



Long File Path Shorten Script Documentation

2024-02-07

Internal © 2024 CGI Inc.

Table of contents

[1 Overview and Purpose 3](#_Toc158285766)

[1.1 Purpose 3](#_Toc158285767)

[1.2 Main Features 3](#_Toc158285768)

[1.2.1 Main Feature – Scanning 3](#_Toc158285769)

[1.2.2 Main feature – Shorten Long Filenames 4](#_Toc158285770)

[1.2.3 Main features – Shorten Long Nested Folders 5](#_Toc158285771)

[2 Getting Started 6](#_Toc158285772)

[2.1 Installation 6](#_Toc158285773)

[2.2 Configuration 6](#_Toc158285774)

[2.2.1 config.ini 6](#_Toc158285775)

[2.2.2 abbreviation\_dictionary.csv 7](#_Toc158285776)

[3 Usage Instructions 8](#_Toc158285777)

[3.1 Running the Script 8](#_Toc158285778)

[3.1.1 Scanning for Long Paths and Filenames: 8](#_Toc158285779)

[3.1.2 Shortening Folder Names: 8](#_Toc158285780)

[3.1.3 Shortening Filenames: 8](#_Toc158285781)

[3.2 Option Flags 8](#_Toc158285782)

[3.3 Dry Run Mode 8](#_Toc158285783)

[3.3.1 Enabling Dry Run: 8](#_Toc158285784)

[4 Understanding Output 9](#_Toc158285785)

[4.1 Scan Results - How to Interpret? 9](#_Toc158285786)

[4.2 Shortening Process – What to Expect? 9](#_Toc158285787)

[5 Troubleshooting and FAQs 10](#_Toc158285788)

[5.1 Basic Troubleshooting 10](#_Toc158285789)

[5.2 FAQs 10](#_Toc158285790)

[5.2.1 Question 1 10](#_Toc158285791)

[5.2.2 Question 2 10](#_Toc158285792)

[5.2.3 Question 3 10](#_Toc158285793)

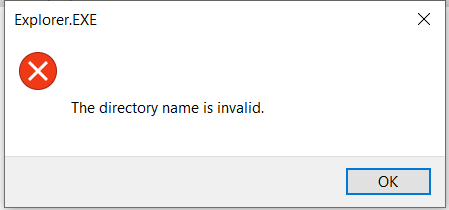
[5.2.4 Question 4 10](#_Toc158285794)

[Appendix A Appendix Heading 1 11](#_Toc158285795)

# Overview and Purpose

## Purpose

This document serves as the official guide for using the **FilePathShortener** script, designed to identify and shorten long file paths and filenames within a Windows environment. Its primary goal is to mitigate issues related to path length restrictions.


## Main Features

FilePathShortener provides three primary features: **Scanning**, **Shortening Long Filenames**, and **Shortening Long Nested Folders**.

### Main Feature – Scanning

The scanning feature methodically examines directories and their subdirectories for files and folders with paths exceeding predefined length thresholds.

**Output:** The scan results are outputted into two distinct files, categorizing the findings into long filenames and long directory paths, respectively.

A close-up of a document

Description automatically generated

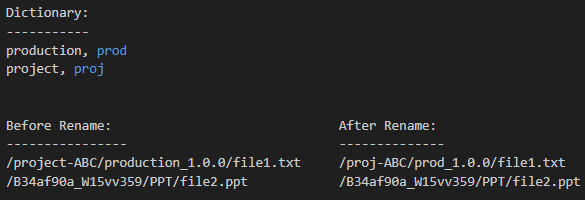
### Main feature – Shorten Long Filenames

Upon identifying filenames that surpass the length limit, the script employs an abbreviation strategy to shorten these names. This process involves:

* Breaking down filenames into identifiable components
* Applying abbreviations based on a customizable dictionary
* Ensuring the uniqueness of the shortened names to avoid conflicts

A close-up of a diagram

Description automatically generated



### Main features – Shorten Long Nested Folders

For directories with paths that exceed the length threshold, the script executes a similar abbreviation strategy to shorten folder names within the path. Key steps include:

* Analyzing the entire directory path and identifying segments that can be abbreviated
* Utilizing the dictionary to replace long segments with their abbreviated forms
* Maintaining the structural integrity of the file system by carefully managing changes

A close-up of several arrows

Description automatically generated

# Getting Started

## Installation

1. **Prerequisites**: Ensure Python 3.6.4+ is installed on your system
2. **Script Download**: Download the latest repo of FilePathShortener from the official repository
3. **Configuration**: Extract the repo to your desired location and open the **config/config.ini** file in a text editor to adjust the settings according to your environment.

## Configuration

### config.ini

The **FilePathShortener** script is highly customizable through its **config.ini** configuration file. Below are the default settings and their explanations:

| **Key** | **Default Value** | **Description** |
| --- | --- | --- |
| base\_dir | C:\Users\JohnDoe\Desktop\LongPathTest | Root directory for the scan. |
| config\_dir | config | Directory for configuration files. |
| log\_dir | logs | Directory for log files. |
| output\_dir | output | Directory for output files. |
| dir\_scan\_dir | dir\_scan | Subdirectory for directory scan outputs. |
| filename\_scan\_dir | filename\_scan | Subdirectory for filename scan outputs. |
| filename\_length\_threshold | 150 | Max length for filenames before flagging/shortening. |
| dir\_length\_threshold | 200 | Max length for directory paths before flagging/shortening. |
| scan\_entry\_threshold | 50 | Max entries in a scan output file before splitting. |
| number\_of\_retry | 10 | Attempts to rename with conflicts before stopping. |
| dictionary\_path | abbreviation\_dictionary.csv | Path to abbreviation dictionary CSV file. |
| long\_dir\_path\_scan\_output | long\_dir\_path\_scan\_output | Prefix for output files listing long directory paths. |
| long\_filename\_scan\_output | long\_filename\_scan\_output | Prefix for output files listing long filenames. |
| long\_dir\_path\_modified\_output | long\_dir\_path\_modified\_output | Prefix for output files listing successfully shortened directories. |
| long\_filename\_modified\_output | long\_filename\_modified\_output | Prefix for output files listing successfully shortened filenames. |
| long\_dir\_path\_modified\_error | long\_dir\_path\_modified\_error | Prefix for output files listing directories that could not be shortened. |
| long\_filename\_modified\_error | long\_filename\_modified\_error | Prefix for output files listing filenames that could not be shortened. |
| dry\_run | False | Enables dry run mode; no actual changes made. |
| dry\_run\_dir | dry\_run | Directory for dry run reports. |

### abbreviation\_dictionary.csv

The **abbreviation\_dictionary.csv** file plays a critical role in the script's ability to shorten long filenames and directory names. It contains key-value pairs that define how specific words or segments within paths should be abbreviated.

**Format:** Each line in the CSV file should contain two values separated by a comma:

* The **key** (the original word or segment to be abbreviated).
* The **value** (the abbreviation to replace the key).

**Example:**

documentation, docs

information, info

configuration, config

**Guidelines for Customization:**

1. **Additions:** To add a new abbreviation, simply include a new line with the key-value pair.
2. **Modifications:** To change an abbreviation, update the value part of the respective key-value pair.
3. **Deletions:** Remove the entire line corresponding to the abbreviation you wish to delete.

# Usage Instructions

## Running the Script

### Scanning for Long Paths and Filenames:

* + To initiate a scan, use the following command:

python long\_filepath\_filename\_shortener.py -p scan

* + Alternatively, running the script without any options defaults to this scan mode:

python long\_filepath\_filename\_shortener.py

### Shortening Folder Names:

* + To shorten long folder names within the specified **base\_dir**, execute:

python long\_filepath\_filename\_shortener.py -p dir

### Shortening Filenames:

* + For shortening long filenames, use:

python long\_filepath\_filename\_shortener.py -p filename

## Option Flags

* **-p or --process**: Specifies the operation mode (**scan**, **dir**, or **filename**). If omitted, the script defaults to **scan**.

## Dry Run Mode

To preview the changes the script would make without applying them, enable the dry run mode by setting **dry\_run = True** in the **config.ini** file. This mode is particularly useful for reviewing potential changes and ensuring accuracy before making any modifications to file or directory names.

### Enabling Dry Run:

* + Open **config.ini** in a text editor.
  + Locate the **dry\_run** setting and change it to **True**.
  + Save the changes and run the script as described above to see what actions would be taken.

**NOTE: Remember to revert dry\_run back to False in config.ini when you're ready to apply the changes.**

# Understanding Output

The **FilePathShortener** script generates various output files to provide users with detailed insights into the scanning and shortening processes. Here's how to interpret each type of output:

## Scan Results

## The scan feature outputs result into two directories under the output folder:

## output/dir\_scan for long directory paths.

## output/filename\_scan for long filenames.

### Naming Format:

### Files are named according to the pattern: **long\_dir\_path\_scan\_output\_YYYYMMDD\_partX.txt** or **long\_filename\_scan\_output\_YYYYMMDD\_partX.txt**, where YYYYMMDD is the date of the scan, and X denotes the part number if the output is split.

### Contents:

## Each .txt file contains a list of file paths identified during the scan, formatted as full paths to ensure accuracy, such as:

\\?\C:\Users\John.Doe\long\_dir\long\_sub\_dir\long\_document-production\_release\_version\_1234.pdf

\\?\C:\Users\John.Doe\long\_dir\long\_sub\_dir\public\_presentation\_file.pdf

### User Actions:

Review these paths for accuracy and necessity. Paths can be edited or removed from the output files if certain files or directories should not be modified.

## Shorten Filenames Output

Post-processing, the script generates a list of files whose names have been shortened, stored in files like **long\_filename\_modified\_output\_YYYYMMDD.csv**

### Contents:

* The **.csv** output files detail each filename before and after the shortening process, separated by a comma:

Original Filename, Shortened Filename

## Shorten Nested Folders Output

Similarly, for directories that have been shortened, output is captured in **long\_dir\_path\_modified\_output\_YYYYMMDD.csv**

### Contents:

* Each line in the **.csv** output represents a directory path before and after renaming, formatted as:

Original Directory Path, Shortened Directory Path

## Dry-Run Shortening Output

When **dry\_run** is set to **True** in **config.ini**, the script simulates the shortening process without making actual changes, outputting the proposed modifications to **output**/**dry\_run**:

### Contents:

* Dry run outputs mimic the format of the actual shortening outputs but represent what would occur if the script executed changes, allowing users to review and approve modifications before implementation.
* So, again, for **Shorten Filenames** dry-run, the output would look similar to this

Original Filename, Shortened Filename

* for **Shorten Directories** dry-run, the output would look similar to this

Original Directory Path, Shortened Directory Path

# Troubleshooting and FAQs

## Common Issues

* **Python Not Recognized in Command Line:**  
    
  If you encounter an error message stating Python is not recognized as an internal or external command, it's likely that Python is not added to your system's PATH environment variable. Ensure Python is installed correctly and that its installation directory is added to PATH.

## Insufficient Permissions: Running the script may require specific permissions to read directories or modify files. Ensure you have the necessary permissions or run the script as an administrator/user with sufficient rights.

## FAQs

### **Q: What should I do if the script doesn't start?**

A: First, verify that Python is correctly installed on your system and accessible via the command line. Next, check that you are in the correct directory where the script is located. If the problem persists, confirm that all required configuration files are in place and correctly formatted.

**Q: How do I handle 'Access Denied' errors during scanning or shortening?**

A: 'Access Denied' errors typically indicate insufficient permissions to read a directory or modify a file. Try running the script with elevated permissions. On Windows, you can do this by opening your command prompt as an administrator. If specific files or directories are consistently problematic, check their security settings to ensure your user account has the necessary access rights.

**Q: The script runs, but no output files are generated. What could be wrong?**

A: Ensure the **output\_dir**, **dir\_scan\_dir**, and **filename\_scan\_dir** paths in **config.ini** are correct and that the script has write permissions to these directories. Also, verify that the **base\_dir** setting correctly points to the directory you intend to scan.

**Q: Can I exclude specific directories or file types from being scanned?**

A: While the current version of the script processes all files and directories within the specified **base\_dir**, you can manually remove entries from the scan output files before running the shortening process. Incorporating exclusion rules directly into the script could be a valuable feature for future versions.

**Q: How do I update the abbreviation dictionary?**

A: Open the **abbreviation\_dictionary.csv** file in a text editor or CSV editor. To add a new abbreviation, insert a new line with the format **term\_to\_abbreviate,abbreviation**. Save your changes, and these will be reflected the next time you run the script.



[**cgi.com**](https://www.cgi.com)