



VIS GLOBAL

Automation Anywhere Practice

Duplicate File Identifier Bot (Hash-Based)

Readme Document

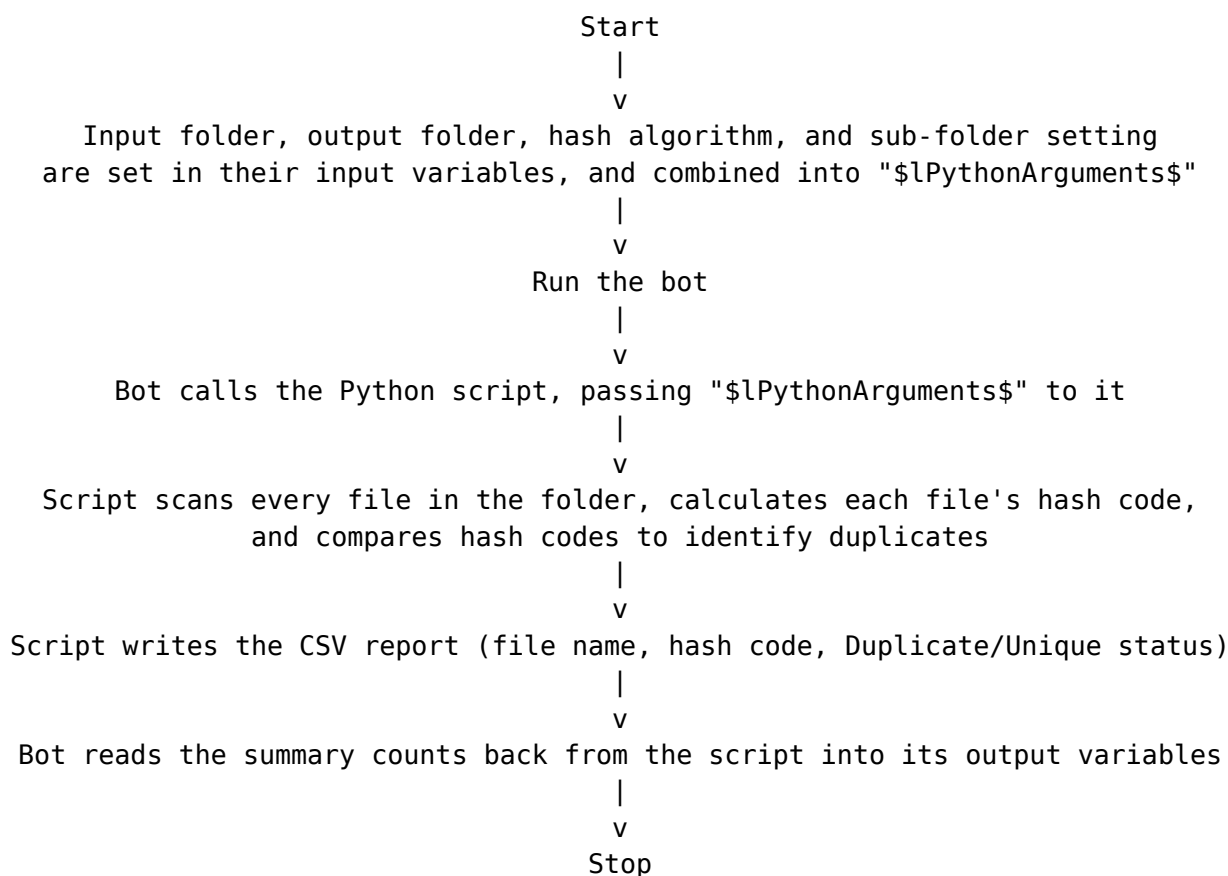
Version 1.0.0
08/09/2026

1. Introduction

1.1 Overview

- The Duplicate File Identifier Bot (Hash-Based) identifies duplicate files inside a given folder, based on file content rather than file name.
- The hash calculation and comparison logic is handled by a Python script, which the bot calls (passing the folder path as a parameter) using the Python/Script package.
- For every file in the folder, the script calculates a hash code from the file's content. Because the hash depends only on content, two files with completely different names but identical content are still correctly identified as duplicates.
- The script writes its results to a CSV report, listing the total number of files scanned, each file name, its hash code, and whether that file is a Duplicate or Unique.
- The bot reads the script's summary result (total files scanned, duplicates found) into its own output variables, so this information is available for use elsewhere in the automation — for example, in an email notification or a log entry.

Process flow:



1.2 Use Cases

The Duplicate File Identifier Bot (Hash-Based) can be used wherever files may have been saved more than once under different names — for example, shared drives, project folders, document repositories, or backup locations — and a reliable, content-based way of finding duplicates is needed rather than a name- or size-based comparison.

2. Requirements & Prerequisites

2.1 System Requirements

Automation Anywhere Enterprise A2019 (cloud-deployed) or Community Edition.

Review the hardware specification, operating system version, and browser support published by Automation Anywhere for building and running bots and command packages on the target machine.

2.2 Prerequisites

- Python installed on the machine running the bot, with the hash-comparison script (`duplicate_file_finder.py`) available at a known path
- Python/Script Package (to call the Python script and pass the folder path to it)
- File/Folder Package (to validate the input folder path and locate the generated CSV report)
- CSV/Excel Basic Package (to read the summary counts back from the script's CSV report)
- String Package

3. Getting Started

3.1 Quick Start

3.1.1 Setup

- Identify the folder that needs to be checked for duplicate files, and note its full path. This can be a folder on the local desktop, a mapped network drive, or a Control Room-managed location, depending on where the source files are held.
- Ensure Python is installed on the machine that will run the bot, and that the `duplicate_file_finder.py` script is present and accessible at a known path on that machine.

3.1.2 Configuration and Use

- Set the folder to be scanned as the value of the "\$strInputFolder\$" input variable (required).
- Set the folder where the CSV report should be saved as the value of the "\$strOutputFolder\$" input variable (required).
- Set the hash algorithm to be used (e.g. MD5, SHA-256) as the value of the "\$strHashAlgorithm\$" input variable (required).
- Optionally, set whether sub-folders should also be scanned using the "\$strIncludeSubfolders\$" input variable (True/False).
- These values are collected into the "\$lPythonArguments\$" list variable, in the order the script expects them, and the bot passes this list to the Python script when it is called.
- Run the bot. It calls the Python script with "\$lPythonArguments\$" and waits for it to finish.

This bot is a reusable component: instead of comparing files manually by name or by eye, the Python script compares the actual content of every file using its hash code, which means renamed copies, moved copies, and files with different extensions but identical content are all still caught. If the folder to be scanned, the hash algorithm, or any other setting changes, only the relevant input variable needs to be updated — no change is required to the bot's internal logic or the script itself.

The CSV report generated by the script includes, for every file scanned:

- File Name — the name of the file
- Hash Code — the content hash calculated for that file, using the selected hash algorithm

- Status — Duplicate or Unique, based on whether its hash code matches another file's

Variables passed to the Python script:

Variable Name	Type	Mandatory	Purpose
strInputFolder	String	Yes	Path of the folder to be scanned for duplicate files.
strOutputFolder	String	Yes	Path of the folder where the Python script saves the output CSV report.
strHashAlgorithm	String	Yes	Hash algorithm to use when calculating each file's hash code (e.g. MD5, SHA-256), passed to the script.
strIncludeSubfolders	String	No	Whether sub-folders should also be scanned (e.g. True/False), passed to the script.
lPythonArguments	List	Yes	Holds the above values collected together as a list, in the order the script expects them, and passed to it as its arguments.

Output Variables — Output variables of the bot:

Variable Name	Type	Mandatory	Purpose
strPythonResult	String	Yes	Holds the result returned by the Python script once it finishes running — including the duplicate-check summary and the path to the CSV report.
strTaskName	String	Yes	Name of the task; used to name the error screenshot with the task name.
strTodayDate	String	Yes	Stores today's date.
strIsError	Boolean	Yes	Returns True if an error is encountered.
strException	String	Yes	Holds the error message, if any.
strLineNo	Number	Yes	Holds the line number where an error occurred, if any.

4. Support & FAQs

4.1 Support

This is an internal VIS Global Pty. Ltd. utility bot. For support, contact the Automation Anywhere practice team responsible for the Agentic Bot Store submissions.

4.2 FAQs

To be populated with frequently asked questions relevant to this bot as they arise from usage.

Appendix A: Record of Changes

No.	Version Number	Date of Change	Author	Notes
1	1.0.0	08/09/2026	VIS Global Pty. Ltd.	Created

Appendix B: References

No.	Topic	Reference
1	Automation Anywhere Enterprise A2019 overview	Automation Anywhere product documentation portal
2	Python/Script Package and File/Folder Package guidance	Automation Anywhere product documentation portal
3	CSV/Excel Basic Package guidance	Automation Anywhere product documentation portal
4	Agentic Bot Store submission guidelines	Automation Anywhere Bot Store