Pack N Hash 4.42

Release Notes & Guide

Release Notes V4.42:

03/14/24

- Hotfix adjusting commands for encryption passwords containing special characters that could create failures of 7z commands.

Release Notes V4.41:

02/10/24

- Archive testing functions now multithreaded. Internal testing showed approximately 30% reduction in archive testing time when compared to same sized case batch.
- Lots of additional cleanup of code and message formatting in output console.
- Further improvements to messaging in Activity Dashboard.
- Improvements to automatic disabling/enabling of processing buttons in UI at start/ end of various processing tasks.

Release Notes V4.4:

02/04/24

- Major UI Updates:
  - Refreshed UI
  - Significant recoding of subprocess modules eliminating secondary console windows and all 7zip status messages now piped into integrated 'Activity Dashboard'.
- Advanced source queuing and case information customization inspired by the updated queuing system 'BFIP', now integrated. Allows for user to verify what case folders are queued for archiving before starting processing, as well as adding cases from different drives to a single queue.
- Lots of updates to messaging and logging to take advantage of updated activity dashboard, and to minimize redundant notifications.

- Archive testing has gotten a significant update. While archives generated from PackNHash have always been automatically tested for integrity and CRC errors, this information and the results are now brought front and center to the user's attention with clear status indications for each tested archive. This new results window will be presented at the conclusion of any full PackNHash, or standalone Archive Test passes.
- API for automatically checking for updated versions of PackNHash now integrated. If connected to the internet, available updates will be queried on startup, or can be triggered from right click menu.

Release Notes V4.3:

02/28/23

- 7z Libraries Updated to 22.01
- Updated Dependencies to Improve STDOUT Readability in console window during file integrity testing.

Release Notes V4.2:

07/03/22

- Hashing Engine Rebuilt to now conduct multithreaded parallel hashing of archive segments. Direct tests between prior hashing engine and new multithreaded version yielded a *400% increase in speed.*
- 7z Binaries now directly integrated and no further 7Z installation necessary.
- UI Refinements including update console output that dynamically resizing when program window is maximized.
- Right Click Cut, Copy, Paste, and Help options.
- Integrated User Manual into program.
- Logs Moved to users appdata consistent with other Breakpoint Forensic Tools and button to open Logs folder added to UI.
- Other miscellaneous code cleanup, and improvements.

Release Notes V4:

03/21/22

- Massive UI Update – Completely rebuilt from ground up
- All console entry is completely banished and handled through 100% GUI-based elements
- Exposed additional archive settings for optional configuration of:
    - Compression Level/Speed
    - Archive Encryption
- Common user configuration settings are now automatically saved and automatically set for you the next time PackNHash is run.
- New Validation Section allowing individual testing, file list generation, and hashing tasks to before performed on their own.
- Huge speed update to entire process when using recommended settings, that can yield 400% increase in speed over entire PackNHash process.

- New integrated console for cleaner feedback on processing progress with completion percentage updates provided throughout various stages.
- Refactored hashing process that now results in a single CSV file containing hash values for all segments of a case-folders archive.

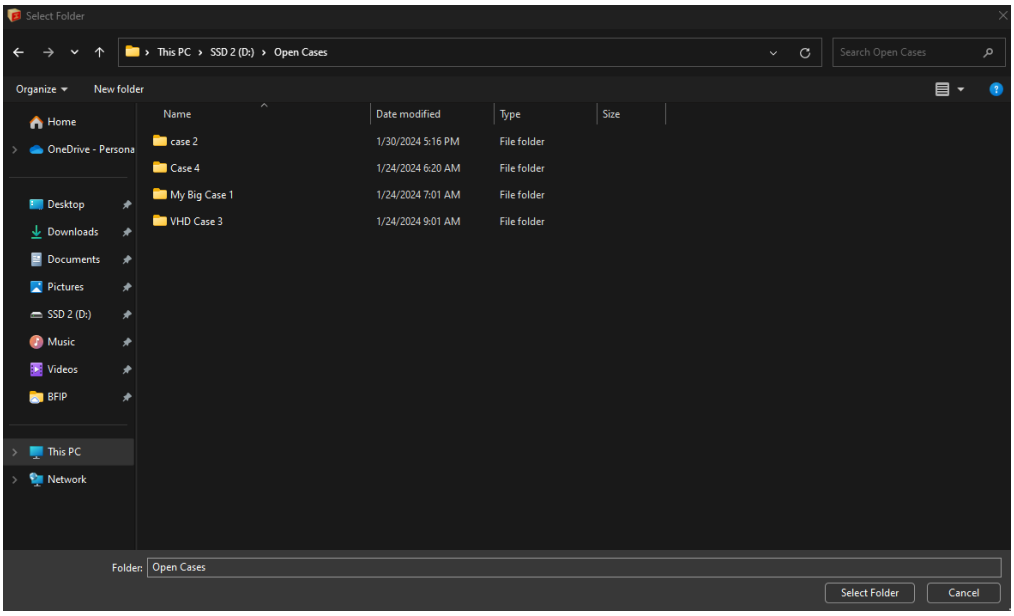
Requirements:

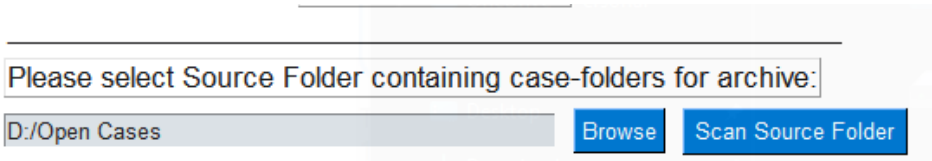
- Windows 7, 10, 11

Archiving Guide

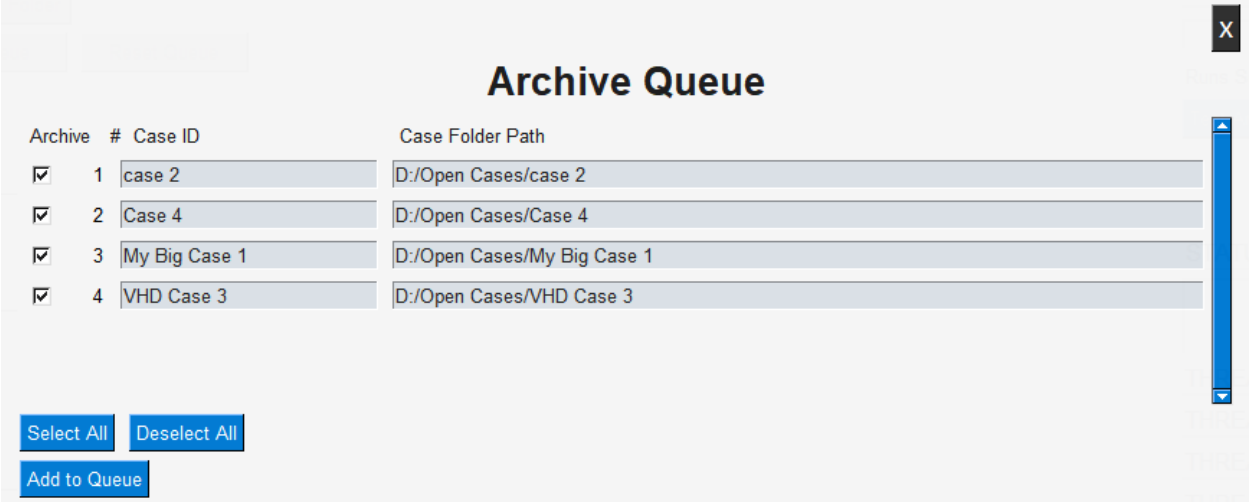
1. Select the Source Folder containing ***all your various case folders*** requiring archive.

   Note: Cases requiring archiving should be placed inside their own 'Source Folder'. This source folder is then what is selected in the folder chooser. *Do not select an individual case folder in the Source Folder Selection.



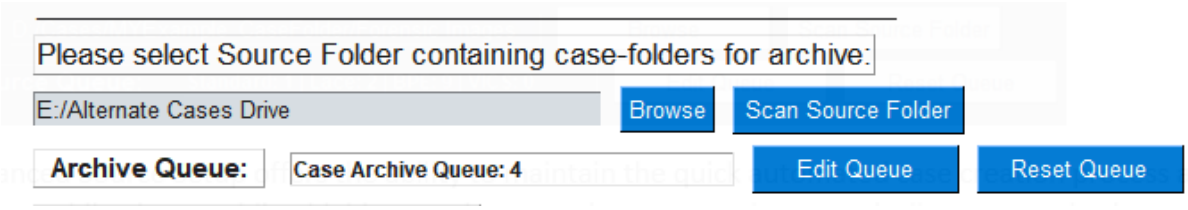


2. Click ‘Scan Source Folder’. PackNHash will then automatically identify all the individual case subfolders one level down, and will present located folders in a new Archive Queue window where the user can except the defaults, deselect any cases you do not wanted archived, or can change the final archive name.



3. Adding additional files to queue (Optional):

   The user may repeat this workflow adding a different ‘Source Folder’ to scan, pressing ‘Scan Source Folder’, and adding case folders to the queue multiple times. This allows for stacking multiple case folders scattered amongst several different folder locations to a single queue.



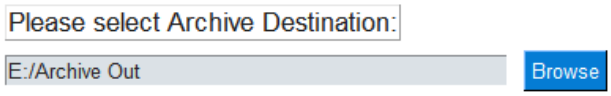
4. Edit Queue

   This opens a menu containing all previously selected cases and their current settings that have already been added to the queue using the ‘Scan Source Folder’ button. Existing folder paths, case ID’s, etc., can all be further edited, removed or changed from here.

5. Reset Queue:

   This function completely clears the Case Archive Queue.

6. Archive Destination
   Designate a destination folder for where to send the archives. To avoid unnecessary additional hashing, please make sure the destination directory is initially empty.



7. Configure optional segment size and compression settings.

   a. **Segment Size**
      Configurable in increments of Gigabytes
   b. **Compression Levels**
      There are 10 levels 0-9.
      Level 0 = No Compression and is the fastest.
      Level 9 = Most Compression and will take the longest to archive.
      Level 1 is set as initial default and offers reasonable compression and good speed.

   c. **Encryption**
      Check box and enter password to encrypt with AES encryption

8. Click 'Start' to Initiate Complete PacknHash Process

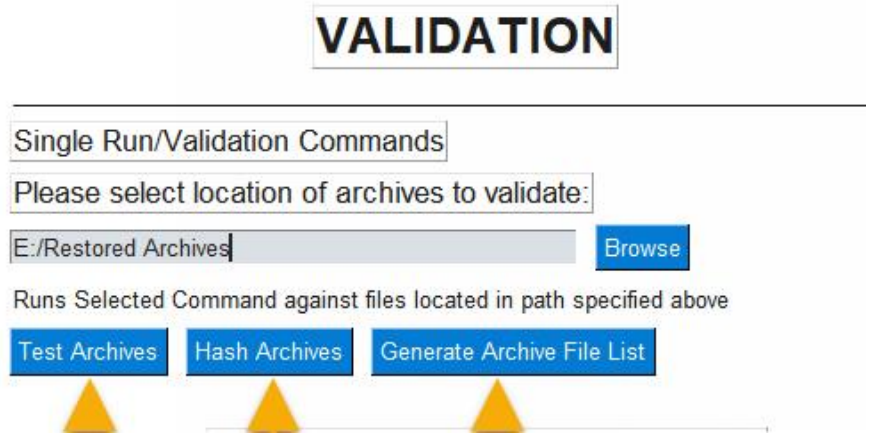
   <u>This will include:</u>

   i. Package each case folder into individual case subfolders and archives.
   ii. Each generated archive will then be tested to ensure it can be opened and all files are intact.
   iii. A file listing text file is generated for each individual case archive.
   iv. All archive segments are MD5 hashed and logged into a CSV file in the case archives destination directory.

Single Run/Validation Commands

PacknHash includes to option to selectively execute the various validation process from the full PackNHash process on an individual as-hoc basis.

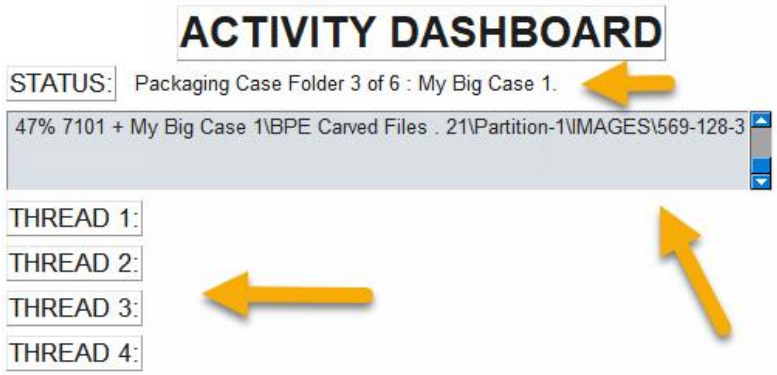
This includes the ability to select either a parent folder containing several subfolders of various cases, or you can select an individual case archive folder containing your packed archives. PacknHash will recursively search for all archive segments within the selected folder and conduct the selected validation process.



Output Window and Activity Dashboard

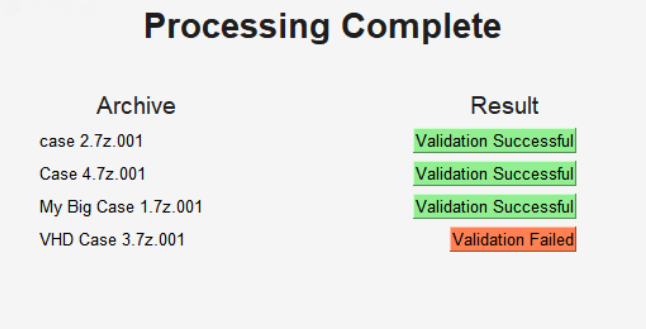
PacknHash utilizes an integrated output window and Activity Dashboard. Various confirmation messages, processing status, errors, etc. will be printed here for reference.

These areas provide a dynamic status messages that will occasionally update with hashing progress indicators and completion percentages.



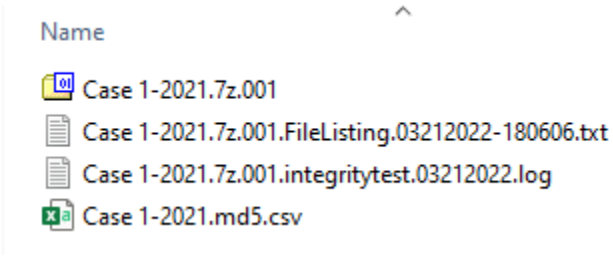
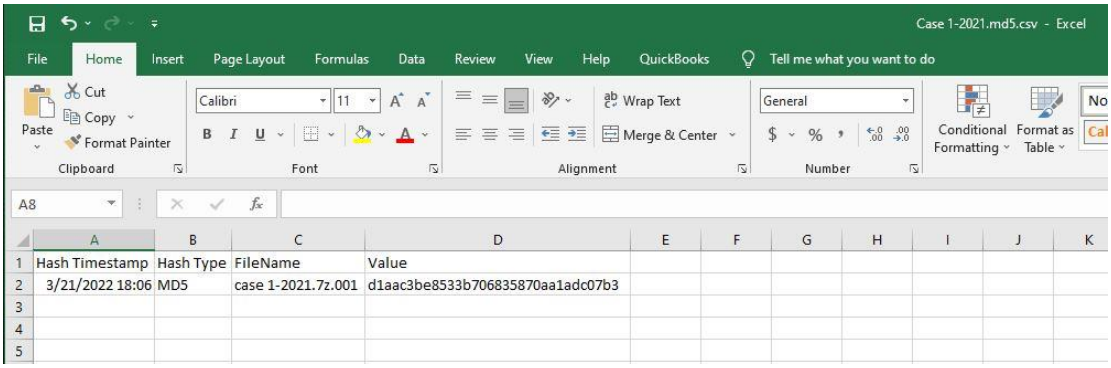
Result Window

At the conclusion of any full PackNHash or Testing run, the validation result for all generated/tested archives will be presented in a final window, along with any warnings. Validations are triggered when an archive or one of its segments is corrupted or a missing.



Generated Log/Validation Files

Inside each case folder will be 3 different log files:



1. A file Listing containing a full list of all files that were added to archive.
2. An integrity test log indicating the result of the final integrity test of the completed archive.
3. A CSV holding MD5 values for a segment of the archive.