able to add my custom app to the context menu and have it work on both files and folders. Is there a way to do

this without writing a
context menu
handler?

[contextmenu](#)[explorer](#)

asked Dec 8 '13 at 2:52



[Elan](#)

2,771

8

54

82

3 Answers



149



**Context menu
for right click on
folders in left
panel of
Windows
Explorer or on
background of a
directory in right
panel:**

1. **HKEY_CLASSES
_ROOT\Directory
\Background\shel**
If you are
administrator
2. **HKEY_CURREN
T_USER\Softwar
e\Classes\direct
ory\Background\
shell** if you are a
normal user

**Context menu
for right click on
folders in right
panel of
Windows
Explorer:**

1. **HKEY_CLASSES**
_ROOT\Directory
\shell if you are
administrator
2. **HKEY_CURREN**
T_USER\Softwar
e\Classes\direct
ory\shell if you
are a normal user

Context menu for any file:

1. **HKEY_CLASSES**
_ROOT*\shell if
you are
administrator
2. **HKEY_CURREN**
T_USER\Softwar
e\Classes*\shell
if you are a
normal user

In all cases:

1. add a new key
under "shell",
naming it as you
want to name the
context menu
item
2. add a new key
inside this key,
named **command**
(mandatory
name)
3. edit the "default"
property in
"command" to
myprogrampath
path\path\execut
able.exe %1 to
pass the file path
and name of the
selected file to
your custom
program

More customization:

- **Add icon:** adds a string value named `icon` for key created at step 1 with value matching an icon resource path. You can also provide an integer arguments to specify which icon to use. Example:

```
%SystemRoot%\System32\shell32.dll,  
3
```

- **Display only on shift-click:** adds an empty string value named `Extended` for key created at step 1
- **Customize menu entry label:** change the value of default value for key created at step 1
- **Change menu entry location:** adds a string value named `Position` with one of: `Top` , `Bottom`

edited May 11 '17 at 12:29

community wiki

3 revs, 2 users 86%

jumpjack

2 HKEY_CLASSES_ROOT
 \Directory\Background\shell can only
 affect right click on

background of a
directory – [Amos](#)
Sep 23 '15 at 12:59

12 Should be "%1" in
case path has
spaces. – [Blrp](#) Jan
13 '16 at 19:18

1 immediately. There
isn't even need to
open a new explorer
window: new
settings are loaded
as soon as you
right-click. –
[jumpjack](#) Mar 1 '16
at 18:43

1 seems environment
variables do not
work in Command ,
do they? – [Jack Lu](#)
Mar 25 '16 at 14:47

4 Regarding my
previous comment:
The problem can be
fixed by replacing
%1 by %V . –
[Michael Herrmann](#)
Aug 23 '17 at 10:48



I found the solution in
the below article,
which describes how
to do this via the
registry for files, as
well as for folders:

- [How to Add Any
Application
Shortcut to
Windows
Explorer's
Context Menu](#)

The following two
articles provided

additional info and options:

- [Ultimate Tutorial to Customize Desktop Context Menu in Windows Vista, 7 and 8](#)
- [Add Cascading Menus for Your Favorite Programs in Windows 7 Desktop and My Computer Context Menus](#)

answered Dec 8 '13 at 19:39



Elan

2,771 8 54 82



I went back and also answered this in another [topic](#) since there doesn't appear to be much on this question specifically.

I found the simplest way was to add a String Value to the key called "AppliesTo" and set its value to "under: {path}"

In my example, I want it to only look in the T Drive, so my String value is "AppliesTo":"under:T".

In C#, this is easily accomplished with the following:

```
RegistryKey _key = Reg:
RegistryKey newkey = _l
newkey.SetValue("Appli
```

```
RegistryKey subNewkey :
subNewkey.SetValue("",
```

```
subNewkey.Close();
```

```
newkey.Close();  
_key.Close();
```

edited May 23 '17 at 12:26



Community ♦

1 1

answered Jul 6 '16 at 15:41



Bobby Byrnes

117 1 12

2 Thanks for providing sample code for doing this via C#. It's a small help for anyone who wants to do this quickly from an app. – [Ayo I](#) Apr 12 '17 at 19:09

2 Unfortunately, the "AppliesTo" value doesn't seem to have any effect on a "Directory\Background" entry. Does anyone know how similar limiting can be accomplished for right clicking on a folder's background? – [cl0rkster](#) Feb 22 '18 at 20:26
