

Forum:[Hack Forums](#)

Thread: [Fill Up Disk Space Automatically \(FREE Python Script / Source Code\)](#)

Hackers - January 7, 2024, 9:00 am

Automatically Fill Your Disk Space

Using this Python script you can create files that take up any amount of disk space. You can simply set the variable for each bytes_amount, which is the filesize, and the number_files, for how many. There is also an automatic mode that will find the available disk space & then create files that fill the entire space automatically. See notes below!

In my example, I create a 7 files that are just under 1GB each. This creates each file in the same directory that I run my script from using my command:

Code python run.py

Working as of 1/7/2024 with Python 3.12

Source Code:

```
Code import os
import platform
import ctypes
import shutil
import math
```

```
def set_hidden_attribute(file_name):
    if platform.system() == "Windows":
        # Using ctypes to set file as hidden in Windows
        FILE_ATTRIBUTE_HIDDEN = 0x02
        ctypes.windll.kernel32.SetFileAttributesW(file_name, FILE_ATTRIBUTE_HIDDEN)
```

```
def create_files(byte_size, number_of_files, use_hidden_files):
    for i in range(number_of_files):
        file_name = f"file_{i+1}.txt"
        if use_hidden_files:
            if platform.system() != "Windows":
                file_name = "." + file_name # for Unix-like systems
            with open(file_name, "wb") as file:
                file.write(b'0' * byte_size)
        if use_hidden_files and platform.system() == "Windows":
            set_hidden_attribute(file_name)
```

```
def get_free_disk_space(path="/"):
    total, used, free = shutil.disk_usage(path)
    return free
```

```
bytes_amount = 1000000 * 1000 # 1MB * amount of megabytes, for readability
number_files = 7
use_hidden_files = True
automatic_file_size = False # if True, bytes_amount above is overridden and auto calculated to fill
```

entire disk

```
if automatic_file_size:
    print("Finding Available Disk Space...")
    free_space = get_free_disk_space("/")
    print("Free Disk Space: " + str(free_space) + " bytes " + f"({free_space / (1024**3):.2f} GB)")
    bytes_amount = math.floor(free_space / number_files)
    print("Bytes Per New File: " + str(bytes_amount))
else:
    print("Using Provided Disk Space Settings")
    print("Bytes Per File: " + str(bytes_amount) + ", # of Files: " + str(number_files))
```

create_files(bytes_amount, number_files, use_hidden_files)

If you are on Linux/Unix, you create hidden files simply by adding the . in front of it. On Windows, we need to use the Platform package (or win32 packages, or similar) to make the file hidden. This is because a file with a . period in front of it is not hidden automatically on Windows. For that reason, we use the platform.system() function to check what OS we're running Python on.

You can see the difference here, where I have 3 files that are not hidden, and 7 that are. Each one takes up a little less than 1GB:

These are real files, with real bytes inside - not just a fake file that pretends to take up space, which other methods may use. This really does eat up disk space!

With the automatic bytes size enabled, we are finding how much space is left, then figuring out how many bytes per file should be used. Here is the script output confirming these settings (which takes my available space and divides by my number of files (7)) :

If you set automatic_file_size to False, it will use the provided bytes_amount and file size. I have kept the default as False so you don't accidentally fill your entire disk space while testing.

Good luck, I hope this helps you! Please be safe out there - keep in mind that a computer/server with a full disk space may not have enough space to do normal file operations such as saving a session file to let you log in, etc.

Binance - January 7, 2024, 7:51 pm

This is very useful for doing wipe passes... thanks.

Hackers - January 8, 2024, 6:52 am

(Jan 7, 2024 08:51 PM)signed0 Wrote: [\[CITE\]](#) This is very useful for doing wipe passes... thanks.
No problem, glad you have a use case for it!

1000EyesUponMe - January 10, 2024, 12:59 am

What are other use cases for this type of thing?

Hackers - January 10, 2024, 6:11 am

(Jan 10, 2024 01:59 AM)1000EyesUponMe Wrote: [\[CITE\]](#) What are other use cases for this type of thing?

Filling up disk space on your computer or server can cause problems. For example, if disk space is full then login systems can't create necessary session, log, or cache files for normal use. This can

cause default or third party software on the computer / server to completely fail, shut down, freeze, etc.

It's useful for repurposing disk space for example if you've deleted secure files and want to write over the disk with new data. I am not sure how useful this is, but the concept of writing over disk multiple times to obscure activity or make it near impossible to retrieve deleted or corrupted data has come up a few times throughout the years (in discussion boards at least).

It can be used for simulating large file transfers, timeout issues, etc on code. You can use it for sending large files to your/a server for testing server processing times & max file size rules, etc.

There are other use cases but these are some things that come to mind - granted its just past 9am and I'm half a coffee in only.

Stanley - *October 28, 2025, 6:49 pm*

I'm Stanley HF's helper bot. The OP (original poster) of this thread wants everyone to see this page and has elected to bump it to the top of the showthread listing. Since HF doesn't want a ton of LQP (Low Quality Posts) this is the best method of bumping your content. There is a small ?yte fee for each bump. Any OP can bump any of their threads on HF from the linked icon on the top of their first post.

If you haven't already read the first post of this thread we encourage you to do so.

Thanks,
Stanley

Hackers - *March 16, 2026, 8:19 pm*

Seemed like this was useful to a couple people in the past, for niche use cases.