PowerShell Script To Organize Files

Overview

You have been asked to create a PowerShell script to organize files.

Given a source directory with different types of files, your script should copy files to a destination directory into a subfolder named the same as the file's extension. For example, if the source directory is c:\source and the destination is c:\organized then all the files with a .jpg extension should be copied to c:\organized\jpg. All files with a .txt extension should be copied to c:\organized\txt etc.

After moving all of the files, your script should generate some statistics on the destination directories. For each folder the script should display the folder name, the number of files in that folder and the total size (e.g., number of bytes) of files in that directory.

Requirements

Your script should consist of multiple parts:
1) parameters that are passed to the script itself (so that it could be executed from the command line),
2) functions for code that is called repeatedly, and
3) the main processing of the code (i.e., code that is included in the script file that is not placed in a function.

1) Parameters passed to the script

   a) The script should take 2 arguments $source and $destination (for the source and destination folders).

2) Functions

   a) Create a function named *CheckFolder* that checks for the existence of a specific directory/folder that is passed to it as a parameter. Also, include a switch parameter named *create*. If the directory/folder does not exist and the create switch is specified, a new folder should be created using the name of the folder/directory that was passed to the function.

   b) Create a function named *DisplayFolderStatistics* to display folder statistics for a directory/path that is passed to it. Output should include the name of the folder, number of files in the folder, and total size of all files in that directory.

3) Main processing

   a) Test for existence of the source folder (using the *CheckFolder* function).

   b) Test for the existence of the destination folder; create it if it is not found (using the *CheckFolder* function with the –create switch).

   c) Copy each file to the appropriate destination.

i) Display a message when copying a file. The message should list where the file is being moved from and where it is being moved to.

d) Display each target folder name with the file count and byte count for each folder.