

Change Ownership of a File or Folder Using Takeown Command-Line Tool

<https://www.winhelponline.com/blog/take-ownership-of-file-or-folder/>

Every file or folder in an NTFS volume has an owner. Certain system files are owned by TrustedInstaller, some by SYSTEM account, and others by the “Administrators” group. If a user creates a file or folder, that user is usually the owner of the file or folder. The owner is the one who can assign permissions (Allow or Deny) to other users for that object.

If a user is not the owner of a file or folder or has no permissions to access the file, he gets the “access is denied” error when accessing the object. If that user is an administrator, he can take ownership of the object using the file or folder’s Properties – Security tab. Then he can assign himself the required permissions.

This post tells you how to **take ownership of a file or folder**, and assign required permissions for it using the command-line instead of GUI.

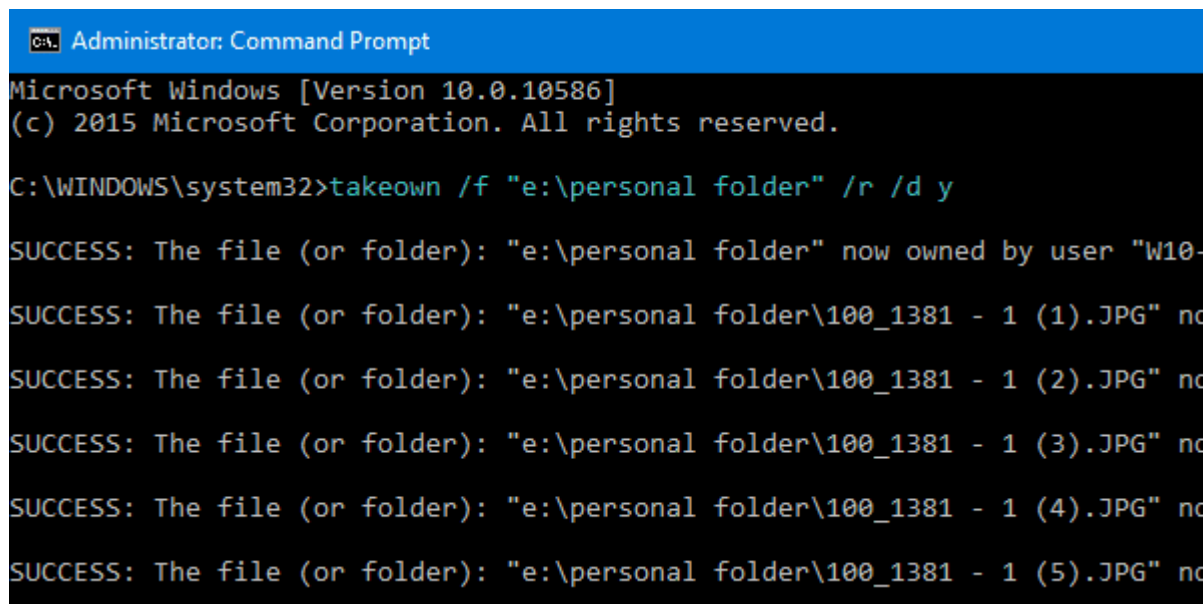
Contents

- Taking ownership of a file using takeown.exe
- Assign file permissions using icacls.exe
- Taking ownership of a folder using takeown.exe
- Assign folder permissions using icacls.exe
- Take Ownership & Assign Permissions via Right-click Menu [Script/Tweak]

Additional Info

- INFO: icacls.exe can also change ownership
- Revert the Ownership to TrustedInstaller
- ERROR: icacls.exe /setowner “Access is Denied”
- Using SetACL.exe to take ownership and assign permissions [3rd Party freeware]

Windows includes a command-line tool named **Takeown.exe**, which can be used from an [admin Command Prompt](#) to quickly change the ownership of a file or folder. Here is how to **take ownership of a file or folder** and then assign permissions for an account using the command-line.



```
C:\> Administrator: Command Prompt
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>takeown /f "e:\personal folder" /r /d y

SUCCESS: The file (or folder): "e:\personal folder" now owned by user "W10-
SUCCESS: The file (or folder): "e:\personal folder\100_1381 - 1 (1).JPG" no
SUCCESS: The file (or folder): "e:\personal folder\100_1381 - 1 (2).JPG" no
SUCCESS: The file (or folder): "e:\personal folder\100_1381 - 1 (3).JPG" no
SUCCESS: The file (or folder): "e:\personal folder\100_1381 - 1 (4).JPG" no
SUCCESS: The file (or folder): "e:\personal folder\100_1381 - 1 (5).JPG" no
```

Taking ownership of a file

Open an [elevated Command Prompt](#) window. Use the following syntax to take ownership of a file:

```
TAKEOWN /F <filename>
```

Replace **<filename>** with the actual file name with the full path.

The currently logged on user is now the owner of the file.

To set **Administrators** group the owner of the file, use the **/A** switch in addition:

```
TAKEOWN /F <filename> /A
```

- **/A** Gives ownership to the Administrators group instead of the current user. If **/A** is not specified, file ownership will be given to the currently logged-on user. This parameter is **not** case-sensitive.

If the operation was successful, you should see the following message:

```
"SUCCESS: The file (or folder): "filename" now owned by user "Computer Name\User name"."
```

or

```
SUCCESS: The file (or folder): "filename" now owned by the administrators group.
```

Assign File Permissions

Then to grant **Administrators** Full Control permissions for the file, use **ICACLS**. Here is the syntax:

```
ICACLS <filename> /grant administrators:F
```

Example 2: To assign Full Control permissions for the currently logged on user, use this command:

```
ICACLS <filename> /grant %username%:F
```

%username% represents the account name of the currently logged-on user. **ICacls** accepts this variable directly.

Example 3: To assign Full Control permissions for the user named **John**, use this command:

```
ICACLS <filename> /grant John:F
```

Taking ownership of a folder

Use the following syntax:

```
takeown /f <foldername>
```

(or)

```
takeown /f <foldername> /a
```

- **/a** Gives ownership to the Administrators group instead of the current user. If **/a** is not specified, ownership will be given to the currently logged-on user. This parameter is **not** case-sensitive.

You'll see the following output:

```
SUCCESS: The file (or folder): "folder_name" now owned by by user "computername\username"
```

(or)

```
SUCCESS: The file (or folder): "folder_name" now owned by the administrators group.
```

Change ownership recursively:

To change the ownership of a folder, its subfolders, and files in each subfolder, use this syntax:

```
takeown /f <foldername> /r /d y
```

The currently logged on user is now the owner of the folder.

To set **Administrators** group the owner of the folder, its subfolders, and files recursively, use the **/A** switch in addition:

```
TAKEOWN /F <foldername> /a /r /d y
```

- **/R** Recurse: instructs the tool to operate on files in the specified directory and all subdirectories.
- **/D** prompt Default answer used when the current user does not have the "list folder" permission on a directory. This occurs while operating recursively (**/R**) on sub-directories. Valid values "Y" to take ownership or "N" to skip.

Assign Folder Permissions

Then to assign the **Administrators** group Full Control Permissions for the folder, use this syntax:

```
icacls <foldername> /grant administrators:F /T
```

The **/T** parameter is added so that the operation is carried out through all the sub-directories and files within that folder.

Command-line help:

To know the complete usage information for Takeown.exe and ICacls.exe, run these commands from a Command Prompt window.

```
takeown /?
```

```
icacls /?
```

Easier Methods for Taking Ownership

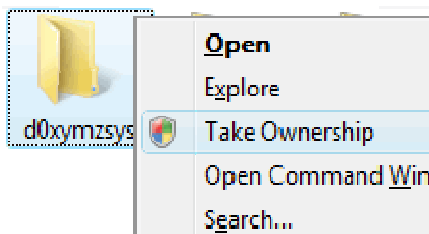
Command Script

To further simplify the process of taking ownership, [Tim Sneath](#) of Microsoft provides a .CMD file (Windows Command Script) which takes ownership and assigns Full Control Permissions to Administrators for the directory. For more information, read Tim's post, titled [Secret #11: Deleting the Undeletable](#).

Add the "Take Ownership" command to the right-click menu

This again uses the special *runas* verb in Windows Vista and higher, which I've covered earlier ([REF RunAs](#)).

via [WinMatrix.com](#)



Download takeown_context.reg and save to Desktop. Right-click on the file and choose **Merge**. Click **Yes** when asked for confirmation. This adds an extended command named **Take Ownership** in the context menu for files and directories. To access the command, you need to press and hold the **SHIFT** key and then right-click on a file or folder.

Windows Registry Editor Version 5.00

```
;Adds Take Ownership command to the context menu in Vista  
;Also provides "Administrators" group with Full Control permissions.
```

```
[-HKEY_CLASSES_ROOT*\shell\runas]
```

```
[HKEY_CLASSES_ROOT*\shell\runas]
```

```
@="Take Ownership"
```

```
"Extended"=""
```

```
"NoWorkingDirectory"=""
```

```
[HKEY_CLASSES_ROOT*\shell\runas\command]
```

```
@="cmd.exe /c takeown /f \"%1\" && icacls \"%1\" /grant administrators:F"
```

```
[-HKEY_CLASSES_ROOT\Directory\shell\runas]
```

```
[HKEY_CLASSES_ROOT\Directory\shell\runas]
```

```
@="Take Ownership"
```

```
"Extended"=""
```

```
"NoWorkingDirectory"=""
```

```
[HKEY_CLASSES_ROOT\Directory\shell\runas\command]
```

```
@="cmd.exe /c takeown /f \"%1\" /r /d y && icacls \"%1\" /grant administrators:F /t"
```