PeaZip

Open source, portable file and archive manager

Document revision: 2024 04 25

Author: Giorgio Tani

Translation: Giorgio Tani

This document refers to:
PeaZip 9.8.0

Licensing:
Present documentation is released under GNU GFDL License;
PeaZip executable implementation is released under GNU LGPL License.

PeaZip official website:
https://peazip.github.io , and https://peazip.sourceforge.io

For more information about the licenses:
GNU GFDL License, see https://www.gnu.org/licenses/fdl.txt
GNU LGPL License, see https://www.gnu.org/licenses/lgpl.txt

Content:

What is PeaZip

PeaZip is a general purpose file and archive manager application for Linux, macOS, BSD, and Windows, aiming to provide a cross-platform graphical interface for many Open Source archiving and compression utilities in order to handle most of available archiving formats like 7Z, RAR (extraction), TAR, ZIP and many other ones, see **Supported formats** chapter for more information.

The program features powerful and flexible inclusion/exclusion filters and search tools, provide optional two factor authentication through password and keyfile, and allows deeply fine tune the task's definitions, exposing through a single, consistent frontend GUI the options of underlying applications.

The list of objects to be archived or extracted (Layout) can be saved for future use, to speed up backup and restore tasks.
Also the resulting command for archive creation and extraction operations can be saved as script, to get the full control on task's definition, helping the user in bridging the gap between GUI and console applications to get the best of both worlds.
A detailed log is available after each operation.

PeaZip also collects a set of handy file management tools: robust file copy, split and join files, fast or secure file deletion, calculation of a wide set of checksums and hashes over selected files, byte-to-byte comparison of two files, web search etc.

PeaZip can be used as file manager, or can be used from **context** and **SendTo menu**; file associations and menu entries (both for context and SendTo menu) can be changed running the setup program any time it is needed.

If no system integration is preferred, **PeaZip Portable** is available as standalone application, not needing installation and not modifying the host system; both packages are available on application's website.

All the open source backend binaries included in base packages contain only open source software released under OSI-approved licenses.

Optional plugin based on closed source binaries are available as separate add-ons only, and can be browsed on PeaZip official website Add-Ons page.

How to...

This mini-tutorial introduces the most common operations that can be performed through PeaZip, following chapters contains a more detailed explanation.

Extract archive(s)

- **From the system**
    - o Right-click on the archive(s) and in system's context menu click "Extract here", "Extract here (smart)" or "Extract here (in new folder)" to extract with no further interaction.
    "Extract here (smart)" works as "Smart new folder" option in extraction screen, avoiding creation of a new extra directory if a single file or folder is extracted
    - o Otherwise use "Extract..." menu entry for having more options: output path, password, extract files to a new directory, chose to skip, rename or overwrite existing files etc.
- **Open an archive in PeaZip**, i.e. with double-click, or drag an archive on PeaZip's window or icon
    - o click on "Extract" in toolbar or in context menu to extract only selected objects trough the confirmation dialog allowing to set all options (output path, password, naming policy, extract to a new directory etc)
    - o click "Extract all to..." button (shortcut Ctrl+E or F12) to directly extract the entire archive to the specified folder, or click the quick extraction dropdown button on its right to extract into most common destinations skipping confirmation dialog, and to set most important options; shortcuts: Ctrl+Alt+E extracts archive in its current folder; Ctrl+Shift+E extracts to desktop; Ctrl+Alt+Shift+E extracts to documents; Ctrl+0 extracts to the previous path, Ctrl+1..8 extracts all to 1..8 bookmark's path (if defined)
    - o drag and drop files and folders to the desired destination
- **Open PeaZip**, select one or more archives and click "Extract" in toolbar or context menu, or use quick extraction destinations as explained at the previous point.

When a **folder** is sent to extraction procedure, PeaZip browses it recursively to find and extract all the archives it contains.

*Hint: to **extract multi volume (spanned) archives, place all volumes in the same folder** and use "Extract all" (from context menu or program's GUI) on the first volume, which is usually identifiable for the extension .001, or .r01, .z01 and similar.*

Extract selected content from archive
Open an archive in PeaZip, i.e. with double-click, or drag an archive on PeaZip's window or icon
- click "Extract" in toolbar: only selected items will be extracted
- drag files and folders to the desired destination, only selected content will be extracted
- right-click and in "Extract" group of the context menu click "Extract selected" (to extract only selected items) or "Extract displayed" (to extract content of current directory or of current search filter)

Browse or extract password protected archive
Click on the padlock icon to enter a password and optionally a key file. The icon is featured both in PeaZip's file browser's status bar and in the archive extraction interface; once the password is set the icon will change colour.

Create archive(s)

From the system, rightclick on files/folders and click on "Add to archive" in context menu or SendTo menu, it will open the archive creation confirmation dialog, more options are available in "Advanced" tab; click OK to create the archive.
Alternatively, drag files/folders to PeaZip's window or shortcut; the same archive creation interface will be shown.
From PeaZip select objects to be archived and click on "Add" button; the same archive creation interface will be shown.

Add content to existing archive (write-supported formats)
Open an archive in PeaZip (i.e. with doubleclick, or drag an archive on PeaZip's window or icon), then drag files and folders to be added in the archive (or click on "Add" button and use application's context menu to add objects to the archive). It will open the archive creation confirmation dialog, click OK to update the existing archive.

Edit files inside existing archive (write-supported formats)
Open the archive in PeaZip, double click to preview desired file(s), which will be extracted to a temporary work path.
If a file is modified on the disk, the application will automatically ask to update the archive with the new version.
If synchronization is declined, choosing "No", temporary file(s) will be cleared from disk.
"Yes / Clear" will update the archive and delete preview file(s) from disk, choosing "Yes" will update the archive and preserve preview files, to be able to re-open them faster for further edits.
From context menu More, "Update edited files in archive" group of entries it is possible to manually request to update the archive and/or clear preview files.

Span archive in smaller files of a pre-defined size
While creating an archive as explained in previous points, click on "Single volume" dropdown menu to select a size for output files (volumes) the archive will be split in. Most common types of archives supports this option.

Create encrypted archives
Click on the padlock icon to enter a password and optionally a keyfile; the icon is featured both in PeaZip's file browser's status bar and in archive creation interface.
To hide names of files and directories contained in the archive check "Encrypt also filenames", please note that it will be applied only if the archive will be created in a format supporting this function, like 7Z and ARC.
Note that in archive creation interface, alongside the padlock icon, there is a colour line (red for weak password, green for strong password) at the bottom of the form to visually inform if encryption is set and if the current archive format supports encryption.

Archive items in separate archives at once
Add objects to be archived (with PeaZip's "Add" button, or from system's context menu or SendTo menu) and before confirming with "Ok" check "Add to separate archives" option

Convert single or multiple archives
From PeaZip select archives to be converted and click on "Convert" on toolbar or context menu, non-archive files and folders can be added as well, the difference being in archives being extracted before compression stage. Using the "Convert existing archives" switch in conjunction with "Add to separate archives" (default) it performs a mass conversion of listed archives, without that switch archive conversion is meant to consolidate input data in a single archive, improving compression efficiency because it allows recompression of the original data from its uncompressed form.

Directly create an archive of specified format with given compression level
It is possible to directly add files/folders to a new archive of a specific format, and with specific compression level, from presets in the dropdown menu on the right of Add button. Most common presets are also available during installation / configuration as context menu entries

Configure PeaZip

- Options > Settings > Localization (in General tab) allows to set application's **language**, and translates PeaZip entries in system's context menu
- Options > System integration configure file associations, context menu and SendTo menu entries (on Windows)
- Options > Settings allows to edit applications' settings and Theme (application's icons and appearance)
- Organize menu contains entry to customize browser's features, like toolbar, address bar, side bar, etc

Please note some context menu entries, meant to provide quick access to a minimum core of features of the application (**extract, add to archive, and open as archive**), cannot be deselected during installation. Anyway context menu integration can be unchecked, preventing any context menu entry to be created.

Export tasks as scripts

PeaZip is meant to allow to easily export the tasks defined in the GUI as **command line scripts,** from Console tab in archive creation and extraction screens, see "Save and schedule tasks" chapter for more information.
This allows to thorough inspection, later re-use, customization beyond the capability of the UI, and helps learning the syntax of the underlying binaries.
Also PeaZip itself is easily scriptable, see "Customisation and scripting" chapter for directly accessing PeaZip functions from the command line.

Quick access to PeaZip functions

Drag & drop files and folders to PeaZip or a shortcut to a function, e.g. Add to archive, or Extract here
You can **drag and drop** single or multiple items (files and folders) **on PeaZip, or to a shortcut of PeaZip (which can point to a specific function)**, to easily pass the entire list of files & folders to the application.
Dropping files & folders on a PeaZip's shortcuts which points to a specific function, e.g. Add to archive, Add to ZIP, Extract here, Open as archive, Extract here (smart new folder), Extract…, etc **immediately send the item(s) to such function**.
This method works on all supported platforms, and also works for PeaZip Portable packages as it does not require any kind of system integration – and does not leave tracks on the host system.

Set custom default action double-clicking an associated file, e.g. extract here instead of open
From Options > Settings, Archive manager tab is possible to define **Default action on input** for the application, **the action which is performed when a file associated with PeaZip is opened with double click**, or when single or multiple files associated with PeaZip are opened from "Open" context menu.
By default action is set to open (browse) the input file(s), but it is also possible to directly send them to the extraction screen (selecting Extract… action instead of Open), or even to set the application to directly extract input files as it usually done by archive managers on macOS or Gnome, selecting "Extract here", or "Extract here (smart new folder)".
In this way, double-clicking a file associated with PeaZip, or using a generic "Open" context menu item, the content of the file(s) will be immediately extracted without requiring further interaction.

Use system-wide keyboard shortcuts
On Windows you can create **system-wide keyboard shortcuts** pointing to any of PeaZip's internal functions made available as shortcuts, as ones mentioned in the previous paragraph.
Righ-click on one of shortcuts, click Properties, in Shortcut Key field press the key combination (e.g. Ctrl+Alt+E for Extract…) you would like to use to directly access the function.
Unfortunately this method does not allow to pass selected items to the function, but it is still useful to open a specific PeaZip's function directly from the keyboard – e.g. to open an empty archive creation layout, or an empty archive extraction screen, in which input files & folders can be dropped from the system.
For the installable version it is also possible to access to PeaZip's shortcuts from the context menu, or from application's folder (open command prompt and type: explorer shell:AppsFolder).

Context menu

PeaZip can be integrated with Linux, macOS, and Windows contextual menu (right click).
Sample scripts to customize system integration are available in (peazip)/res/share/batch folder, being (peazip) the folder where PeaZip is installed, or where PeaZip Portable is placed – more information is available on Customization and scripting chapter.
Fore reference usually PeaZip is installed in:

- Linux: /usr/share/peazip for non-binary files as the aforementioned batch scripts
- macOS: /Applications/peazip.app (right-clicking on peazip.app it is possible to browse PeaZip's directory)
- Windows: C:\Program Files\PeaZip\ (or different drive letter if system disk is not C:)

On **Linux** systems, .desktop files are provided to integrate the application with most Desktop Environments (Cinnamon, Gnome, KDE, Mate, XFCE).
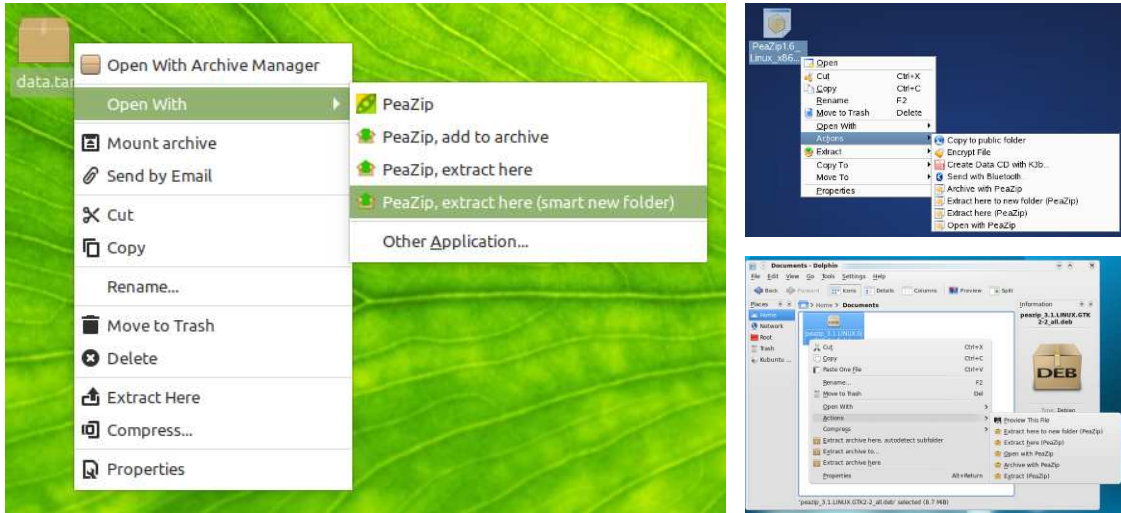

Image 1: examples of menu integration in Linux; see (peazip)/res/share/batch/ for sample scripts; freedesktop_integration/ subfolder contains .desktop files and other scripts allowing to integrate the application on Gnome, KDE, Xfce, Cinnamon and other DE following freedesktop standards.

On **macOS**, .workflow Automator scripts for Service menus are made available in the root of the DMG, installation is optional.
To install a .workflow item, double-click on it.
To uninstall a .workflow item, follow system's instruction from context menu customization entry, or simply delete it from ~/Library/Services/ directory inside user's home.
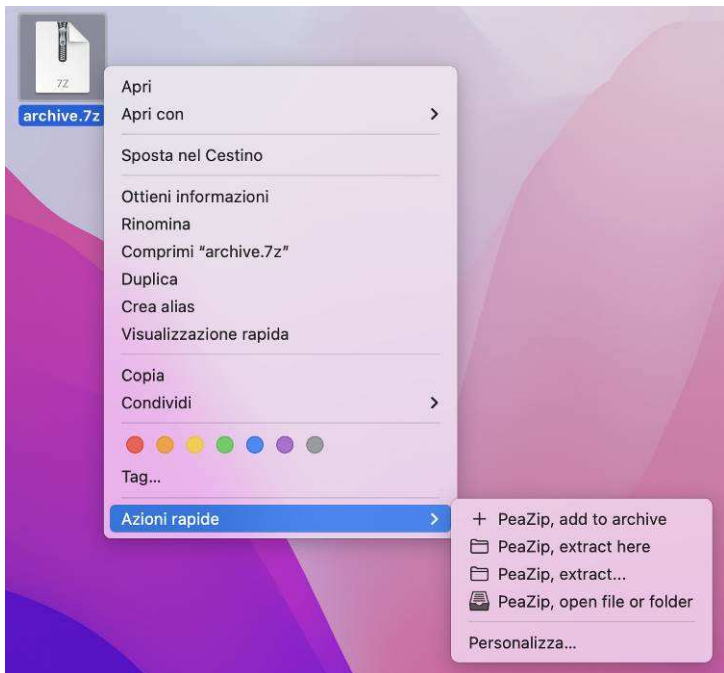A backup copy of the scripts is available in the aforementioned (peazip)/res/share/batch folder

Image 2: Service menus on macOS Monterey

On **Windows** platform, the installer takes care to create context menu and SendTo menu items, which can be customized during the installation, or later running main menu > Options > File associations and system menu integration wizard.

The Windows context menu items type created by PeaZip **can accept up to 99 selected items at once as input**, and should not be displayed by the system otherwise (but this latter behaviour may change on different Windows versions and updates).
As alternative, you can either drag and drop 100+ items from the system to PeaZip, or (faster alternative) select them from PeaZip's own file manager - hint: to directly open current path in PeaZip use system context menu entry "Browse path with PeaZip".

On Windows 11 it is possible to add PeaZip entries to the mini context menu (Image 3b) with simple .reg scripts which uses whitelisted IDs, e.g. SetDesktopWallpaper.
The scripts are provided as example in the aforementioned (peazip)/res/share/batch folder, and are natively capable to pass multiple input items to a single program's instance.
The scripts can be modified to be used with PeaZip Portable too, pointing them to the actual path of the application (as well as the Linux and macOS context menu integration scripts).
It is possible to localize the language of the menu entries changing the text in MUIVerb entries of the scripts.
All context menu and SenTo menu entries are available from "Show more options", or Shift+F10, in Windows 11 full context menu, and are not affected by mini context menu scripts.
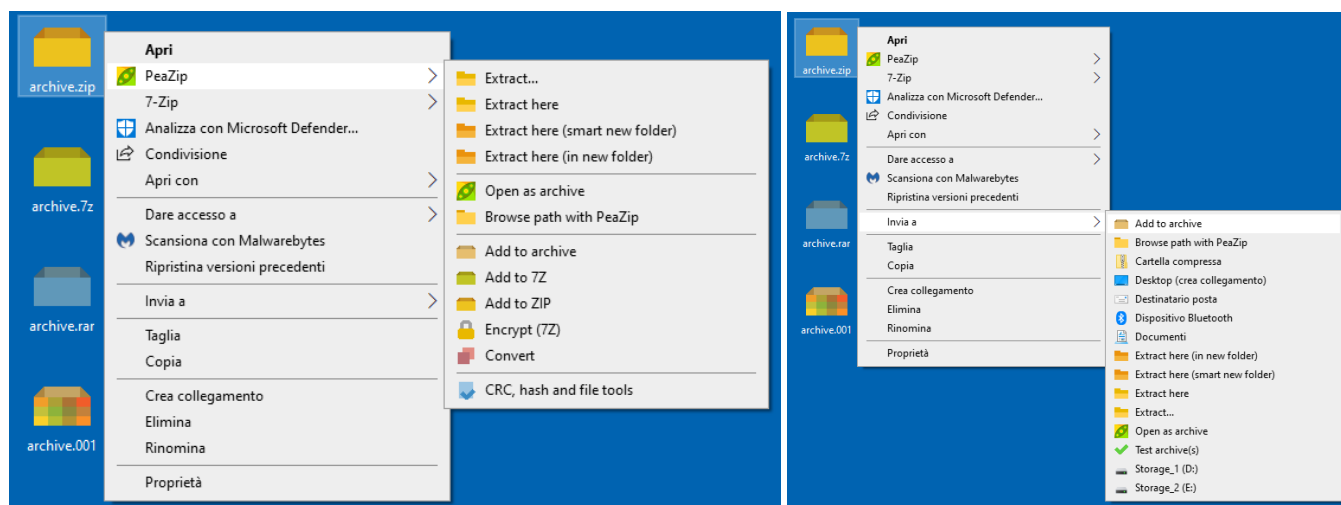
Image 3: high resolution icons, SendTo and context menu integration.
The context menu is optionally cascaded, only for Windows 7 or newer systems that supports cascading menus and display icons through static registry entry, which is now recommended over COM methods.
System integration can be customized running the installer or in PeaZip in Options > System integration, choosing "Custom installation".
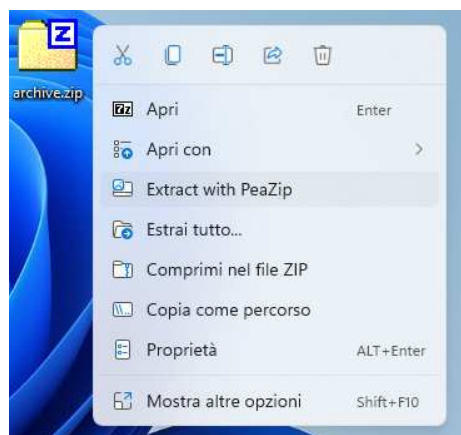

Image 3b: Windows 11 mini context menu integration

Dark mode

PeaZip uses system's colors, so it does natively support dark mode on all platforms.
On **BSD**, **Linux**, **macOS** and **Windows** it is sufficient to set the system to use a dark theme to use same colours on PeaZip.
In some instances this may not happen, if the app is running in a way which prevents it to receive correct system colors - virtualization, emulation, containers, some package formats (e.g. Flatpak), depending on how the host system is configured.
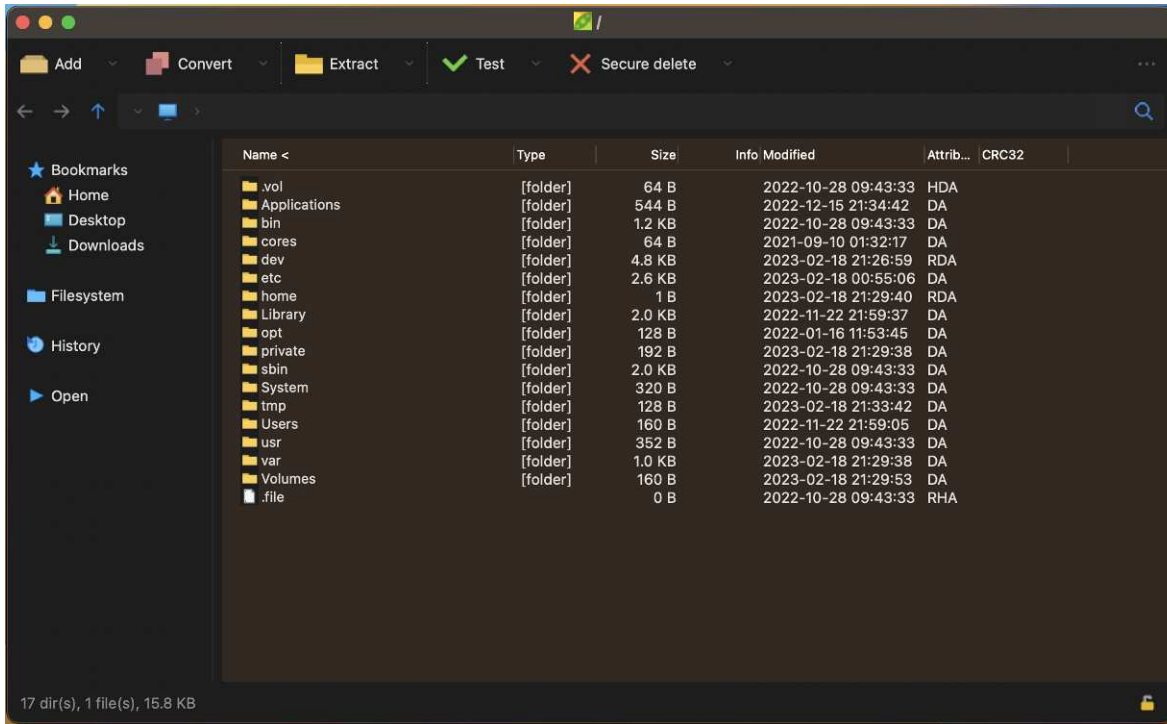
Image 4: Peazip in dark mode on macOS
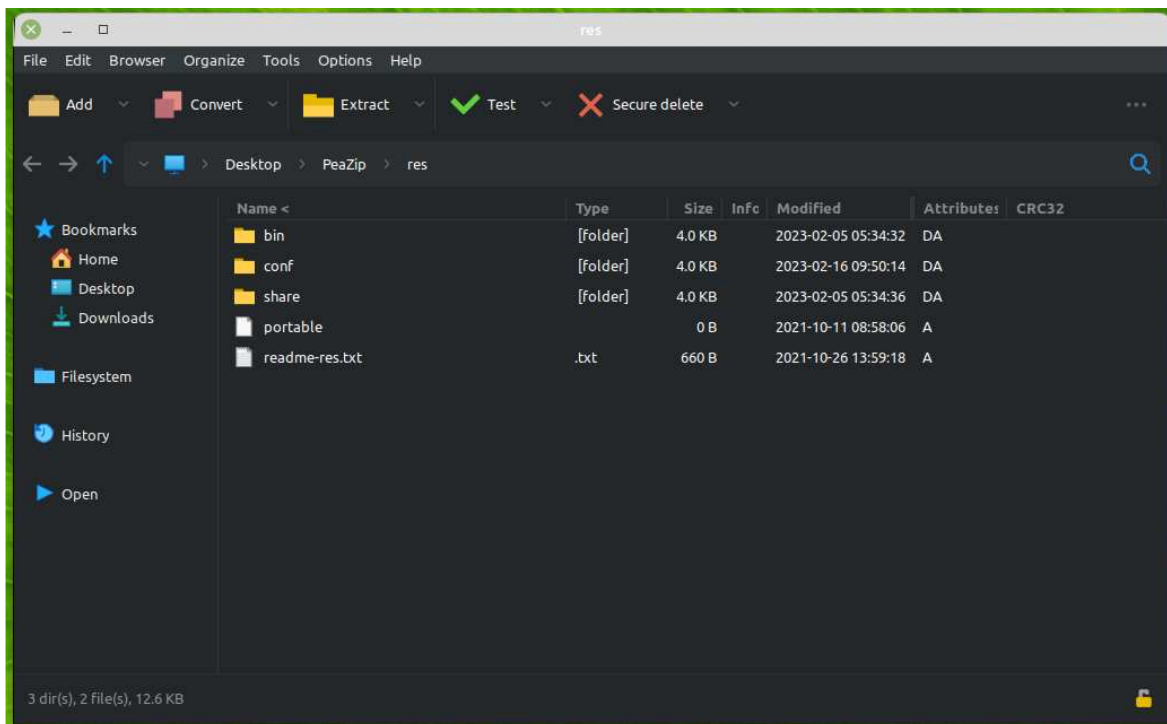
Image 5: Peazip in dark mode on Linux Mint

Please note that on Windows 10 and more recent systems the dark mode is set separately for "modern" apps (from Personalization > Colors) and for Win32 apps (from Settings > Ease of Access > High contrast) and for
PeaZip supports both methods, the first provides better look&feel, while the second allows to apply dark color scheme to all Win32 apps at once, even ones not supporting the new dark mode method.

On older Windows systems it is sufficient to set a dark Windows Theme (or a dark high contrast theme) and PeaZip will apply the same dark colors scheme.
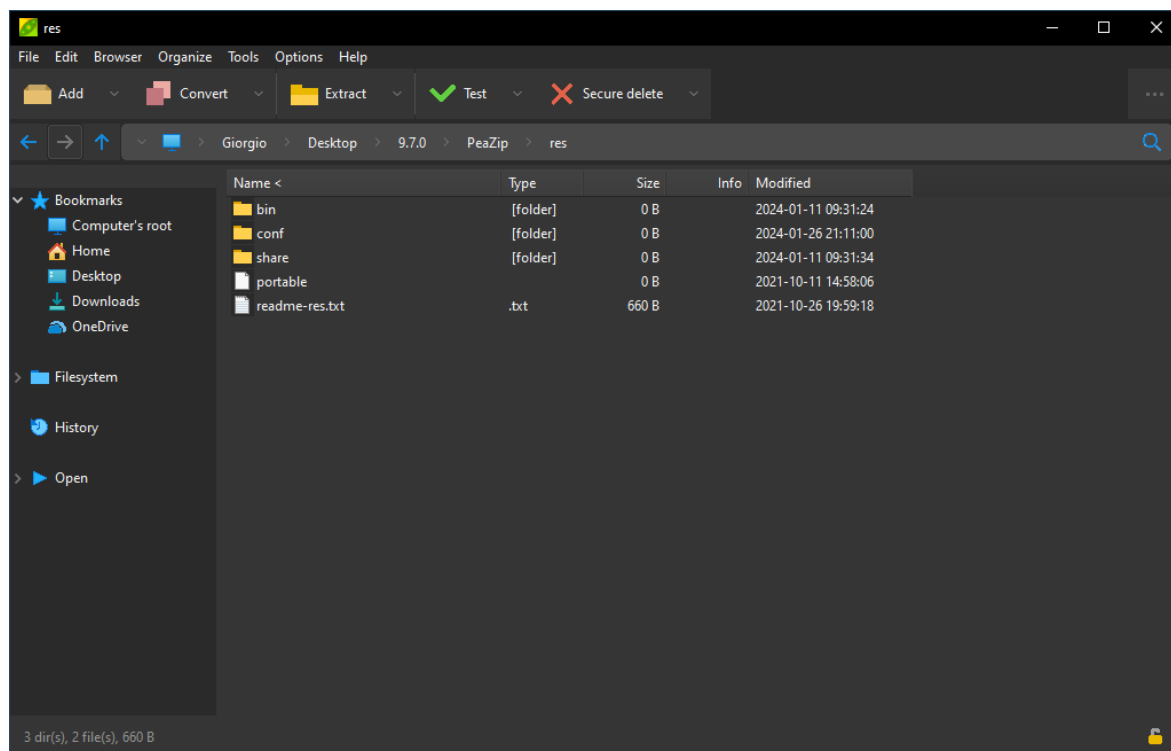
Image 6a: Peazip on Windows in dark mode

It is also possible to **manually force PeaZip to run in light or dark mode** regardless the system settings, creating an empty file named "light" or "dark" in the same path of the executable (this mechanism is not integrated with Themes due to technical limitations).
As alternative method to reduce the brightness of PeaZip's app it is possible to change background color from main menu Options > Settings, Theme tab, from Color control in Advanced (hidden) group.
On the right of the Color button it is available a list of pre-set values to quickly reduce brightness of the app, either with a neutral grey tone or applying tints like sepia, plum, etc.
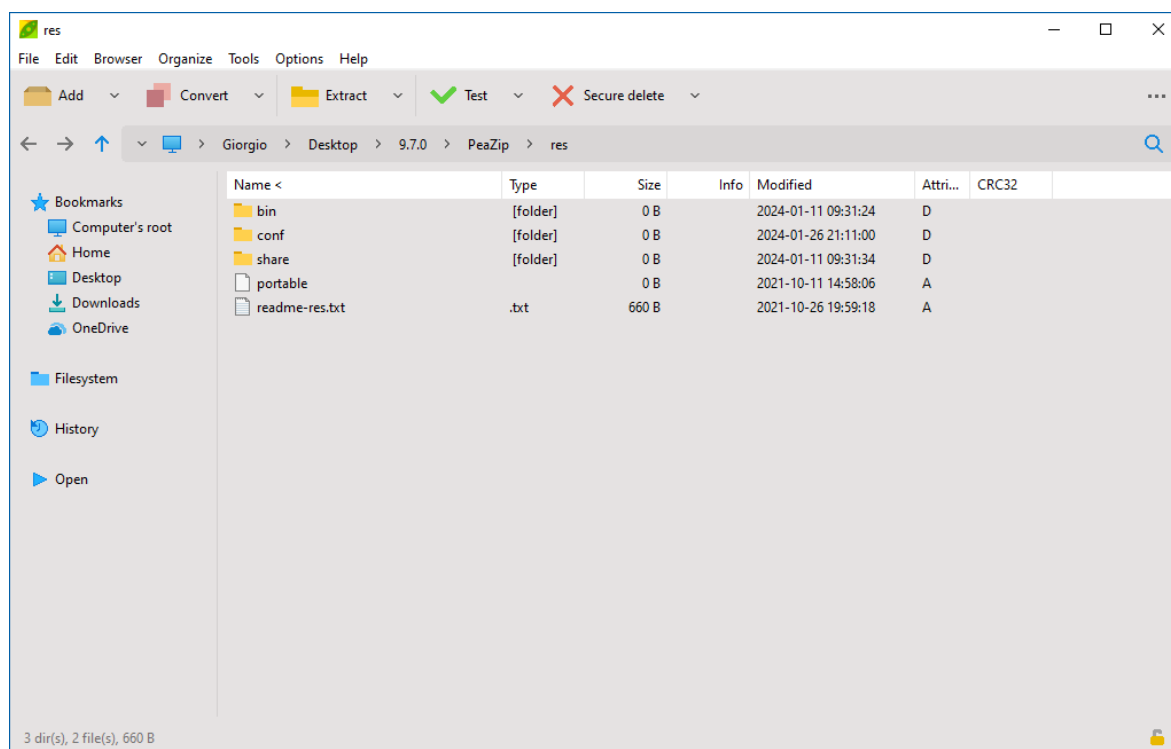
Image 6b: Peazip with "Mocha" warm gray modifier applied to background color.

File manager

The application starts by default with **file manager** interface, pointing to the last visited directory, for navigation in the filesystem and in archives.

*On start-up, PeaZip parses the input parameters to understand to what function they should be passed to (i.e. to open an archive for browsing, or adding selected objects to a new archive), see "**Customisation and scripting**" chapter for the startup parameters that can be passed to PeaZip, to use it in scripts or customize system's integration (registry entries, SendTo menu links etc...)*
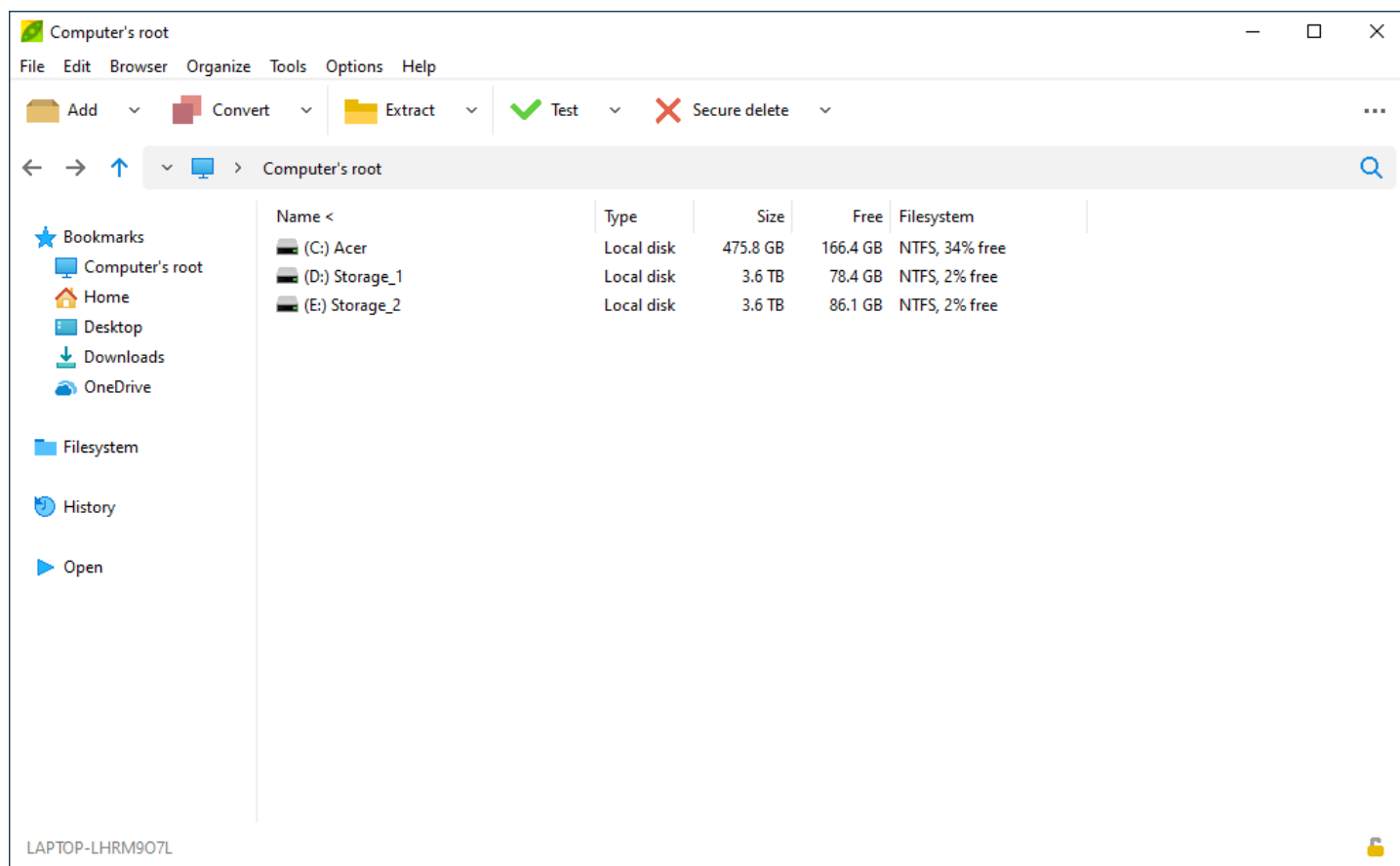

Image 7: PeaZip's file and archive browser, "…" Style button on the right allows to customize application's GUI.
On the left, the **navigation side bar**.
"Bookmarks" group contains user's bookmarked archives and folders
"Filesystem" group shows a tree of mounted units and default system's locations
"History" shows recently browsed archives and folders
"Open" group launches directory selection, file selection, and search dialog, and contains "Functions" (alphabetically sorted PeaZip's functions) and "Applications" group - launches favourite applications, scripts or websites/webservices that can be defined by the user (defined in Settings > Applications).
When browsing an archive, its directory tree is shown above "Bookmarks" group for fast exploration of archive content.

Main menu

File submenu, contains primary application's functions, it can be hidden / shown with Alt+F1.
Create archive activates the archive creation interface, which allows to add files and folders to the archive's layout and to save, restore and merge layouts for further use.
In same menu are featured Bookmarks, history (can be switched off for privacy in Options > Settings); Open path and Open archive entries, invoking standard file and directory selection dialog respectively; Search and drag here which open standard system's search dialog, to search files on the system and then drag them to PeaZip.
"Web search" allow searching for the selected filename (or an arbitrary string) on different web based services, as Google and Yahoo search engines, Wikipedia, etc, optionally on multiple websites at once.
Web search feature can help users in case of any doubt or need of any additional information about the object before archiving it or before extracting it from the archive, or for comparing an object (date, size, checksums etc…) with other known online sources.
Web search can also help in finding file names (i.e. to replace broken/incomplete downloads) on various sources: search engines, social networks, torrents, software, wiki etc.
Please remember to download only trusted files from trusted sources, and to refer to the respective Author of each content for any information about the copyright status.

Reduce to tray entry send PeaZip to tray area; right clicking on the tray icon it is possible to resume the program, or access most common program's functions.
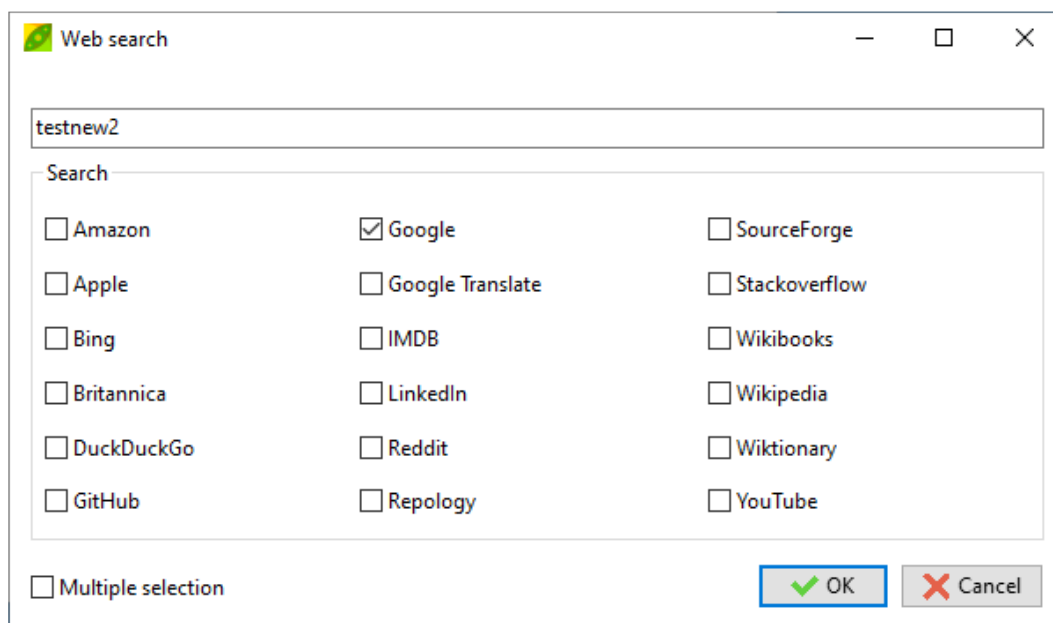

Image 8: web search form, optionally it is possible to search on multiple engines at once

Edit submenu changes the selection of currently displayed files, for date, size, extension, attributes etc...
It works either for the file/archive browser or for the archiving and extraction interfaces.
Select... entry shows a dedicated **advanced selection dialog**, featuring multiple selection options and the ability to add to / subtract from current selection. It is available also in context menu, File manager tool bar, and Search dropdown menu.

Browser submenu, featured only when file manager is displayed, contains
- jump to archiving and extraction layout interfaces
- search (in PeaZip), F3
- up one level (Alt+Up)
- go to archive's or system's root level (Crtl+Shift+F4), Desktop (Ctrl+F4), Home (Shift+F4)
- **refresh** (F5) forcing refresh of currently displayed content
- **toggle browser/flat view** (F6), flat view displays all together the objects contained in the current path or in the archive
- **always open archives in flat view** permanently sets flat view for all types of archives, does not affect file browsing

Organize menu allows to customise the file/archive manager's aspect, replicating most of the features of "..." Style button in Tool bar, and showing less common settings as:
- Dock the application to screen angles/sides (Ctrl+Alt+numeric pad arrows), F11 full screen, Ctrl+Shift+F11 immersive
- Swap tool bar and address bar to give visual prominence either to file navigation or archive manager's functions
- Change the style of the breadcrumb bar (or display it as plain text) to adapt to different host systems' standards - KDE/Windows, macOS, Gnome...
- Toggle smart sorting of file names on/off
- Use enlarged icons in the file browser: with "Enlarge" flag set 16px icons are enlarged to 48px (small details and list modes), 48px to 96px (large details and list modes, small icons mode), 96px to 192 px (large icons mode).
- Use row select in file / archive browser
- Use classic search or simple search (default), more similar to natural language, and set default search to recursive (search in subfolders) or not recursive ("search here")

Tools submenu contains:
- o **Password Manager** (Shift + F9) encrypted password manager
- o **Set password / keyfile** (F9) sets the default password (and optionally keyfile, if two factor authentication is desired) to be used in browsing, testing, extraction and archive creation
- o **Create random password / keyfile** (Ctrl+F9) allows to sample entropy from the system and from user's actions to generate a random keyfile; this utility can be also used to generate random passwords to be used in any other application/website/etc
- o **Verify hash of binaries** check SHA256 hash (which is cryptographically strong) of backend binaries invoked by PeaZip against known values for known versions of those binaries, in order to identify if any one of the binary files has been replaced.

Executable files, libraries (.dll, .so, etc), and sfx modules are verified for integrity in this test.

Please note end users are still able to freely replace backend binaries with custom builds best fitting their own needs, this tool will simply report what binaries have been modified (unless PeaZip binary is modified from sources and recompiled to recognize the new hash values).

In this way users with stricter security needs can know on demand if unknown binaries are in use in their copy of the app.

- It is also possible to harden the PeaZip at compile time defining fixed paths for binaries, configuration, and non-binary resources, setting constants HBINPATH, HCONFPATH, and HSHAREPATH.
  In this way it is possible to force the application to use resources from paths enforcing the desired set of security rules, so binaries cannot be modified unless said security rules are fullfilled.

Please note that the most comprehensive security measure to avoid hijacking of executables and libraries is running the application from a safe path which does not allow unprivileged users to modify app's files, either installing a PeaZip package (or extracting a PeaZip Portable package) in a path which matches the desired security level.

o **Comment** (Alt+M) shows an editor which read and allows to modify the archive-level comment in RAR (if WinRar is featured on the system), ZIP, and ZIPX files. The comment editor can read the comment of the (first) selected file, and can write the comment to multiple archives at once (number of selected archives is shown in window title bar) even of mixed types, and load / save comment to text file. Comments can contain UTF-8 characters and can be up to 64KB in size. Comments are not supported in multi-volume ZIP/ZIPX files (consistently with7-Zip and WinRar), and RAR comments cannot currently being read from (but can be written to) multi-volume RAR files or legacy RAR4 format.
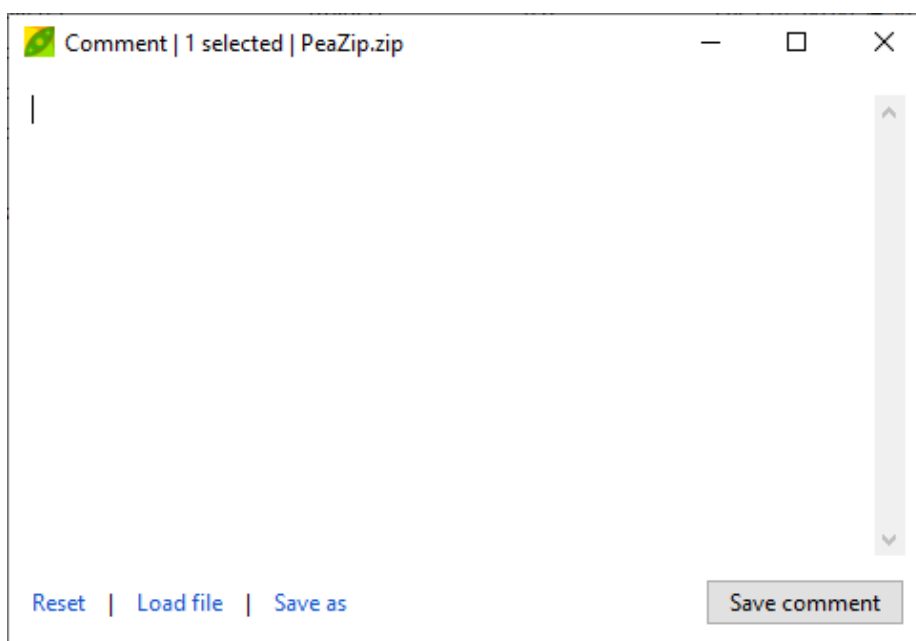
Image 8a: comment editor

o **Use advanced filters** (Shift+Ctrl+F9) sets multiple inclusion and exclusion filters to be used in browsing, testing, extraction and archive creation; filters are applied only to archive formats managed through 7z backend interface, see "Supported file types" chapter

o **Load Layout, Save Layout, Saved Layouts** allows to work with current Layout definition, which for file manager only contains Advanced filters.

o **Saved and scheduled task definition scripts** opens folder containing task definitions saved as script files / scheduled as system task, created from archiving or extraction interfaces, that can be run for unattended repetition of user-defined operations

o **System benchmark (pea)** is a synthetic performances test, single core and multi core.
  o Single core test runs a single process
  o Multi core test runs 16 processes in parallel
    - Ratio, for multi core benchmark, shows how faster was the calculation of the same integer math workload compared to single core
  o Details:
    - Benchmark runs multiple iterations of arithmetic and logic operations over an array of 64K 64 bit integers and 64K double precision floating.
    - Each process uses up to 6 threads: one CPU-intensive thread performs integer calculations, and up to 5 non intensive threads update the GUI.
    - Extra cores (to create and destroy processes, handle multiple GUIs' non intensive threads) and efficient system's scheduler positively impact multi core test results, while thermal throttling degrades general performances.

- Benchmark result unit, which is inversely proportional to completion time, is an arbitrary unit and does not represent directly any other measurement unit: benchmark results are purely meant to be self-consistent and allow comparison between different machines.
  - For reference a 2020 MacBook Air M1 chip score is 100/515 (100 for single core and 515 for multi core) when the benchmark is compiled for aarch64
- **System benchmark (7Z)** LZMA-based test to rate the host system in terms of MIPS (millions of integer instructions per second) and Core 2 Duo equivalent speed in MHz
- **PeaUtils**, standalone GUI for file management tools, if a reduced scope user interface is preferred.

"**Options**" submenu contains:
- **Run as administrator** (Windows Vista or more recent) closes current instance and opens a new one, prompting UAC elevation request, in order to work in protected paths.
- **Run as different user** (Windows only) closes current instance and opens a new one with alternative user profile.
- **Load / Save profile** store all configuration data (including bookmarks, custom compression settings, saved scripts, open with apps, themes, password manager… everything in conf path - except Profiles subfolder) to be loaded / saved in a single pass, either for different users or for different purposes, or to easily export all the conf data to another machine
  - Profiles are stored as *.profile.7z archives in "Profiles" subfolder in conf path and can be optionally encrypted with password/keyfile
  - Loading a saved profile will overwrite all of the current configuration data (including the Password Manager, if used) so it is recommended to keep a fresh backup of the conf data (with Save profile function)
  - Profiles created with older versions of the app will work, but configuration may miss newly introduced settings, so it is recommended to routinely update profiles to keep up with newer versions of the app
- **Localization** brings to Options > Settings to change applications language and **System context menu language** (W7+)
- **System integration**, to customize file associations, context menu and sendto menu entries (Windows only).
- **ZIP filenames encoding** allow to set a custom codepage to encode filenames in ZIP archives (please note this setting applies to .zip format only) from a list of known codepages; last menu element brings to Settings, Advanced tab which allows to set all text encoding related options
- **Settings**, to customize application's behaviour and theme
- **Set advanced archiving options**, to customize archiving / conversion specific options from archiving screen
- **Set advanced extraction options**, to customize extraction specific options from extraction screen

"**Help**" submenu opens settings & help system interface, pointing to check for updates, Plugin, Themes, and Translations in project's website, and providing access to most up to date documentation available online, and the offline localized PeaZip tutorial (F1), and help file (Alt+F1).

From this submenu it is also possible to visit PeaZip's Donations page, which links to funding pages of a few suggested charitable organizations. This allow to turn the perceived project's value in real help to whom it need it most.

Tool Bar

On the right side of application's toolbar the "…" **Style** button allows to toggle the toolbar between Archive Manager, File Manager, and Image manager, to display large, medium, or small version of tool bar, and to customize application's GUI.
It also serves as alternative way to show the Main menu - which can be optionally hidden - as a popup menu.
If the tool bar is hidden (size set to None or Compact), the Style button is automatically shown on the right side of the Address bar.
If the tool bar size is set to Compact, a small selection of the tool bar buttons is shown on the right side of the address bar.
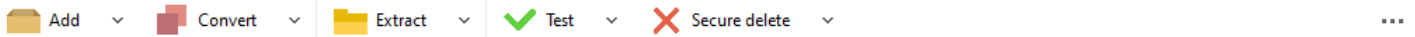

Image 9: Archive Manager tool bar.

The **Archive Manager** toolbar features:

- **Add**
    - While browsing the filesystem, the button adds selected files and folders to the current archive layout; before confirming the creation of the archive with "Ok" it is possible to modify the list of objects to be archived (dragging them or using the context menu) as well as other options, output name etc, see "Create archive" chapter for more information.
    - While browsing an existing (writeable) archive, the button brings to the archive update interface; it is possible to add files and folders to be added to the archive as in the previous case, dragging them or using context menu.
- On the right of add button a dropdown menu features most common compression tasks and most used formats, allowing to send items from the file browser to compression screen, loading in the meantime the chosen custom compression settings. "Immediate execution" option, if flagged, allows to start compression without requiring to click on "OK" button for confirmation, but as downside it does not allow further customization of the task.
- **Convert**
    - Convert selected archives in the desired format (non-archive files and folders can be added as well, the difference being in archives being extracted before compression stage). Using with "Add to separate archives" (default) it performs a mass conversion of listed archives, without that switch archive conversion is meant to consolidate input data in a single archive, improving compression efficiency because it allows recompression of the original data from its uncompressed form.
- On the right of convert button, a dropdown menu similar to archiving one allows to directly send items to conversion operation with most common pre-set parameters.
- **Extract**
    - While browsing the filesystem, extracts all selected archives; before conforming the extraction with "Ok" it is possible to modify output path and other options, and to add other archives to be extracted.
    - While browsing an archive, extracts the selected items or all the displayed items if nothing is selected, like WinRar/7-Zip (it is possible to be sure to extract the whole archive using context menu's Extract all entry, or clicking "Extract all" link in extraction interface); before confirming the extraction with "Ok" it is possible to modify output path and other options.
    - Right clicking Extract button shows a menu to extract all and (when browsing archives) displayed or selected objects, for partial extraction from archives.
- On the right of extract button, an arrow shows a menu with functions to directly extract all the content of the archive to most common destinations without further interaction with extraction confirmation interface
    - In the upper area are listed most common extraction operations: extract all here (to new folder, smart new folder), **extract all to** allowing to select output directory and proceed with extraction without leaving the file browser UI, extract all to desktop, extract all to documents, extract to 1..3° bookmarked paths (if defined), and extract to default extraction path. Only first, second and third bookmarks are shown as quick extraction destination to not clutter the menu, but bookmarks 1..8 are accessible with keyboard shortcuts Ctrl+1..8. Ctrl+0 extracts to the previous output path.
    - In the lower area it is possible to set most common extraction options: extract to new folder, naming policy (only for formats supported through 7z frontend), open output path after the task completes, set default extraction path…
- **Test**
    - Test selected archives for integrity
- On the right of test button, a submenu shows more informative functions: quick info or full listing of archive, system's file properties dialog, analysis of files and directory (showing recursively directory content and size), file checksum/hash, hexadecimal view of file content.
    In "File browser" submenu are shown functions displaying the result in the file manager windows, as "Find duplicate files".
    On the bottom of the menu are featured software configured for antivirus / antimalware scan (automatically detected or as user-defined in advanced applications list)

- **Secure delete**
  - o Erase selected files and folders (see File Tools chapter). Inside archives it becomes Delete from archive button. Please note that in both cases the files being securely deleted, or deleted from archive, cannot be recovered from the recycle bin.
- On the right, an arrow shows a menu with other functions replicating some ones featured in context menu, explained in details in context menu section of this chapter:
  - o File/archive manager functions
    - Copy to and Move to, create new folder, calculate file checksum, Quick delete, Zero delete, and Secure delete, zero or secure delete free space.
  - o Misc functions: Add to bookmarks, Web search (search selected item's name on the web), Explore path and Open command prompt here, to open the path being currently browsed with Explorer (or other default file manager) and command prompt respectively, Properties.
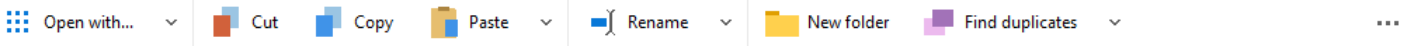

Image 10: File Manager tool bar.

The **File Manager** toolbar features 4 areas.

The first area contains Open with button, and dropdown menu showing custom applications (can be defined by the user with Settings > Applications). If nothing is selected, this dropdown becomes a launcher for the custom applications.

The second button group contains Copy/Paste buttons; Cut, Copy to, Move to, Copy path, and Select… (advanced selection dialog) are featured in the dropdown menu on the right.

Third group contains Rename button and a dropdown menu featuring a set of common renaming operations: convert the filename(s) in **lowercase** or **uppercase**, cut filenames at specified length, append or prepend directory name, number files accordingly to current sorting order, **add a string** (or **delete n characters**) at specified position, **replace or remove all occurrences of a string** (optionally case sensitive), replace or remove all non-AXCII characters, **change file extension**.

Rename functions are also featured in context menu (**File manager > Rename submenu**).

The last area contains various file management features, similar to ones featured in the last dropdown button of Archive manager toolbar; create new folder and find duplicate files (see below File tools chapter) are featured as buttons.

Image 11: Image Manager tool bar.

Image Manager toolbar features tree areas with quick access to functions related to manage selected graphic files (BMP, JPEG, PNG, and TIFF), operations are batch applied to all selected files of supported types, asking if modifying existing files or creating new ones.

The first area contains rotate left/right buttons and a dropdown menu with more rotation features (180°, flip vertically and mirror horizontally).

Crop button opens a dialog that allows to crop each side (top, bottom, left, right) of selected images of a given amount, expressed in percentage or in pixels.

Transform opens a dialog to resize and/or convert selected images; the dropdown menu on the right of Transform button provides quick access to most common image transformation operations: resize to a given percentage, fit to most common screen sizes, and convert to other image format.

Find duplicate files and Rename buttons, and file manager dropdown completes the toolbar providing additional useful features to manage collections of images.

Address Bar and Tab Bar

In the address bar are featured back and forward arrows to navigate in previously visited path (or any previously applied search filter), and an icon to go to upper level.

If the Tool bar is hidden, Style menu and a minimal group of buttons (add, convert, extract, test/list dropdown menu) become available on the right part of the Address bar.

The **navigation menu**, which replicates the structure of the navigation side bar (see below) in form of context menu, is featured in the dropdown menu at the left of the address field in **address bar**.

If user is browsing an archive, the nodes of current path inside the archive are shown at the top of the root of the menu for fast archive navigation (as well as they are shown on the top in the side bar).

The menu also contains entries for refresh (F5), set password, get information about currently browsed path, open tab bar, add bookmarks, and organize bookmarks - which displays the Bookmarks panel that allows to add, edit, sort and remove existing bookmarks.

Navigation menu is organized in **Bookmarks, Filesystem**, **History** and **Functions** submenus:

- **Bookmarks** menu reflects user-centric point of view, storing user defined favourite files, folders and search definitions. Bookmarks can also be managed from context menu, Navigation submenu**.**
- **Filesystem** menu is organized following a system-centric functional and hierarchical point of view, featuring links to commonly used paths like home, desktop, documents, mounted devices and more system defined paths
  If the system supports it, there are featured also searches' and libraries' folders, respectively introduced in Vista and Seven: files in those folders can be clicked to open the related virtual folder, from which it is possible to drag items to PeaZip.
- **History** menu is a chronological point of view, displaying the history of visited paths, archives, and search filters applied during the current session; below this first section are listed latest accessed archives (this feature can be disabled for privacy, see "Settings" chapter), and breadcrumb address paths.
- **Functions** menu shows most used functions, such as open path, open archives, search and drag to PeaZip, web search, open path in system file explorer, properties. It also features a submenu where most of program's functions can be found listed alphabetically, and a submenu to open selected items with user-defined custom applications.
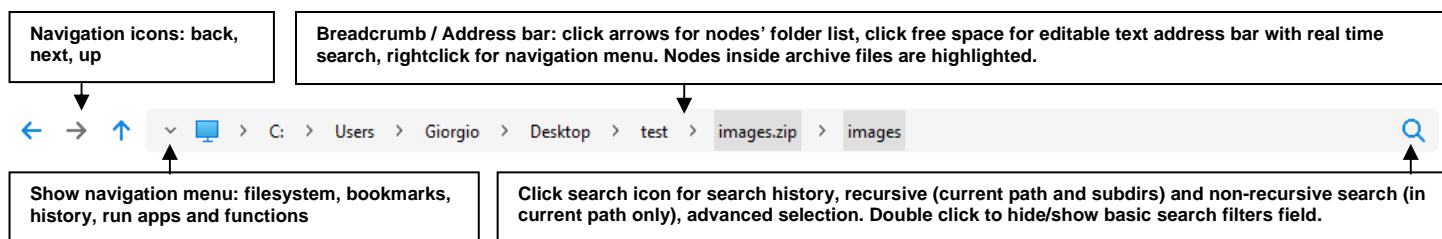

Image 12: address bar; back and next icons are greyed when non available (displaying last or first path in history)

A minimal session history menu is available right-clicking back and next buttons, and a minimal breadcrumb menu is available righ-clicking the up button.

The address field can be viewed as a breadcrumb or as an address field.

The **breadcrumb** (different styles are available from main menu, Organize) allows to click on parent folder levels in order to quickly navigate the file system (up to 8 levels represented) and inside archives (up to 4 levels represented, highlighted), the first path is always the root level (computer icon).
Breadcrumb number of items is automatically adjusted to fit address bar width.
Right-clicking breadcrumb items shows the aforementioned navigation menu.
Clicking on arrow spacer icons between breadcrumb nodes displays list of file system (or archive) directories at chosen level.

Clicking on the breadcrumb's free space transforms it in **address field**, which allows editing and cut/paste (with its own proper context menu for text field editing), and performs real time search while typing in the address filed.
Real time search function filters the content of the file browser (browsing, flat view, or search results) to match input, including wildcards.

Search (F3) field is on the right of filter's field, it can be used for basic recursive and non-recursive searches; * (string) and ? (single character) wildcards are supported.
Click on search icons to show the **search menu**: history of previously searched strings, Search… shows the complete search dialog, Select… shows the selection dialog (a fast alternative to search to identify objects matching desired filters), reset search filter, reset search history, launch search in current machine or on the web.
Double click on search icon, or click on "Show search bar" in search menu, to keep the search bar expanded, to be able to type for search directly in this rightmost portion of the address bar rather than opening the search dialog.

Please note that this basic search filter is overridden by advanced filters (Shift+F9), for archive types allowing inclusion/exclusion filters (i.e. notably 7z, rar, tar, zip…).
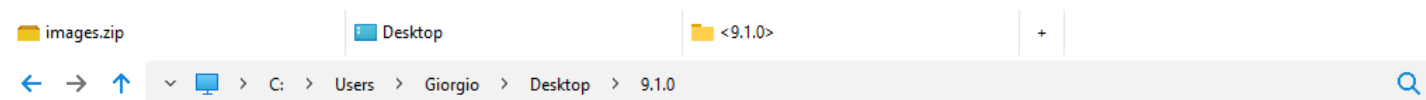

Image 13: tab bar, and address bar showing path (or search filter) of active tab. Active tab is shown with tab name between "<" ">" characters.

Tab bar, above address bar (if shown), enable tabbed browsing.

Tab bar is toggled on/off flagging "Tab bar" entry in Main menu > Organize, also featured in address bar's and tab bar's context menus. The bar is also activated clicking "Open in a new tab" in application's context menu.

New tabs can be created from "+" tab (on the right of the bar) or from context menu "New tab", cloning the selected tab.

Tabs can be closed from mouse middle click or tab's context menu, which also allows to close selected tab, all other tabs, or tabs on the right of selected one.

Closing the last remaining tab collapses the tab bar, which can be then re-opened from "-" entry in main menu.

Tabs can be arranged and resized; collapse / expand entries in tab's context menu set them to default small/large sizes.

Tab's context menu features "Bookmarks" submenu to add tab to bookmarks, and to organize (sort, delete) existing bookmarks.

"Save tabs" option in tab bar context menu allows to remember open tabs on exit: saved tabs will be re-opened at next application's start-up.

Navigation side Bar

To quickly jump to desired directory or archive PeaZip offers a **Side bar** on the left side of the file browser (see Image 7 on page 11, left area), showing either **Navigation** shortcuts (system's paths, user-defined bookmarks, recently visited archives, most common operations) or a classic **Treeview**, bar mode can be set (or hidden) from Style menu clicking "…" button on the top right of the tool bar.

While browsing an archive file, the Navigation bar will show in-archive directory tree structure before navigation shortcuts (bookmarks, system, recent, operations), that can be used to navigate the archive. A context menu item allows to expand / collapse the structure in a single click in order to immediately explore archive structure.

A smaller **Compact side bar** can be optionally made visible, or hidden, separately from the larger main Side bar.

This side bar only shows links to most commonly used filesystem paths, and it is mainly meant for use when the navigation bar is hidden or in treeview mode.

Status Bar

Information area: files and folders in current path or search result; stats of selected items and clipboard; info on current archive (if applicable). Click to display more comprehensive information including free space and filesystem type on current disk.	Set password (and optionally a keyfile) for browsing, extracting and creating encrypted archives

3 dir(s), 1 file(s), 38.8 MB   Selected 0 dir(s), 1 file(s), 38.8 MB   Clipboard: Cut 1 file(s)   🔓
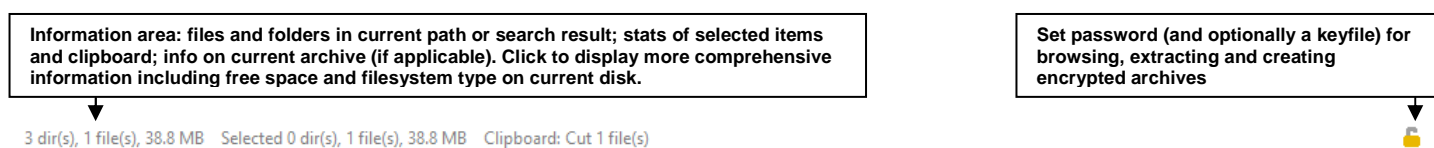
Image 14: status bar; can be right-clicked to display alternative views (bookmarks, Clipboard, etc…)

The padlock icon, on the right of the status bar, as well as "**Set password / keyfile**" menu entry (F9) allows to set password and keyfile to be used when operating on archives (browsing, extracting/testing or creating encrypted archives); this information is kept for the current session, and can be different for each separate instance of the application.

Encrypted content, while browsing an archive, is shown with a '*' character appended to filename - please note some archive types allows to mix encrypted (even with different passwords) and not encrypted content in the same archive.

If the archive is encrypted and the password is not provided, a popup will ask the user to enter the password/keyfile when extract/test/list operation are attempted.

If the password/keyfile is set, the padlock icon is changed to highlight that.

Keyfile is not mandatory, it can be used if two factor authentication is preferred to password-only authentication; keyfile creation utility can be launched from Tools menu or with Ctrl+F9.

If the directory structure area of the archive is encrypted (i.e. .7z archives created with –mhe option), browsing is not possible until the correct password/keyfile is provided: the archive browser will be empty and "no matches" will be displayed in the status bar until it becomes possible to browse the archive, having the user provided the right password.

Please note the same can happen when the archive cannot be browsed for other reasons, i.e. it was corrupted due to bad download or storage media failure. When strong encryption is involved, it may not be possible to determinate if the provided password is incorrect or if the file is corrupted, since resulting output will be random-looking in both cases.

If errors are detected when trying to browse the archive (either PeaZip being successful or not in recovering the archive's table of content), it will be notified in status bar: it is recommended to use test function for a detailed report of the problem.

When opening an archive, PeaZip attempts to browse the file, not only the root, so the user can be warned in advance of issues.

This pre-emptive check may take longer times while opening very large archives (containing hundreds of thousands of files), and can be disabled setting Options > Settings > Browser to "Fastest" in Performances group.

Archives with errors can in some cases being browsed and extracted without issues, if changes are minimal, but it cannot be granted since the nature of the data corruption (or willing, possibly malicious changes) cannot be identified a priori.
Browsing of archives containing errors can be disabled unchecking Options > Settings > Archive manager "Try to open archive containing errors"

Featured views in the status area can be set from "..." Style button on the right of the tool bar at the top of the application:
- **None**, hide views to maximize browser's area (can be set from Organize menu in main menu and context menu)
- **Status bar** (default) displaying synthetic information about the currently selected and displayed content; clicking on the information string shows more in depth information about the filesystem, the current archive (if applicable), displayed and selected files, and clipboard content. Right clicking the information string shows a context menu allowing to navigate the alternative status bar modes listed below.
- **Bookmarks** shows bookmarked files, folders and archives, bookmarks can be dragged to arrange them in the desired order, clicking on the leftmost column of the Bookmarks table; otherwise the bookmarks can be arranged alphabetically by bookmark's path or by description clicking on the respective column's header (clicking again will invert the order). It is possible to open bookmarked objects with PeaZip with doubleclick, but from bookmarks' context menu it is also possible to explore object's path with PeaZip or system's file manager, and to open command prompt in item's path.
- **History** collects recently visited archives, paths / search filters. In Options > Settings, General tab is possible to reset or disable history
- **Session history** contains paths / archives / searches visited in current session
- **Clipboard** allows to check objects currently cut and/or copied, and to remove single objects from clipboard if it's need to refine the selection. Clipboard behaviour can be switched between two modes:
  - **Standard clipboard** (default) behaves like usual file browser's clipboard, allowing a single cut or copy operation. Any further selection replaces the previous one, and on paste operation cut objects, have been moved, are removed from clipboard, while copied objects are kept in clipboard.
  - **Advanced clipboard** allows to store multiple (and mixed) cut and copy operations; any selection is added to the previous ones (if objects are duplicate, previously selected are kept), even from different paths and disks, and executed on paste operation, which clears the clipboard content.

Views can be closed, returning to default status bar, clicking on the grey title bar of the view.

Right-clicking on the status bar shows a context menu that allows to quickly switch between Views, and also provide quick access to Info, and Comment functions when archive files are selected or opened for browsing.

	Bookmarks	Full name	Name	Include	Exclude	Notes	Used	Last visited	Date added
🖥	Browsing	Computer's root	Computer's root	*			0		2023-02-18 12:04:57
🏠	Browsing	C:\Users\Giorgio\	Home	*			0		2023-02-18 12:04:57
🖼	Browsing	C:\Users\Giorgio\Desktop\	Desktop	*			2	2023-02-18 12:11:36	2023-02-18 12:04:57
↓	Browsing	C:\Users\Giorgio\Downloads\	Downloads	*			0		2023-02-18 12:04:57
☁	Browsing	C:\Users\Giorgio\OneDrive\	OneDrive	*			0		2023-02-18 12:04:57

Image 15: status bar showing Bookmarks view, which allows to add, remove, edit, and sort bookmarks. View can be changed from "..." Style menu in tool bar on the top right area of the application. Close views returning to the status bar clicking on the view title area (grey).

Drag and Drop

Drag and drop from system to application

PeaZip supports **drag and drop from system to application** on all supported operating systems.
Pressing Ctrl when dropping to the application will display a contextual menu for directly accessing the desired function, add to archive, open in PeaZip, copy or move here.
When files and folders are dragged to file manager, they get listed in the archive creation interface (as if they were selected and added with "Add" button), allowing to fine tune the task before confirming, or cancel the task.
If a single archive file is dragged to the archive browser PeaZip will open it for browsing.
Objects dragged into PeaZip while browsing an archive will be added to the current archive (if a single archive is dragged, a disambiguation message will allow to choose if to open it, or add it to current archive), this happens if the file type allows modifications - in example, it will not be possible adding objects to archive types supported only for reading, or to some solid archives.
It is generally not recommended to add objects to some container types that can be browsed and extracted by PeaZip, like i.e. .jar files, MS Office and Open Office documents etc, that are derived by other standard formats (i.e. zip, tar etc) but also features extra fields (i.e. digital signatures) that does not belong to the original standard: it is recommended to use format-specific tools for editing those file types.
PeaZip checks file extension to prevent improper modifications to those file types, so if it is necessary to bypass this security recommendation it is possible by
- changing the file extension in one with read+write support (i.e. .zip) to be able to edit the file with the application, or
- check "Edit non-canonical archive types" in main menu, Options, Settings, Archive manager.

Drag and drop from application to system or other applications

PeaZip supports **drag and drop from application to system** (and other application supporting drag&drop) on Windows, for copying/moving files and for extraction from archives.

PeaZip drag and drop is fine tuned for fast extraction from archives and is by default integrated with native Windows drag and drop mechanism, and optionally can switch to a purely custom drag and drop implementation (as implemented in early versions) from Options > Settings > General.

Advantages of PeaZip drag and drop mechanism for extraction from archives (either using native or custom option) are due to the data being written only once to system (trying to get the exact or nearest location to intended output) rather than being written to system temporary path and then moved to destination as in standard Windows drag and drop.
This brings consistent advantages in some areas:
- Improved **security**, as no write operation occurs outside the intended output path that may have different security policies than the system's temporary path.
- Improved **performances** if the target directory is on a different physical or logical unit than system's temporary path (quite common for network or removable units), as the data is written only once to disk – otherwise it would need to be written twice, first time in temp path, then in output path, which increases operation times and doubles disk usage/wear.
- Improved **stability**: unlike custom drag and drop, standard drag and drop operation will fail if not enough free space is featured in system's temp path.

It is possible to customize temporary work directory also from drag and drop extraction, allowing temporary data to be saved to output path (default, providing aforementioned advantages), user's temp (in this way PeaZip behaves like most other applications, so the advantage is having a single place to look at for deleting orphaned data), custom path (in example a ramdisk, so orphaned data will be safely deleted even in the event of a system crash), or none (disabling drag and drop extraction).
It is also possible to drag and drop files, or compressed items inside archives, from PeaZip to other applications i.e. 7-Zip, WinRar, etc.

Thumbnails

From ".." Style menu in tool bar, and from main menu > Organize, it is possible to set file browser to **show picture thumbnails** (by default off in details mode and on in icon mode), or to show generic picture file type icon.

Thumbnails are only available browsing the file system, not while browsing archives: no data is automatically trusted to be extracted from archives (even for temporary preview purpose) without the explicit intervention of the user.

Thumbnails can be toggled on/off in context menu > Navigation and in main menu Organize > Browser; in same menus are also available pre-sets with most common combinations of icon (or thumbnail) sizes and browser layout styles.

Thumbnails are generated in real time, PeaZip does not save thumbnails (neither as a temporary files), this is especially useful if PeaZip is used as portable application (or when privacy is important), as no track is let on the system and no extra space is taken. Thumbnails calculation generates a set of 4 dedicated threads, if folder or search filter is changed a new set of threads is created and old threads are stopped as soon as possible (file level and thread level granularity: each thread can be stopped separately, at the completion of each image).
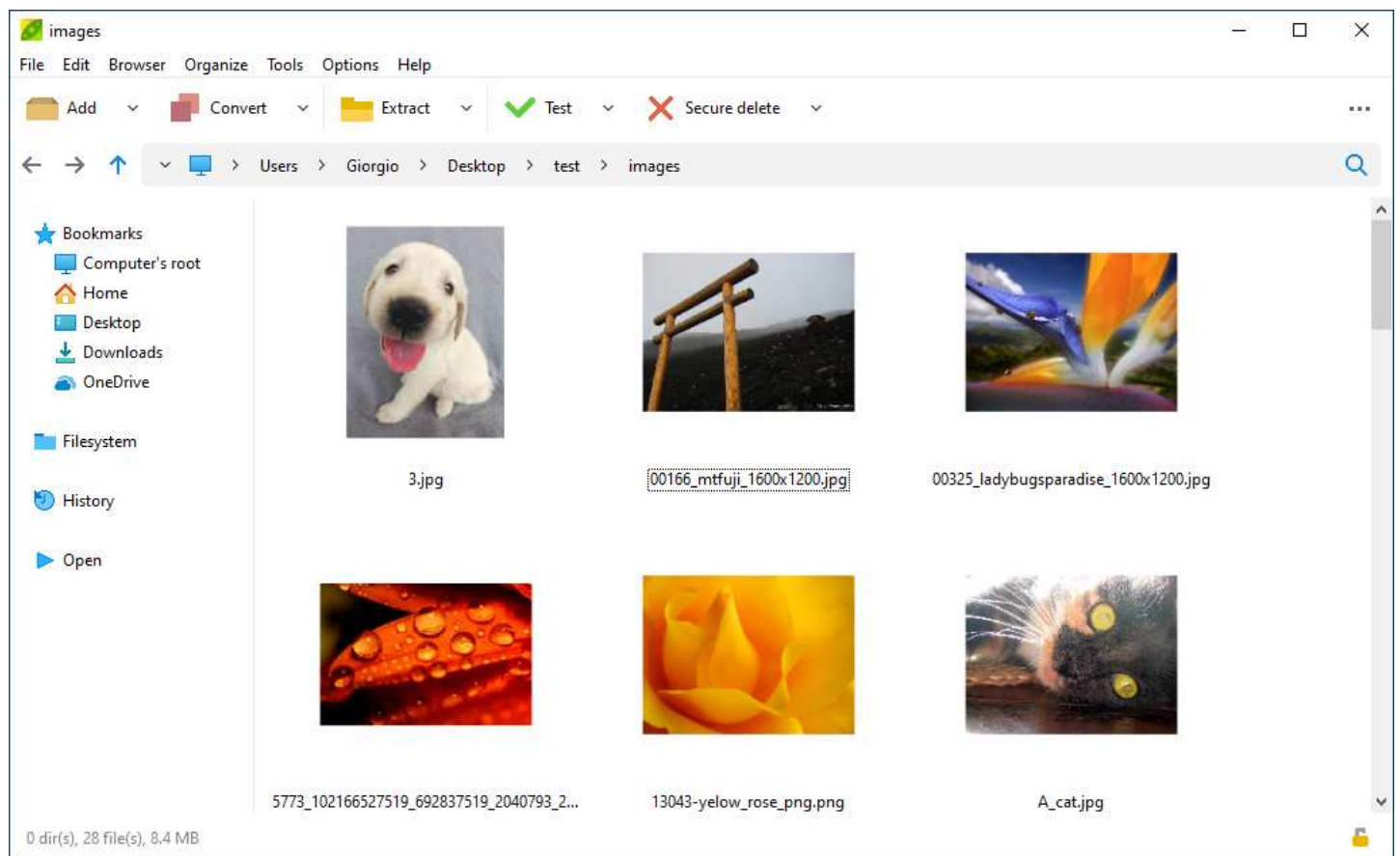
Image 16: browser showing large sized thumbnails of graphic file types, with Organize > Enlarge file browser icons option enabled.
Browser style can be set from "..." Style button on the right of the Tool bar, or Address bar if the Tool bar is hidden.
Thumbnails are shown only browsing the filesystem (off by default), not while browsing archive: images are calculated in real time with multiple threads and no image cache is ever saved to improve privacy.

File Browser

In the file browser, clicking on **titles bar** column's header sorts displayed objects by the selected column i.e. name, full (file and path) name, extension, date, size etc; a second click inverts the order.
Archives contained into another archive will be opened in a separate instance of PeaZip; please note that by default this is a "preview" operation, extracting data to a temporary folder (by default in the same path, if writeable).

*Hint: to open multi volume archives, the first volume (i.e. the one with .**001 extension**) must be selected.*

Info column, when browsing filesystem, shows '+' for known archive file types (open with double click), and '.' for known file types usually not handled as archive, i.e. MS Office files (open with context menu "Open as archive").
For graphic files (if the file browser is set to display thumbnails), Info column reports about the image: width, height, colour dept.
When browsing archive files:

- "Packed file size" column is featured instead of "Info", however only some archive formats, i.e. ZIP, features information about packed size for single items.
- "Created" and "Accessed" timestamps columns, and "Method" column are shown for 7z/p7zip supported file types –those columns can be hidden (as other ones) from columns header context menu or from Organize > Browser > Columns.
- "Checksum" column will show CRC or hash for compressed files (the function depends on the archive format and options employed at archive creation), and – if feasible – it will show content of directories: number of folders, files, and total uncompressed size.

*Hint: unsupported file types can be forcedly open as archives in PeaZip as **custom format**: this allows to use an arbitrary binary, and to customize the command's syntax, for dealing with them.*
*Otherwise, context menu **Open as archive** forces trying to open unknown file formats through the 7z/p7zip backend, so it is useful for opening a file thought to be a variant of another format as zip, tar, cab etc, rather than switching to the "Custom" interface.*

The browser can switch between **flat view** mode, displaying all objects contained in the archive, and classic browser mode, using "**Toggles browse/flat view**" (F6) in main and context menus – which applies both when browsing filesystem and archives.
"**Always open archives in flat view**" switch (in main and context menu) keeps applying flat view, but only when browsing archives.
Flat view is used also when performing basic search or applying advanced filters; opening a folder will browse it in classic mode, exiting the flat view mode.

Manually switching to flat mode (F6), or using search or filter feature, will allow displaying all the content of those archives.
Extraction, listing and testing of the archive is not affected by this issue.
*ACE, ARC, *PAQ, PEA and QUAD/BALZ/BCM archives can be browsed by PeaZip in flat mode only.*

While browsing 7z/p7zip supported archive types, "Rename" and "Move to" options are available, through in-archive rename function - "Rename" changes object name keeping path constant, "Move to" keeps name constant and allows to change path.
Those functions allow to change structure of existing archive, without having to rebuild the entire archive.

Context Menu

The browser's **context menu**, activated right clicking on the archive browser area, is organized in 4 areas, and is context sensitive as provides different options while browsing the filesystem or archives of various types (which supports different operations).

Archive management

The top area features **archive management**-related entries: **Add**, **Add to separate archives**, **Convert**, **Extract**, **Test**, **Info** (synthetic information about the archive), **List** (applies also to uncompressed files/folders), and **List (with details)** for a in depth report of archive content.
While browsing the filesystem **Add** will send selected files and folders to the archive layout, while browsing an archive it opens a submenu allowing to add files or folders or to open a search dialog from which files/folders can dragged into the archive itself.
In this case objects will be archived starting from archive's root, compressed and encrypted accordingly to the archive's settings.

Please note that if it is possible for the archive format to store objects with different passwords into the same archive (i.e. in .7z format), you can set the different passwords each time you add objects.

*Note **adding files/folders to encrypted .7z archives**: .7z archive format can store objects encrypted with different passwords in the same archive, so when adding an object to an encrypted 7z archive the object will be encrypted with the password/keyfile currently set (in the form which popup clicking on the padlock icon).*
If the .7z archive is encrypted with "Content and filenames" option, objects can be added only using the same password / keyfile for the entire archive.

By default, if same object exists in the archive (same name and same path in archive's directory three), the object will be updated: if the object the user is adding is newer, it will replace the older object; if it's older, the archive will not be updated.
Please note that for some archive types, notably most solid archives (i.e. .7z created with solid archive switch enabled), or archive types supported only for browsing extraction (i.e. CAB, RAR…), doesn't allow neither add/update nor delete operations.
Convert entry allows to convert files/folders or existing archive(s), even of multiple mixed types, in the desired output format, either converting single archives or consolidating input in a single output archive (recompressing the original content)

Extract and test/info functions in "**More**" submenu are shown disabled when no archive is selected.
While browsing archives, extraction and test/info functions are transformed in submenus allowing applying the action to the entire archive, or only to displayed or selected objects.
More submenu also contains menu items to **save edited files to archive** on demand, or to discard temp edited files so editing can start from scratch from last version of files saved to archive.

Image 17: Application's context menu

Editing files in archive, when a file is saved (to temp work path) PeaZip will ask to save edited files into the archive; options are:
- **Yes/Clear** save edited files to archive (may require time for large files, or solid archives) and delete temp work files; further edits will require files – which are now synced with edits - to be extracted again
- **No/Clear** do not save edited files to archive and delete temp work files: basically discard all changes and start from scratch (last save)
- **Yes** save edited files to archive (may require time for large files, or solid archives) and keep temp work files for further editing without needing to re-extract them
- **No** do not save edited files to archive but keep temp work files for further editing, edits can then be saved later i.e. from context menu > More group.

ZIP filenames encoding allows to set a custom codepage to encode filenames in ZIP archives (please note this setting applies to .zip format only) choosing it from a list of known codepages
The last menu element **Text encoding** points to Options > Settings > Advanced, Text encoding options, which contains all text encoding related options, as freely editable custom codepage for ZIP filenames, change encoding for console commands, etc.
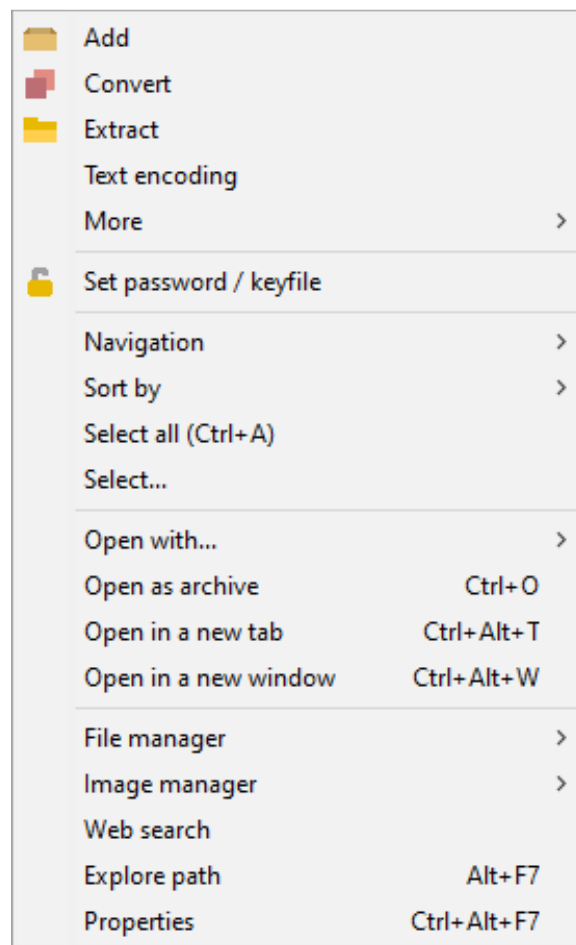
On the bottom of the archive management area is displayed "Set password / keyfile": clicking it will display password form in order to set password to open archives, or for creation of new encrypted archives.

Direct extraction

Direct extraction (extract here, to new folder, smart new folder) functions, and **Extract to…** submenu, are visible only when browsing an archive, or if at least one archive file is selected in the file browser.
These entries allows to directly extract all or selected items without further confirmation.
Extract to… submenu displays bookmarked paths and history (previously visited paths, session history, and breadcrumb paths) allowing to **directly extract all or selected items to any path saved in bookmarks or history with a single click**.

Browsing / Navigation

The second area contains features related to navigate, and to manage the file browser.
"**Navigation**" submenu allows to add / manage Bookmarks, open archives and paths, toggle browse / flat view (show all), set flat view for archives, and save list of current files/dirs (including size, date, attributes, checksum if present, etc) to TXT file.
It also provides an alternative access to Settings and to enable/disable main menu, Tool Barr, Side bar, Status Bar etc…
This area also contains "**Sort by**" submenu, that allows setting / inverting sorting column (column's header is not available when the browser is in List or Icons mode), "**Select all**" (Ctrl+A) and "Select…" (**advanced selection dialog**).

Open with / Preview

In the third area **Open as archive** forces opening the selected file as an archive type, if it is possible to find a suitable backend to handle its structure.
Open in a new tab shows tab bar (if hidden) and browses / opens as archive the selected item in a new tab, and **Open in a new window** browses or opens the item in a new instance of PeaZip.
In "**Open with…**" submenu it is possible to run selected files and folders (in filesystem as well inside archives) with preferred programs, scan with antivirus/antimalware etc; applications can be set in Options > Settings > Open with… which is reachable also from this menu.

Hint: in Settings > Applications it is possible to define a sets of up to 16 simple and 8 advanced (more flexible syntax) custom application or scripts (i.e. editors, players, antivirus/antimalware etc) to be used to open files and folders bypassing the default system's file associations; by default PeaZip tries to find some of most common applications.
Security related applications like virus scanners are conventionally placed in the second area (advanced) when automatically detected by PeaZip.

While browsing an archive, "**Open with…**" submenu is replaced by "**Preview with**…" submenu; preview functions will extract selected objects to a temporary folder, and then take the requested action on the resulting output.

Hint: if the archive is in a read only path, preview functions will transparently switch to user's temporary folder, while extract here functions will warn the user and ask to select a writeable output path.

Double-clicking an object (while browsing the filesystem) triggers "open with associated application" action, while browsing an archive double-click triggers "preview with associated application" action.
If previewed file is modified, PeaZip will automatically ask to add it back to the archive replacing the older version of it - in case the archive type is supported for writing.
From context menu, More, "Force update modified files in archive" it is possible to manually try to save edits even if the archive extension is not known as supported type.
From main menu, Options > Settings > Archive manager, Extraction group, it is possible to enable "**Extract and open with**…" submenu in this area, actions in this submenu, unlike ones in "Preview", will not extract content to a temporary path - so should be used with caution.

File Management

The fourth area contains **file management** features

"**File manager**" submenu allows to create new folder, rename (F2 or Ctrl+R), **advanced rename** set of functions (lowercase, uppercase, number files, append/prepend directory name, cut filename at n characters, add, delete, or replace string in file name, replace/remove non ASCII characters, change file extension), move to Recycle Bin, quick delete, zero delete or secure delete, zero or secure delete free disk space, copy or move file, open command prompt in the selected path.
Cut, copy and paste operations can also be performed with keyboard shortcuts of Ctrl+X, Ctrl+C and Ctrl+V respectively.

Pressing Ctrl+V while browsing an archive adds objects from clipboard to the archive; in this case objects will not be removed from the filesystem even if Ctrl+X was used, since PeaZip tries to not automatize potentially dangerous operations such file deletion, letting that kind of decision under the full control of the user.

Quick delete performs the fast deletion of selected objects, without needing to move them to recycle bin, **zero delete** and **secure delete** (only in the filesystem, not available in archives) performs a more secure file deletion as described in "File tools" chapter.

While browsing archives filesystem related functions (including cut, copy, paste) are disabled and the area shows only simple deletion from archive, if this action is supported for the current archive type.

File manager submenu also contains two submenus:

"**System tools**" collects system's disk utilities (clean, defrag, manage, remove), system management tools (control panel, computer management, task manager) and display environment variables (both for Linux and Windows).

"**File Tools**" contains features to find duplicates, calculate checksums and hashes and more, see File Tools section in this chapter for more information.

"**Image manager**" allows to rotate, crop or resize graphic file types (JPEG, PNG, BMP, TIFF etc).

The area also contains **Web search**, **Explore path** to open system's file manager in the selected path, to quickly switching working with PeaZip and standard system's file manager, and **Properties**, starting a standard system's dialog for displaying properties of selected files/folders.

Other Context Menus

Clicking on **Style button** on the top right corner of the application (in the tool bar if visible, otherwise in the address bar) shows Style menu allowing to quickly set look and feel of the app: visible bars, treeview and file browser mode.
Rightclicking on the same button shows the main menu as a popup menu, which allows to hide the main menu bar without losing in functionalities.

Clicking or right-clicking on the **Navigation button** on the left of the breadcrumb bar (in the address bar) shows the navigation button, which provides quick access to bookmarks, tab bar and navigation functions.

Rightclicking **columns header**, while file browser is in details mode, shows **Browser header** context menu allowing to
- set visible columns
- adjust columns to default sizes
- save the list of displayed files as text file (tab-separated TXT including all columns of the file browser)
- copy to clipboard current path, or paths of selected items
- perform on the fly **analyse content of folders**, **search of duplicate files** or **checksum/hash calculation** without leaving the file browser UI (result is displayed in Checksum/hash column, on the right) - the algorithm for both functions can be set from Algorithm submenu - see File Tools chapter for more information. *

* Please note those functions are only available when browsing the filesystem - checksum/hash values of files inside archives are pre-computed during the archive creation (usually CRC32 is the standard, optionally RAR5 can use BLAKE2 hash).
Computing a different checksum/hash for each single input file (if simply checksum/hash verification of the whole archive is not fit for the purpose) needs to be done before or after archiving - in example, calculate any desired hash for input files, save the report as TXT and add the file to the archive.

Alternative way to access to columns' header context menu is available from "Open Browser header menu" in the Navigation menu, which is accessible from the aforementioned Navigation button.
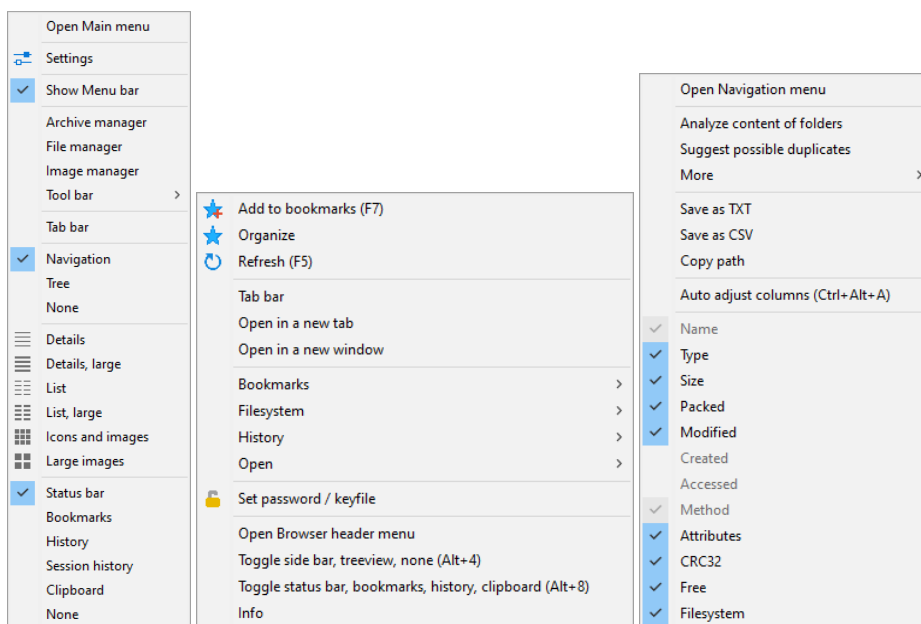

Image 18: from left to right Style, Navigation, and Browser header context menus

File manager's keyboard shortcuts and mouse controls

File/archive browser supports following keyboard shortcuts; some functions are format-specific and will be ignored if not supported for the current archive type.

Functional keys:

F1	brief translated help instructions / Alt+F1 help file (pdf)
F2	rename file(s) / Ctrl+F2 copy selected to / Shift+F2 move selected to
F3	search (recursivity option is remembered) / Ctrl+F3 start as non recursive (search here) / Shift+F3 recursive / Alt+F3 system search
F4	show navigation menu / Ctrl+Shift+F4 browse root / Ctrl+F4 browse desktop / Shift+F4 browse home
F5	refresh / Shift+F5 add selected to archive / Ctrl+F5 extract selected / Alt+F5 extract archive (opens selection dialog)
F6	toggle browse/flat view
F7	add to bookmarks / Ctrl+F7 open in a new instance of PeaZip / Shift+F7 open command prompt in selected path / Alt+F7 explore selected path
F8	browse first item in bookmarks list (Ctrl, second, Shift, third)
F9	set password/keyfile / Shift+F9 password manager / Ctrl+F9 create keyfile or random password / Ctrl+Shift+F9 set advanced filters
F10	menu / Ctrl+F10 run as different user / Alt+F10 run as administrator (Windows Vista and newer systems)
F11	toggle full screen mode / Ctrl+Shift+F11 immersive (full screen, no window border, remove with F11 or Ctrl+Shift+F11)
F12	extract all to...

Basic operations

Open as archive	**Ctrl+O**
Extract/ Extract selected	**Ctrl+E**
Quit / close window	Ctrl+W
Read and modify archive-level comment	Alt+M

Extract without confirmation (from file browser):

Extract all to… (open selection dialog)	Ctrl+Alt+E
Extract all here	Ctrl+Alt+Shift+E
Extract all here (smart)	Ctrl+Alt+Shift+S
Extract all here (to new folder)	Ctrl+Alt+Shift+N
Extract all to Desktop	Ctrl+Alt+D
Extract all to Documents	Ctrl+Alt+Shift+D
Extract to previous output path (if set)	Ctrl+0
Extract to path of 1..8' favourite (if set)	Ctrl+1..8
Extract to default path (if set)	Ctrl+9
Toggle extract to new folder on/off	Ctrl+N

While browsing an archive:

Extract displayed content	Ctrl+S
Extract all content	Ctrl+L
Test all (when browsing an archive)	Ctrl+T
Extract and open with PeaZip	Ctrl+Shift+Z
Extract and open with default application	Ctrl+Shift+O
Extract and open with …	Ctrl+Shift+W
Preview functions	same as previous ones, using Ctrl+Alt+Z/O/W
Preview selected	Enter or doubleclick
Properties	Alt+Enter

File Tools (when browsing the filesystem):

Compare selected object with…	=
Checksum and hash of selected objects	?

Navigation:

Select all	**Ctrl+A**
Toggle browse mode / flat view mode	*** or F6**
Display/organize Bookmarks	Ctrl+D
Display History	Ctrl+H
Display Session history	Ctrl+Alt+H
Go to Computer's or archive's root	Ctrl+Shift+F4

Go to Desktop	Ctrl+F4
Go to Home	Shift+F4
Go to Downloads	Ctrl+Alt+F4
Go to Documents	Ctrl+Alt+Shift+F4
Search	F3 / Ctrl+F3 non recursive / Shift+F3 recursive / Alt+F3 system search
Add to bookmarks	F7
Browse first item in bookmarks list	F8
Open directory/archive	< or Enter or doubleclick on the folder/archive
Up one level	> or Backspace or Alt+Up arrow
Go to object's path	Ctrl+P (useful in flat view and search/filter mode)
Go back in history	Ctrl+B or Alt+Left arrow
Forward in history	Ctrl+F or Alt+Right arrow
Open in a new tab	Ctrl+Alt+T

Bars

Highlight menu bar	Alt
Swap tool bar and address bar	Alt+0
Hide/show menu bar	Alt+1
Toggle Tool bar	Alt+2
Toggle navigation side bar	Alt+4
Toggle sort column	Alt+5 (*)
Toggle browser's pre-set appearance (styles, sizes)	Alt+6
Toggle status bar view	Alt+8 (**)

Sort Column (*)

Name	Ctrl+Alt+0
Type	Ctrl+Alt+1
Size	Ctrl+Alt+2
Packed Size	Ctrl+Alt+3
Modified date/time	Ctrl+Alt+4
Created date/time	Ctrl+Alt+5
Accessed date/time	Ctrl+Alt+6
Method (compression)	Ctrl+Alt+7
Attributes	Ctrl+Alt+8
Checksum or hash	Ctrl+Alt+9
Auto adjust column sizes	Ctrl+Alt+A

Views (**)

None, hide "Views" panel	Shift+Alt+0
Status bar: basic information, navigation and password icons	Shift+Alt+1
Details, replicate and extend status bar's functions	Shift+Alt+2
Bookmarks, organize favourite files, folders and search filters	Shift+Alt+3
History, latest folders and archives	Shift+Alt+4
Clipboard, objects currently scheduled for copy or move operations	Shift+Alt+5

Modify:

Recycle bin / Delete form archive	Del
Quick delete	Shift+Del
Zero delete	Alt+Del
Secure delete	Ctrl+Del
Refresh	F5 or icon in first column of titles' bar
Create new folder	Shift+Ctrl+N
Rename	Ctrl+R or F2
Cut	Ctrl+X
Copy	Ctrl+C
Paste	Ctrl+V
Cancel current selection and clear clipboard	Esc

PeaZip for Darwin / macOS replaces some of the keyboard shortcuts with other ones meant to be more natural for MacOS users:

Move to Trash	Command+BackSpace
Search	Command+F
Select all	Command+A
Copy	Command+C
Cut	Command+X
Paste	Command+V
Create new folder	Shift+Command+N
Up one level	Shift+Command+Up arrow
Previous item in history	Shift+Command+Back arrow
Next item in history	Shift+Command+Forward arrow
Go to Computer's or archive root	Shift+Command+C
Go to Desktop	Shift+Command+D
Go to Home	Shift+Command+H
Go to Downloads	Shift+Command+L
Go to Documents	Shift+Command+O

File manager's mouse controls

Doubleclick: preview selected object with associated application
Rightclick: activate file/archive browser's context menu
Middle mouse button: navigate up one level
Extra mouse button 1: go back
Extra mouse button 2: go forward
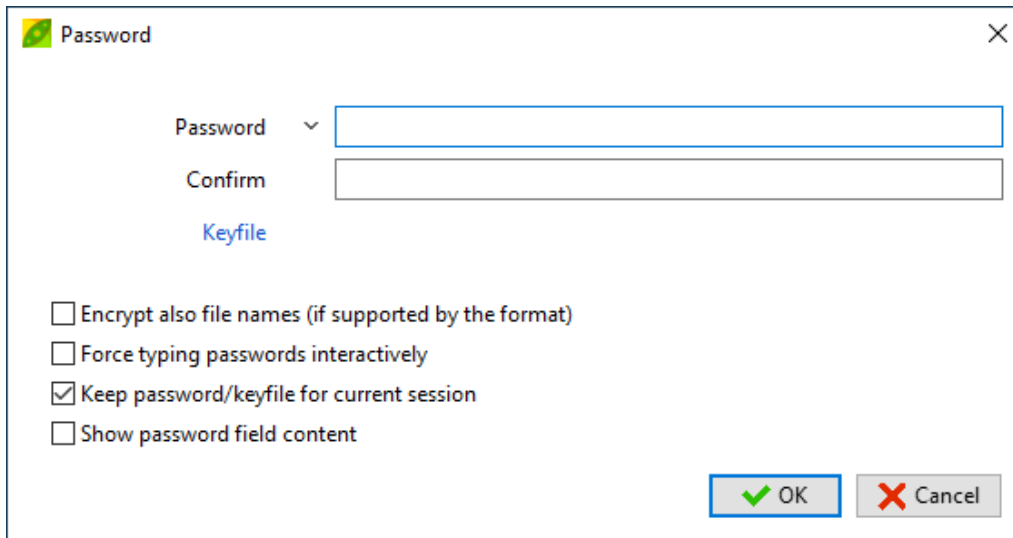
Set password (F9)

Image 19: Set password (and optional keyfile) prompt

Password form allows to set password and, optionally, a keyfile for two factor authentication.
Once a password is entered, it is used by:
- file manager, since some encrypted archives needs password for being browsed or tested
- archive extraction, to open encrypted archives
- archive creation, to create encrypted archive (if the selected format supports encryption); in archive creation interface it is specified if the password is set and if encryption is supported for selected archive format.

Two factor authentication is not applied to sfx archives; as self extracting modules are able to check for password only the keyfile field is ignored when compressing to self executable archives in order to not break compatibility.

If a keyfile is set for any other format than PEA (which has its own way to use keyfile) the SHA256 hash of the file encoded in Base64 (RFC 4648) will be prepended to the password. This convention allows any other file archiver to work on two-factor authentication archives built with PeaZip, even if keyfile parsing is not supported (or has different scheme) simply passing the Base64-encoded SHA256 hash of the keyfile as the first part of the password.

"Encrypt also filenames" option is used during creation of 7Z and ARC formats (PEA format always encrypt filenames regardless this option): if checked, the encrypted archive will need password for being browsed, else the content will be visible; in both cases extraction will require the password.
The padlock icon in file browser's status bar (also shown in archiving and extraction interfaces) will change to visually inform that a password is set.

On the bottom of password form, a bar will express colour code evaluation of password strength, red means the password is inadequate, yellow hints a weak password, green means the password strength is ok, dark green stands for strong passwords.
Please note however that a good password must be in first place not easy to connect with the user through social engineering, that is a far more efficient attack than brute force or dictionary based attacks, and in second place, it is recommended to avoid dictionary attacks not using common words or quotes as password.

On the left of Password field there is a drop down menu showing first 8 passwords saved in the **Password manager**, and a link to access to the Password manager (also featured in main menu, Tools > Password manager) where all the passwords can (optionally) be stored.
The list of password is empty if no password is saved, or if the password manager is password protected and user has not yet authenticated (see next chapter).

Password options

In the bottom area of the form are listed password options.

"**Encrypt also filenames**" option increases security for file formats (7Z, ARC) supporting this feature. If filenames are not encrypted, an unauthorized attacker may open the archive and get an idea of the content, even not being able to decrypt and extract it, which may be unwanted in many scenarios.

"**Force typing password interactively**" if this feature is supported by the backend for the selected format, PeaZip will require to type each time the password directly in the instance of the backend binary, running the in console mode.

This option allows to create scripts that will not run unattended but will rather ask user for interactively providing the password.

Please note this option will not allow to browse archives created with "Encrypt also filenames" option, and will set backend binaries to run in native console mode - backends not supporting encryption are not affected by this option.

Notes:

7z / p7zip, FreeArc, and Rar backends support this option, running in console mode.

Pea backend supports this option even running in GUI mode.

Zpaq backend supports encryption but does not support this option.

Other backend does not support encryption so information in this chapter does not apply to them..

With "**Keep password for current session**" flag (default), password and keyfile are kept until changed or until PeaZip is closed; multiple instances of PeaZip can keep different passwords. Otherwise, password and keyfile are automatically reset each time a new archive is browsed.

"**Show password field**" content makes the password field visible, making easier to set password and not requiring retyping it for verification. However, this way the password is also easier to read for unauthorized bystanders.

Password manager (Shift+F9)

Password manager is accessible from main menu, Tools > Password manager or from Password form, from drop down menu on the left of Password field.

To **add** or **edit** password and notes, and for other options, rightclick on the password list area, or click the menu icon on the right of the tab bar, this will display the password list's context menu

Password list, contains passwords and notes (but can be used to store any kind of information as string, i.e. a keyfile name if desired), that can be copied and pasted to any application.

First 8 saved passwords will also be displayed in the dropdown menu in Password form: items in the password list can be dragged (from first column) to be **sorted** so most used password are displayed directly in the dropdown menu.

You can rightclick on the password list to open the options menu (add and remove items, reset, export…), doubleclick to edit an element, and use Ctrl+C to **copy** a string.
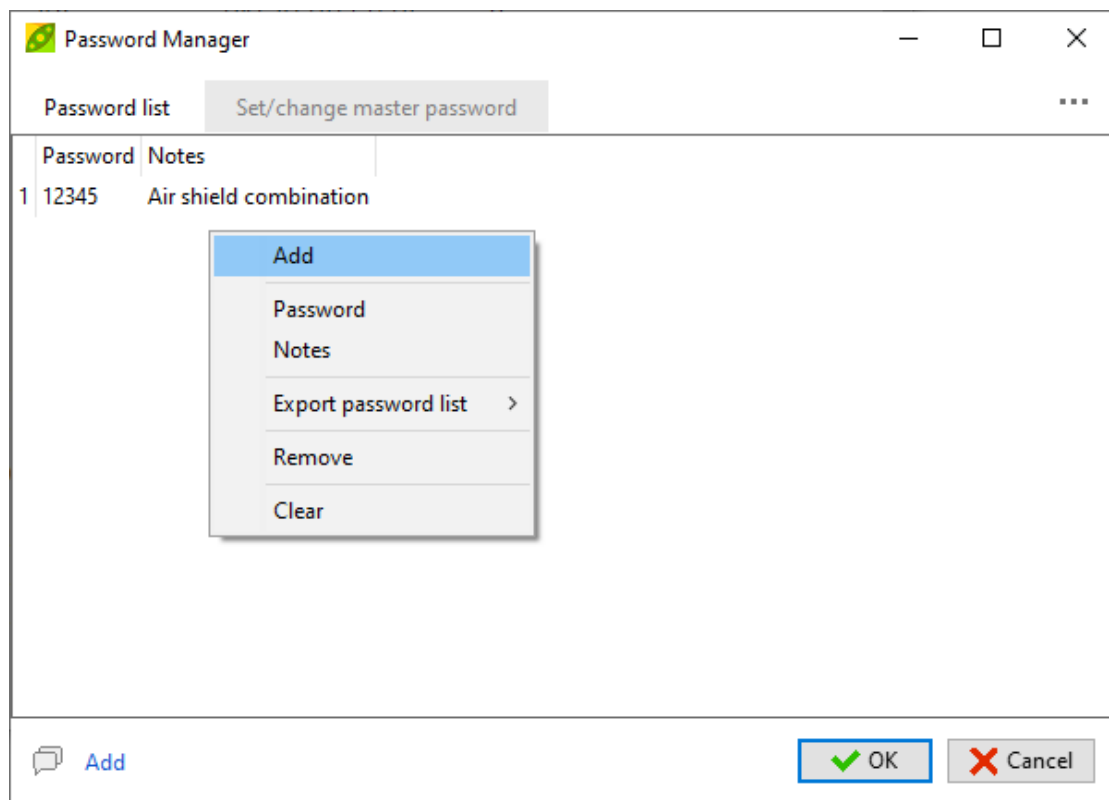
Image 20: Password manager's Password list tab, showing context menu

Set/change master password

If no password is set the password manager will be accessible without prompting for authentication, otherwise password list tab (and passwords in dropdown menu in Password form) will not be displayed until the user correctly authenticates, accessing to the password manager providing correct master password/keyfile.

The Password manager is protected by authenticated encryption (AES256 in EAX mode), and saved as encrypted file "pm" in PeaZip's configuration folder, if the file is corrupted or tampered the user will be asked if trying to recover it or to delete it.

Passwords can be exported as plain text (not recommended) or in its encrypted form, in this way a password list's .bak file can be manually imported as password manager file, renaming it as pm (no extension) and copying it to PeaZip's configuration path (linked in Options > Settings): this file will have the same content and password it had when it was originally saved.

This mechanism allows in example to roll back to a previous password list, or maintain various off line password lists, or exchange protected password lists with other users or other machines

Create keyfile (Ctrl+F9)

For higher security against dictionary and some social engineering attacks, a **keyfile** can be used along with the passphrase to key the encryption.
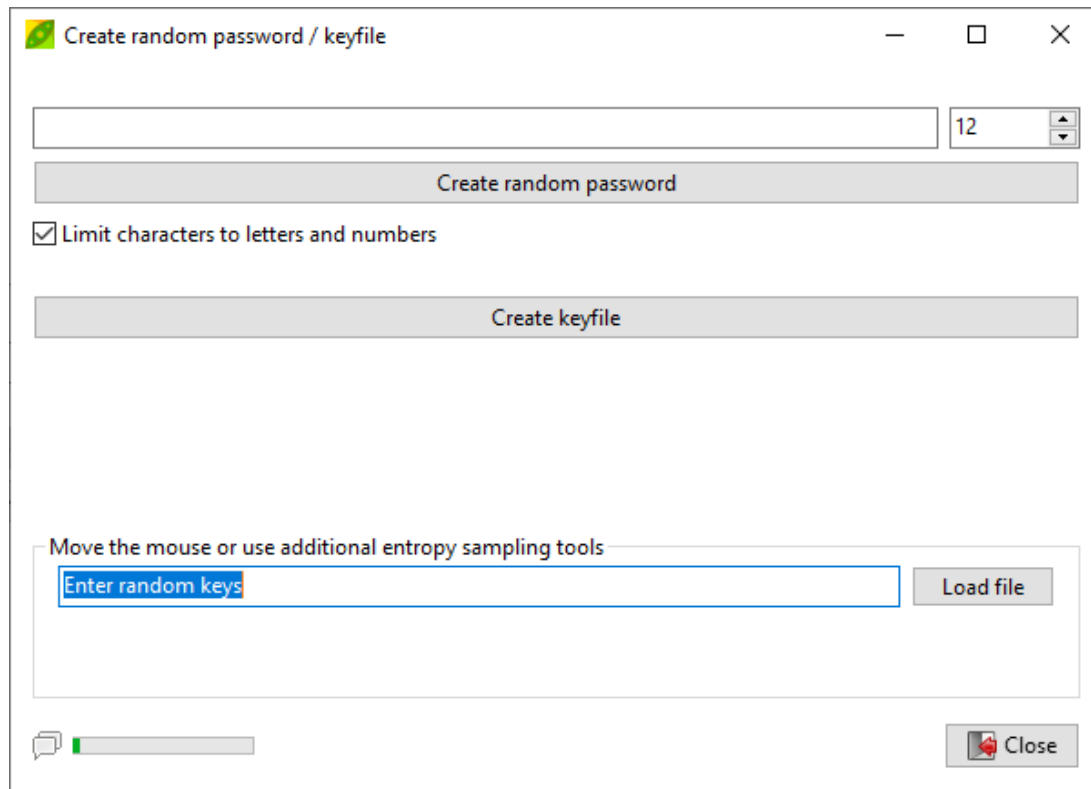
Image 21: keyfile generation utility

The keyfile need to be securely managed since its content need to remain secret as well as the passphrase; any file can be used as a key, but it's strongly recommended to use a randomly generated file.

If a keyfile is used for a non-PEA archive, the SHA256 hash of the file (no size limit) encoded in Base64 (RFC 4648) will be prepended to the password, then it will be possible to work on archives encrypted with a keyfile using PeaZip or any application following the same convention, or simply entering the Base64-encoded hash as the first part of the password.

PeaZip can **create a random keyfile** sampling different entropy sources and submitting entropy collected to a robust random number generation routine, in the same interface it's also possible to generate a 4-64 character **random password**; the password will contain mixed case base characters and digits only, in order to be typeable on any keyboard layout and to be accepted by almost all applications or online password forms.

The passwords and keyfiles generated in that way can be used not only in PeaZip but also in any other application requiring a strong password or a random keyfile.

This utility uses functions provided by pea's libraries so refer to Pea documentation for any detail about random number generation and entropy collection in PeaZip project.

Set advanced filters (Ctrl+Shift+F9)

"Advanced filters" form allows to use multiple inclusion and exclusion criteria, one per line (optionally recursive with "Recurse subdirs" options), and can be applied to archive formats supported through 7z and FreeArc backends.
The filters can be applied to archive's navigation, extraction and creation.
Exclusion filters prevail on inclusion filters.
Advanced filters bypass the archive browser's basic search filter (in mentioned archive types) so address bar is disabled, while browsing archives which supports advanced filters, if "Use advanced filters" option is checked.

In archive creation (Add or Convert) screen are available "**Include only**" filed, to apply inclusion filter to input directories (files individually added are always archived), and "**Include also**" to freely specify extra items/filters to be added to archive – i.e. enter a string containing an extra input (absolute) path and *.extension, to add all files matching chosen extension from the specified input path.

Multiple filters, one per line, can be written in the inclusion and exclusion fields; string delimiters (" on Windows and ' on Linux and other *x systems) are not needed to be explicitly entered by the user. In example, if the user needs to extract (or display) only "myfile.txt" plus all files named "your file" and all .mp3 files, but not .mp3 starting with "a" and "m", could write in the inclusion field:

      myfile.txt
      your file.*
      *.mp3
and in the exclusion field:
      a*.mp3
      m*.mp3
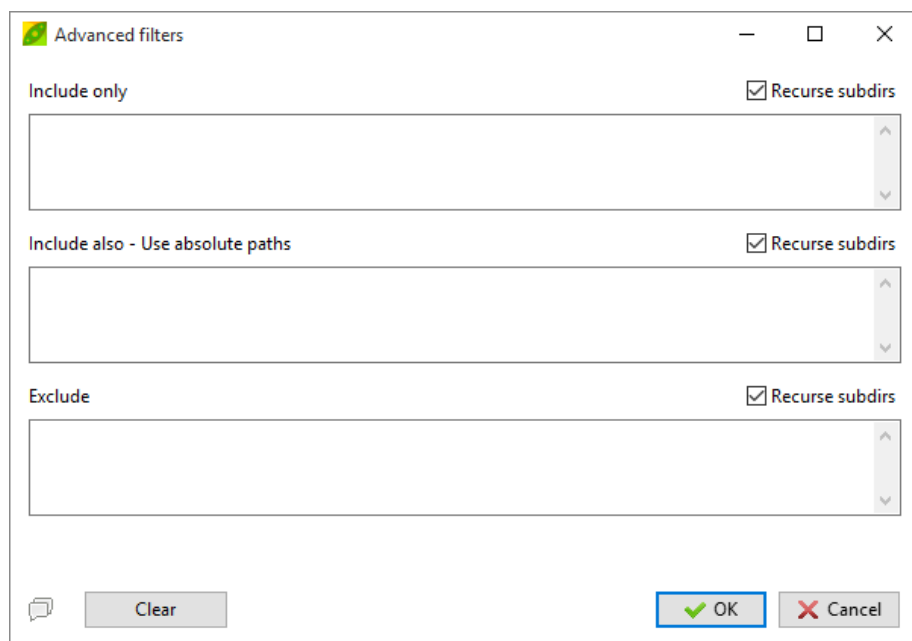
Image 22: inclusion and exclusion filters. "Include also" field is featured only for archive creation/conversion, not featured for archive browsing and extraction.

To exclude **directories**, use the syntax dirname\ which excludes all objects contained in the directory named "dirname".
This can be combined with wildcards:
      dirname*\ excludes all directories starting with "dirname" string
      dirname\ excludes all directories and subdirectories containing "dirname" in name or path
      dirname excludes all files and folders containing *dirname* in name or path
To selectively exclude **subdirectories**, the filter in previous example can be modified as dirname\subfoldername\ to exclude all items inside "subfoldername" which is inside "dirname".

Please note that archives can be built with **full, absolute, or relative paths** (from Advanced tab in archive creation screen), so filters needs to be modified accordingly.
In example adding C:\workfolder\archivefolder to archive using relative paths, the archive will have archivefolder in root, and subfolder1 in archivefolder can be added to inclusion or exclusion filters as archivefolder\subfolder1.
If, instead, the archive is created with absolute paths, the archive will have workfolder in root, and subfolder1 can be added to filters as workfolder\archivefolder\subfolder1
Finally, he string will become C:\workfolder\archivefolder\subfolder1 if the archive was created with full paths.

Advanced filters can be set from main menu, which shows the entry checked if advanced filters are in use, and from checkboxes in archive and extraction interfaces.
Please refer to original 7z and FreeArc documentation about inclusion and exclusion filters to understand how they works to get best result from this very flexible tool.

File tools

PeaZip collect also handy file management utilities

In file manager submenu:

Quick delete removes files and folders from the filesystem without moving them to the recycle bin as in standard deletion operation (featured in the same group), so the operation is quicker and not recoverable unless specific data recovery software is used.

Zero delete overwrites file content with all zero and remove files and folders from the filesystem without moving to the recycle bin.
Zero delete free space overwrites free space of a specified drive with all zero; zero deletion is marginally more secure than quick deletion with some data recovery software, anyway it is primarily meant to improve compressibility of the whole filesystem, in example when optimizing virtual machine disk images or backup disk images. Please note overwriting large volumes is time consuming, and may quickly wear flash based disks.

Secure delete is intended for securely remove files and folders from disk, avoiding possible data recovery; it provides multiple overwriting of file with random data stream (AES256 CTR) forcing flush to disk each iteration, then replacement of content with randomly sized random data to mask original file size, and multiple renaming with random string to mask original file or folder name.
Secure delete free space overwrites free space of a specified drive with random data stream (AES256 CTR) multiple times, forcing flush to disk each iteration. Please note overwriting large volumes multiple times is time consuming, and may quickly wear flash based disks.
In "Settings > File Tools" it is possible to set number of passes to perform (1 VERY_FAST, 2 FAST, 4 MEDIUM, 8 SLOW, 16 VERY_SLOW) over the data for Secure delete and Secure delete free space.

Notes about secure file deletion: please use this function carefully since securely deleted data usually will not be recoverable. Please note that secure file deletion doesn't overcome any known risk of data leakage, i.e. it may exist copies of the data as temporary files saved by other applications, or as not securely deleted older versions of the file, or cached by the system: wiping a file cannot affect that orphaned data, which may be recovered with software utilities or specific hardware probes. Moreover, flash based storage usually re-allocates sectors for writing transparently for the software, in order to reduce unit's wear since flash units have a shorter lifespan in terms of writes; this doesn't allow to efficiently physically overwrite original content, reducing the efficiency of file wiping; in those cases wiping the whole disk would be more effective, but (for flash based disks) it will lead to fast wear and reduced lifespan of devices.

Notes about secure free space deletion: please note that efficiency of software based free space deletion can be limited by underlying factors, i.e. disk areas locked or made not available from the operating system, or disk drivers, or even at lower level (BIOS/UEFI or disk logic).

Note about zero deletion and zero free space deletion: with standard deletion, data is removed from the filesystem's table of content, but remains on disk (that's why overwriting data is necessary for secure deletion). When compressing a filesystem, i.e. a backup disk image, or optimizing a virtual machine disk, all actual data in the filesystem must be handled so having files deleted with standard deletion does not reduce the amount of data the disk image contains. Using secure deletion may worsen the problem, as the random data written to disk to overwrite original data is not compressible. Zero deletion mode offers a solution in this scenario, as the data is overwritten with a regular pattern that can be compressed to extremely high ratios, effectively reducing this overhead.

In File tools submenu:

Analyze files and folders: recursively lists content of selected files/folders
Shows file properties, modification time, and size. For directory, size and content is shown, helping in identifying possible duplicates folders. Shows total number of files and folders, space occupation, estimation of compressibility, older and newer object's date/time and larger / smaller object's sizes.

Check files can perform in a single pass multiple user selected hash and checksum on multiple files at once. In Tools > Settings > File Tools it is possible to select algorithms to be performed over the input files (also available in archiving / extraction interfaces).

Compare files performs byte to byte comparison between two files; unlike checksum or hash based comparison byte to byte comparison can spot exactly what are the different bytes and it is not susceptible of collisions under any circumstance, even if this condition is highly improbable and very difficult or not practically possible to trigger if a proper hash function is chosen (also available in archiving / extraction interfaces).

Hexadecimal preview: a very basic tool to view the content of a file represented as hexadecimal values (also available in archiving / extraction interfaces).
Shows offset, hexadecimal representation of bytes, and possible utf-8 translation of each string of 16 bytes per row.
At current level of implementation please note that (1) not all rows could be correctly represented in the GUI as utf-8 strings, but they will be correctly written to file using save report feature (2) the implementation is limited to small files (< 16MB).

In columns header's context menu (when in details mode, and while browsing the filesystem - not archives)

Analyze content of folders recursively find and display content of folders (number of subfolders, files, and total content size)

Find duplicates is a fast routine to find duplicate files displaying the report directly in the file browser. This utility can be combined with all file browser's features, like F6 to display subfolder's content, or F3 to apply a search filter (recursive by default), for all displayed or only selected items. Only duplicates digests (for files with same size and same digest generated by fast checking routine) are reported in CRC column.

CRC32 / CRC32 (selected by default) calculate the checksum or hash for all items displayed in the file browser (or selected items), using the same control algorithms set for the duplicate finder. This is useful to check files for corruption or substitution when an original checksum or hash value is known, or to thoroughly test for duplicate files (slower than find duplicates routine).

Control algorithm used in the two previous functions can be selected from "Algorithm" submenu (or in Settings > File Tools > Browser) from a wide array of 15 algorithms: Adler32, CRC16/24/32/64, eDonkey, MD4, MD5, Ripemd160, SHA1, SHA256/512, SHA-3 256/512, Whirlpool512.
Ripemd, SHA, SHA-2, SHA-3 and Whirlpool algorithms are cryptographically strong, and suited to detect malicious modifications that might have been introduced to pass undetected by simpler control algorithms.

After running Find duplicates or Checksum/hash functions, Info column will report "D" for all duplicate files with same digest *excluding the last occurrence of the digest*, so selecting items marked with D allows, if desired, to quickly delete all duplicate instances leaving only a single copy of the file.

After running Analyze content of folders, Info column will report "E" for all empty folders, containing no files and no subfolders.

Extract archives

This interface is activated when one or more archives (even of mixed formats) are selected for extraction from file manager, or when browsing an archive the whole content or part of it is selected for extraction; the main application's menu is enabled (except for Browser submenu).

The form is organized in 4 tabs: **Extract**, containing most frequently used controls, is shown by default, **Advanced** (more fine grained options), **Console** (transform the task defined in the GUI into a command line, that can be saved as script for late usage, inspection, further customization, learning purpose, etc), and **Schedule**. On the right of the tab bar, the "+" menu icon shows entries to add/remove objects from the input layout, and display other input-related functions; the same menu is available as context menu right-clicking the input area.
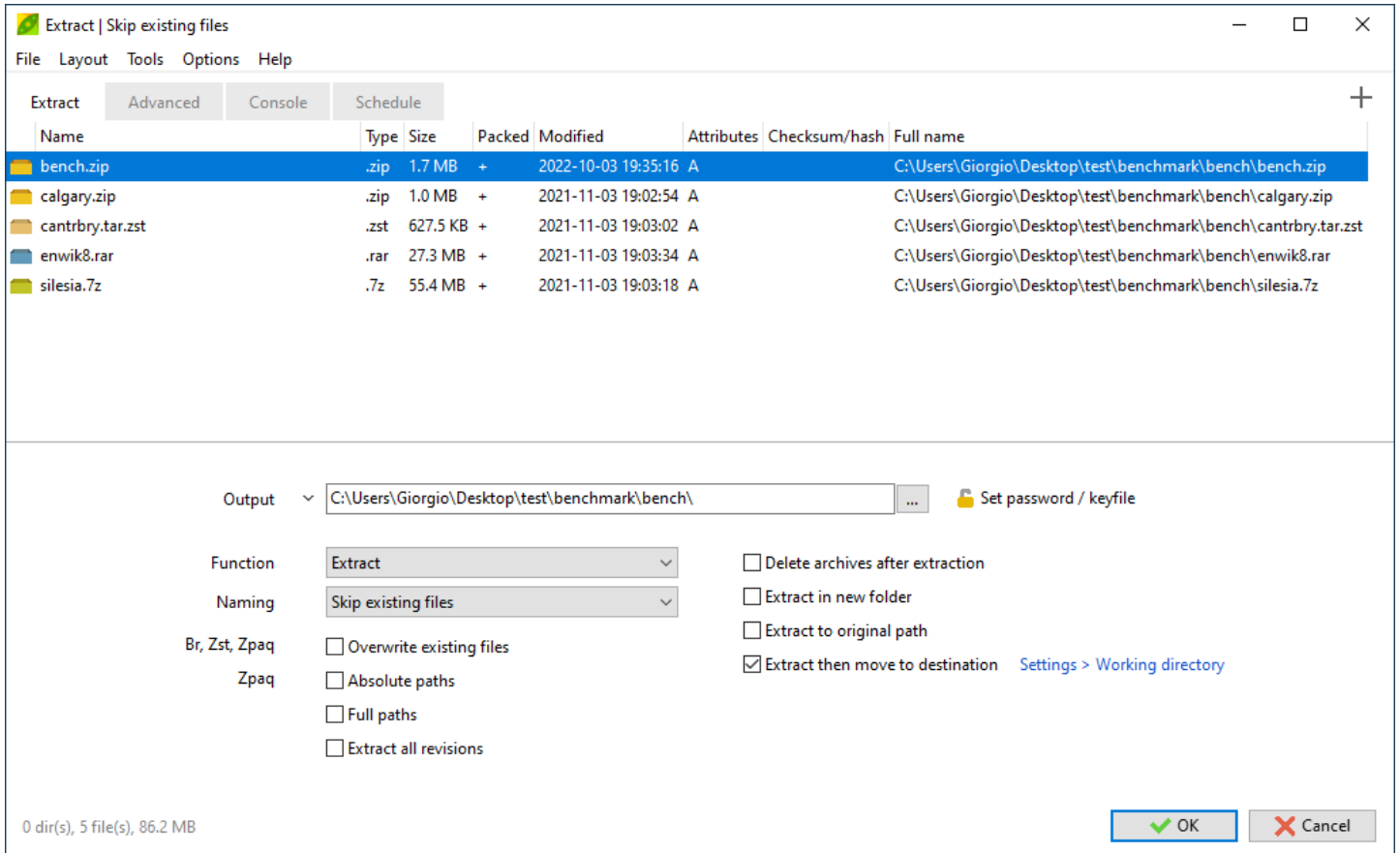

Image 23: Extraction interface: rightclick the input area, or click the menu icon on the top right of the application, to display the context menu to add/remove objects from extraction layout, and display other relevant functions.
Click on the dropdown menu's arrow on the left of Output path for quick selection in system's paths, favourites, and recently opened directories.
From extraction screen PeaZip can extract content from archive, extract single or multiple archives at once, even of different types and saved in different paths.
Format-specific options group for Zpaq, Brotli and Zstandard is shown, the group is hidden if none of the input files belongs tho one of those formats.

It is possible to skip this confirmation interface extracting archives directly from the system using "Extract here" or "Extract here (in new folder)" context menu entries.

It is also possible to skip confirmation interface and to extract directly from PeaZip's file/archive browser, in example:
- open the archive and drag desired files and folders to the desired output location, or drag with right mouse button (or pressing Ctrl) for a contextual menu to choose if to extract all or only selected items, and if to extract into a new folder
- open the archive, rightclick on an item and select "Extract and open with…" to perform both the two operations
- Click "Extract all to..." (shortcut: Ctrl+E or F12), or a quick extraction destination from the dropdown menu on its right:
    - Ctrl+Alt+E extracts archive in its current folder
    - Ctrl+Shift+E extracts to desktop
    - Ctrl+Alt+Shift+E extracts to user's documents
    - Ctrl+0 extracts to the previous output path
    - Ctrl+1..8 extracts to 1..8 favourite's path (if defined)
    - Ctrl+9 extracts to default output path (if defined)

The aforementioned keyboard shortcuts are available both from the file manager and the archive extraction interface, to directly start extraction bypassing the confirmation dialog.

Extract

This is the main screen for extraction tasks, containing most commonly used parameters and **Ok** and **Cancel** button to confirm start or to discard the current operation.
Clicking Ok saves modifications to extraction options, while Cancel discards the modifications; it is also possible to confirm changes with an empty layout in order to modify PeaZip's behaviour without the need to start a task.

Hint: remember that for simply changing extraction options, it is possible to click on "Extract" button
with no input selected (empty layout), modify options and save with OK button

In the top area of the tab is shown the **Input** area, where are listed the archives that are going to be extracted (**extraction layout**), with the total number of archives and total size; additional archives can be dragged here to be added to the list.
Objects in the archive layout can be sorted by name, full name, size, extension, type, attributes etc, clicking on titles in archive layout's title bar.

If extracting items which are stored inside an archive, the layout will display the items listed to be extracted from the archive, and will not accept modifications from drag and drop or context menu, and it will be available "**Extract all**" link below the input list to turn a partial extraction in extraction of the entire archive if desired.

Please note that PeaZip will, by default, **extract to a temporary work path** (which can be set from Options > Settings) and then **move extracted content to the output directory**, interactively asking to overwrite existing files if needed.
This behaviour can be changed from Options > Settings > Archive manager, Extraction section, unchecking "**Extract then move to destination**" option.
This will allow unattended extraction tasks, with existing files being handles as specified in the naming conflict policy (set in bottom dropdown menu on this screen) without requiring further interaction from the user.

Rightclicking on the input list shows the **contextual menu**, which features functions to add archives to extraction layout (add files & folders, add items from bookmarks and history, load layout...), and other related functions.
Files can be simply dragged from the system to the application: to help file selection "Search and drag here" opens a standard system's search prompt, and "Open path" submenu links system's file explorer to paths of bookmarked and recent items.
The layout can be saved to a UTF-8 text file (for maximum flexibility of use); when a layout is loaded each object is checked (must exist, duplicates are skipped).
From the context menu it is also possible to remove objects from the archive's layout ("Remove selected objects" and "Clear layout") and to explore selected object's path.
ZIP filenames encoding replicates a control to set a custom codepage to encode filenames in ZIP archives (please note this setting applies to .zip format only) from a list of known codepages; last menu element brings to Settings, Advanced tab which allows to set all text encoding related options.
Extract > Set default output path set all archive extraction tasks to point by default to the chosen path.
"Go to file browser" can be used to return to file browsing without discarding the current list of archives, in example to navigate and search for other archives to be added to the list from file browser interface.
"Open with..." submenu of context menu allows opening the selected object with PeaZip, associated application, or a custom applications.
"File Tools" submenu allows quick access to some of PeaZip functions to be applied on selected objects (see "File tools" chapter).
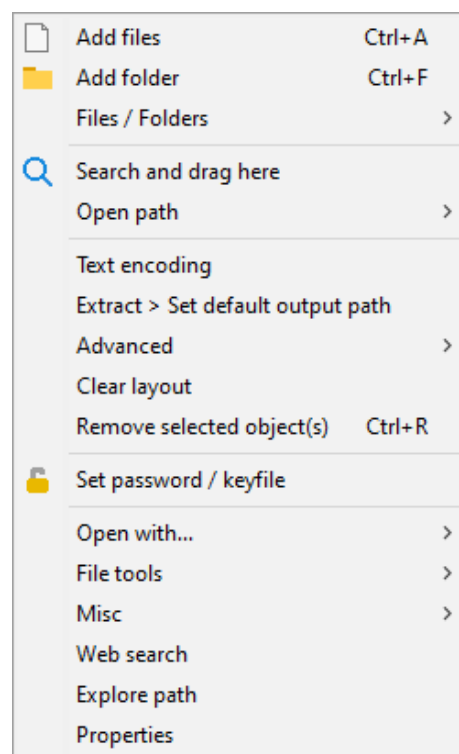
When extracting entire archives (single or multiple files selected) the complete context menu is shown, while when extracting items from a single archive non relevant entries are hidden.

In the bottom area of the form is featured **Output** group, that allows to select output directory; on the left of the address box, the arrow button popup a navigation menu similar to the one featured in file manager, to speed up the selection of most used paths.
The extraction's navigation menu is organized in **Bookmarks, Filesystem** and **Recent** submenus, containing only directory entries.
The menu also features "Use default output path" entry to select an output path to be used by default in place of archive's path, useful if the user often extracts to the same path.
When this feature is active, the "Use default output path" entry is flagged, and can be clicked again to remove the flag and deactivate the option.

On the right of Output field it is featured **Set password / keyfile button** to set the decryption password, if needed for extraction of encrypted archives.
If password is not set, the padlock icon of this button is greyed, otherwise the icon is displayed with normal colours and the bottom of the form will be highlighted with a colour representing the strength (automatically evaluated entropy) of the password, red for easy password, green for more secure ones.

Dropdown menus (on the bottom left of output area) features output options for 7z backend, that is used to support most common file formats ("Advanced" tab contains parameters for other backend executables).

- **Function**, possible actions performed by the extraction routine:
    - ◦ "Extract" extracts archived objects with paths, replicating the directory structure of the input data;
    - ◦ "Extract (without path)" will extract all archived files to the same path;
    - ◦ "List" will show archive's content;
    - ◦ "List (with details)" will give a more detailed report on archive's content, the same given by "Info" entries in context menu. List functions will always be performed in pipe mode (even if 7z option is set to "console mode"), using graphical wrapper in order to make easier reading and saving the report.
    - ◦ "Test" will perform type specific tests to prove or disprove archive's integrity.
- What the extraction routine will do in case of **naming** conflict while extracting data. Anyway, using "Extract in new folder" switch will assure avoiding naming conflict.
    - ◦ "Skip existing files" assure that pre-existing objects are not touched by the extraction operation, being the conflicting objects not extracted from the archive (default).
    - ◦ "Auto rename extracted files" assign a new unique name to objects being extracted from the archive each time a naming conflict is encountered; that policy assures that pre-existing objects will keep their names and new ones will get new unique names.
    - ◦ "Auto rename existing files" assures that extracted objects get the desired name while pre-existing objects are renamed with a new unique name.
    - ◦ "Overwrite existing files" make all pre-existing objects overwritten by extracted objects.
    - ◦ "Ask before overwriting (in console)": no defined policy, the console will be shown alongside graphic process launcher allowing the user to decide interactively if overwriting or skipping files.

Format-specific options group for Zpaq, Brotli and Zstandard becomes visible when extracting file(s) of those types, or if at least one file of those types is listed for extraction.
From this group it is possible to **force overwriting existing files** (off by default) for Zpaq Brotli, and Zstandard.
Zpaq backend can also be set to **extract to absolute paths**, **full paths**, or to **extract all revisions** of files in the archive – option which is mutually incompatible with extract to absolute paths option.
Please note that Zpaq extraction to absolute paths is set on by default for non-Windows systems, as on non-Windows systems the format is quite often used to create packages meant to be extracted to specific locations, rather than archives meant to be extracted in custom paths.

Checkboxes (on the bottom right of output area), can set following options:

Delete archives after extraction if no error was detected removes original archives after complete extraction process. Only enabled if "extract all" is selected.
The user is asked for confirmation before proceeding with deletion unless confirmation request is unchecked in Options > Settings > Archive manager.
If this option is in use, a dropdown menu on the right of the checkbox allows to set the type of deletion to be performed: move to recycle bin, quick delete (files are deleted without being moved to recycle bin), zero delete (files are overwritten by 0s, useful to optimize compressibility of disk images, if the host system is a virtual machine), secure delete (multiple pass overwriting of data).

Extract in new folder checkbox, when checked, triggers extraction to a fresh new folder, avoiding possible naming conflict and unwanted "tarbomb" effect (when an unexpectedly large number of files is extracted in current path getting mixed with existing files).

Extract to original path option can be used when multiple archives from different paths are listed for extraction, in order to perform extraction of each archive in its original input path.

Extract then move to destination option (default on) performs a two-step process for formats supported through the 7z / p7zip backend: extract to a fresh temporary path, and then move content to output path.
In this way
1) naming conflicts will be prompted interactively to the user, as in drag & drop extraction
2) tree structure of files/dirs in the archive will always been replicated, as with "Extract" switch, and extra nesting levels will be automatically removed, as in drag & drop extraction
3) in case of extraction errors, output will be automatically deleted, unless the user clicks on "Keep files" link in extraction progress dialog.

The option is always overridden by "Extract to new folder" switch, which is sufficient to guarantee no naming conflict occur, Unchecking Interactive extraction checkbox perform single step, policy based extraction.

Please note this option is always ignored for composing the command line in Console tab of Extraction screen, so the script which is generated will not need to rely on extracting to temporary work path first, and will not need user interaction in case it is needed confirmation to manage naming conflict issues.

More less commonly used options are available in the Advanced tab, and even more settings, which needs to be adjusted less frequently (on once for all), are available in Options > Settings, Archive manager page.

The status bar contains, on the left, information of free space on current output disk, and input size, and on the right OK and Cancel buttons.

Cancelling an operation also means discarding changes in option switches, while confirming with OK saves changes.

Advanced

Set archive manager options points to Options > Settings, Archive manager page, where it is possible to change the general behaviour of the application – while the options in extraction screen are more focused on current task and are more likely to get frequently edited.

Use advanced filters checkbox shows if advanced filters are set or not; clicking the unchecked box will show the advanced filter's form to set inclusion/exclusion filters, clicking the checked box will deactivate advanced filters.

Halt system when task completes checkbox schedule shutting down the system when extraction operation is completed.

Extract supported non-archive types checkbox, unchecked by default, allows some non-archive types like executables, MS Office and Open Office documents etc… to be treated as archives, in order to be disassembled.

In file manager those file types are displayed with special icons to show they can be accessed as archives by PeaZip, but will not be treated as archives by default.

Extract unsupported file types checkbox, unchecked by default, allows any arbitrary type of file to be treated as an archive, trying to extract it with PeaZip (checked by default) or specifying a custom executable backend to handle the file type in "Custom parameters" group, which is enabled if this option is checked.

This screen also allows to set parameters for backend executables 7Z / p7zip, FreeARC and Zpaq.

7Z / p7zip group allows to set (for current session) extraction to relative paths (default), or to full / absolute paths.

In this way it is possible to extract content directly to intended locations (absolute path includes drive letter in Windows), if the input archive was created archiving original items keeping full / absolute paths information.

Also, it is possible to choose to propagate Zone.Identifier stream to extracted files on Windows – which is especially useful for downloaded archives, treating all contained files as they were downloaded (same zone of the archive containing them).

For WIM (Windows Imaging Format) files it is possible to set to include or exclude NT security information and NTFS Alternate Data Streams in extract, list, and test operations - if such data was saved in the image file.

Notably, in this group it is possible to change action for FreeArc backend to "Repair", which (for ARC archives only) verifies integrity and tries to repair the archive(s) using the recovery records that may have been included at the archive creation.

Console, Schedule

Console screen allow to export the task defined in the GUI as a script, which can be interactively edited, tested and saved.

Schedule screen allows a simplified one pass UI to save the task definition and schedule it in Windows' system task scheduler (Vista and newer systems).

For more information, see "Save and schedule tasks" chapter.

Note about extraction

If Extract then move to destination (extraction screen) is unchecked, the default extraction mode recreates the full parent path's directory tree, as saved in the archive, in the extraction destination path – without removing extra directory nesting when extracting content from archive's sub directories.

This may be necessary to preserve the functionalities of reciprocal links of archived objects, i.e. link files, hypertexts, scripts, programs, databases etc…

It is possible to select "Extract (without paths)" in Function: this will flatten extracted content to a single directory, and original directory tree (both up and below the directory of the selected items) will not be preserved.

If **Automatic extraction of TAR archives from TAR.* compressed files / EXTRACT** (Options > Setings > Archive manager) is set, the application will extract compressed tar content in a single step (atomic extraction, default behaviour), otherwise the TAR archive will be simply uncompressed and the user can chose if to extract TAR content in a separate step.

In Options > Settings, Open archive tab, are also featured "Always ignore paths..." directives (which are overridden by "Extract then move" option), in order to extract items from the archive in a flat mode without recreating the directory tree, as with "Extract (without paths)".
These directives can be fine-tuned for extract and open with, and for extract selected or displayed objects (all objects shown with current browsing or search filter).

Unlike "Extract (without paths)", "Always ignore paths…" directives are overridden if one of the selected objects is a directory, this because flat extraction will lose information about directory tree so it is considered unfit for directories, and will be applied only if files are selected. The same rule applies when extracting objects from archives browsed in flat mode, consistently with behaviour of mainstream archive utilities.

Note about ARC format

The ARC backend always waits for user interaction in console if the archive is protected by password, and it is provided an incorrect password, or no password. This makes difficult to test password correctness when ARC is wrapped in a GUI application.

Testing, extracting, or updating an encrypted ARC archive will always need passwo4rd / keyfile.
Browsing and listing will need password / keyfile only if filenames in archive are encrypted - "Encrypt also filneames" option in PeaZip's password screen.

PeaZip will need the correct password / keyfile to be provided (in the main PeaZip GUI, in the password prompt) before starting operating on encrypted ARC files.
The operation will never complete if a wrong password / keyfile is provided, and can be terminated by the user form PeaLauncher Cancel button; in the archive browser the error should be caught and trigger a password request prompt (in case this not happens, the archive browser should be terminated).

As alternative, **it is possible to set PeaZip to use a console interface**, which allows the needed interaction in case of password mismatching:
- from Password screen checking "Force typing passwords interactively", which forces using the console interface for all operations, or
- from main menu Options > Settings, Advanced tab, Backend binaries group, setting Console or GUI+Console rather than GUI mode, please note this setting does not change the interface for some operations like List or Test (which uses the GUI in order to offer additional features like saving the output of the command as text).

In brief, when working on encrypted ARC files it is critical either to:
- provide correct password / keyfile in advance in the PeaZip main GUI, or
- use PeaZip in a console mode so it will be possible to respond to password error notifications

Note about UTF-8 and Windows console

PeaZip internally uses UTF-8 encoding to compose the commands to send to backed binaries (as well for extraction and for archiving).
Some of the backend binaries (Brotli, Zstd) may not correctly receive the UTF-8 characters from the Windows console depending on the system's codepage settings.
This issue does not apply on non-Windows systems.

Create / convert archives

This interface is activated when files and folders are selected for being added to an archive from file manager, or when, browsing an archive, it is requested to update it adding other files and folders. The main application's menu is enabled (except for Browser submenu). The same interface is activated using for archive **conversion**, which activates by default "Convert existing archives" and "Add to separate archives" options.

Clicking on dropdown menu on the left of the OK button (or in archive buttons group in file browser's toolbar) it is possible to select a **compression preset** to quickly apply all settings recommended for most common file archiving tasks, ranging from fastest compression algorithms to most powerful (and slow) ones, including special tasks like create a password protected archive, split archive in smaller parts, and create self-extracting archive.
"Immediate execution" option, if flagged, allows to start the compression immediately without requiring clicking on "OK" button for confirmation – but, as downside, it does not allow to customize the task further before launching it.

From same menu it is also possible to **save current compression settings**, and to load previously saved custom compression settings, allowing users to create their own pre-sets remembering basic and advanced compression options set in the GUI.
Some options are context-dependant, and even if they can be saved and loaded from custom compression settings they will be lost in subsequent usage, unless the setting is reloaded – volatile variables can be shown in italic, see Settings chapter.
Custom compression settings are saved in "Compression settings" directory in same path of configuration, while built-in presets are saved to "preset" folder in the program's path, but apart for being saved separately presets are standard compression setting files.
To replace an existing preset with a custom compression settings file, copy the desired custom compression settings file in PeaZip\res\presets folder, in place of one of the existing preset files. Optionally, editing the compression setting file is possible to change its displayed file name, the name that will be shown by PeaZip in presets menu.

The form is organized in 4 tabs: **Archive** creation, containing most frequently used controls, is shown by default, **Advanced options**, **Console**, and **Schedule**. On the right of the tab bar, the "+" menu icon shows entries to add/remove objects from the input layout, and display other input-related functions; the same menu is available as context menu right-clicking the input area.
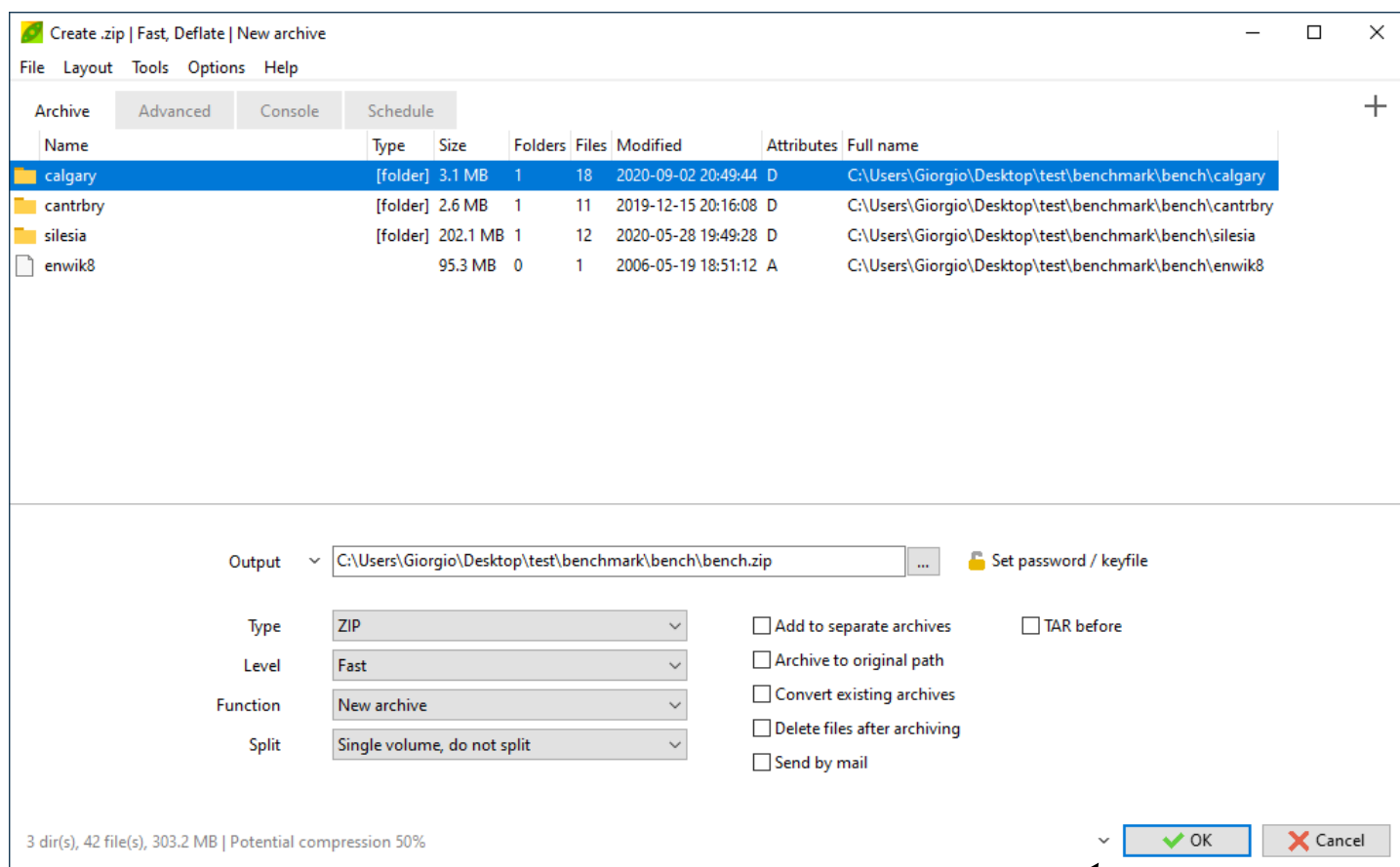
Image 24: "Create archive" interface.
Rightclick the input area, or click the menu icon on the right of tab bar, to display the context menu to add files and folders to archiving layout (add files, add folder, search ad drag here, load layout), and display other related functions.
Click on the arrow on the left of the output address for quick selection of the output path in system's paths, favourites, and recently opened directories.
Click on the arrow on the left of OK button to start compress/convert to the selected format.
From archive creation screen PeaZip create single or multiple archives at once, even from files/folders stored in different paths.

Archive

This is the main screen for archive creation (or update) tasks, containing most commonly used parameters and **Ok** and **Cancel** button to confirm start or to discard the current operation.
Clicking Ok also saves modifications to archive creation options, while Cancel discards the modifications; it is also possible to confirm changes with an empty layout in order to modify PeaZip's behaviour without the need to start a task.
Options will not be saved if forcedly modified entering a special operation mode, i.e. adding files to an existing archives.
Please note that some options (like TAR before or conversion switches, or some advanced format options) are context dependent and will be reset in next task instances.

Hint: for simply changing compression / conversion options, enter Archiving interface with no input (empty layout), i.e. clicking "Add" button when no item is selected in file browser, modify options and save with OK button.

In the top area of the tab is featured the **Input** group, where are listed the items that are going to be archived (**archiving layout**), with the total number of files and folders and total size; additional files and folders can be dragged here to be added to the list.

Right-clicking on the input list shows the **context menu**, which features functions to add files and folders to archiving layout (add files and folders, add bookmarks and history items, load a saved layout), and other related functions.
Files can be simply dragged from the system to the application: to help file selection "Search and drag here" opens a standard system's search prompt, and "Open path" submenu links system's file explorer to paths of bookmarked and recent items.
The layout can be saved to a UTF-8 text file (for maximum flexibility of use); when a layout is loaded each object is checked (must exist, duplicates are skipped).
From the context menu it is also possible to remove objects from the archive's layout ("Remove selected objects" and "Clear layout") and to explore selected object's path.
ZIP filenames encoding replicates a control to set a custom codepage to encode filenames in ZIP archives (please note this setting applies to .zip format only) from a list of known codepages; last menu element brings to Settings, Advanced tab which allows to set all text encoding related options.
Archive > Set default output path set all archive creation tasks to point by default to the cosen path.
"Go to file browser" can be used to return to file browsing without discarding the current list of archives, in example to navigate and search for other files and folders to be added to the list from file browser interface.
"Open with…" submenu of context menu allows opening the selected object with PeaZip, associated application, or a custom applications.
"File Tools" submenu allows quick access to some PeaZip functions to be applied on selected objects (see "File tools" chapter).
Files and folders in the archive layout can be sorted by name, full name, size, extension, type, attributes etc, clicking on titles in archive layout's title bar.

In the bottom area of the tab is featured **Output** group, which allows to select output directory and file name; on the left of the address box, the arrow button popup a navigation menu similar to the one featured in file manager, to speed up the selection of most used paths.
The archiving navigation menu is organized in **Filesystem**, **Bookmarks** and **Recent** submenus, containing only directory entries.

On the right of output field, it is featured **Encryption: Set password / keyfile** button, to set the **password** to create an encrypted archive, the button is only displayed if the selected archive format supports encryption.

From password popup it is possible to check/uncheck "Encrypt filenames" option, which, if supported by the selected archive format (7Z and ARC formats allows that option, PEA and ZPAQ always encrypt filenames) does not allow to browse the content of the archive if the right password is not provided.
Format-specific encryption options (e.g algorithm) can be selected in Advanced tab.
If password is not set, the padlock icon of this button is greyed, otherwise the icon is displayed normally and the bottom of the form will be highlighted with a colour representing the strength (automatically evaluated entropy) of the password, red for easy password, green for more secure ones.

Output options dropdown menus on the bottom left area set the archive **Format** and **Compression level**, more options for each format can be fine-tuned in "Advanced" tab; last used compression options will be remembered to be used as default, avoiding the need to set it in advanced tab following times the format is used; resetting applications default resets also compression options.
The info icon in the right of the archive type selection combobox will display a brief explanation of the characteristics and features offered by the selected format.

"Function" dropdown set the policy when a pre-existing archive with the chosen name is found on the system:
- **New archive** (default) will force the creation of a new archive, with unique name;
- **Add** adds the input objects to the archive, if an archive with the chosen name exists, overwriting any matching files;
- **Update** add the input objects to the archive, overwriting only older matching files already in the archive;

- **Update only files already in archive** "freshen" the archive content: works as update, but ignoring (not adding) input files not already in the archive
- **Synchronize archive with disk** works as update, but archived files not featured in the input will be deleted from archive – warning: please note the files removed from the archive will not be recoverable

In the same group of controls it is possible to set **Volume size**, allow (optionally) splitting the resulting archive in volumes of given size, choosing between pre-set sizes (from 1.44 MB FD to 8.5 GB DVD DL size) or freely composing a custom size.
This option is not supported by some formats (i.e. self extracting archives) and will be automatically overridden if not supported (the combo box will be greyed to be noticed by the user).

*Hint: to store large files/archives on **small supports**, or to respect **mail attachments size limitations**, it is recommended to split the output in volumes of desired size instead of spending more time for a deeper compression which is not always capable to reduce the output under the desired size.*

Options checkboxes on the bottom right:

Add to separate archives send each item in the list to a separate archive.
In example, adding x folders to the archive list with this option checked will create x separate archives; if it is desired to add each file contained in a folder to a separate archive, instead, select the files in file manager and click "Add" to add them individually to the list.

Archive to original path option can be used when multiple files/folders from different paths are listed for archiving or conversion, in order to create each archive in the same path of each input item; flagging this option will consequently activate "Add to separate archives".

Convert existing archives extracts listed archives (not modifies non-archive files and folders) in order to recompress the original data from its uncompressed form, which can improve compression ratio.
Used with "Add to separate archives" option (default, if "Convert" function was requested), it performs a mass conversion of each listed archive into the desired format, keeping them separated.
"Change password" option (default off) asks to provide a new password to be applied to converted archive(s), otherwise current - most recent, if any - password will be applied.
"Verbose" option asks for confirmation before proceeding with some steps, allowing further customization of the task:
- After the extraction stage it asks for confirmation to proceed with compression, this allows to abort the process in case of problems (disk full, unknown password, corrupted archives etc) or to further modify the extracted data before the compression.
- After the compression stage, it asks for confirmation before deleting temporary files and folders created for conversion, however original archives are not deleted and not modified in any stage of the conversion to let the user in control about keeping or removing them.

If **Automatic extraction of TAR archives from TAR.* compressed files / CONVERT** (Options > Setings > Archive manager) is set, the application will extract compressed tar content before re-compressing the archive in the new format, otherwise TAR containers will be let unaltered and simply re-compressed in the new format (default).
Please note you may need to manually set the **TAR before** switch if the converted archive contains multiple items but the target format is a pure compression format (like gz) expecting a single file as input.

Delete files after archiving removes original files after archiving process, the user is asked for confirmation before proceeding with deletion., unless confirmation request is unchecked in Options > Settings > Archive manager.
If this option is in use, a dropdown menu on the right of the checkbox allows to set the type of deletion to be performed: move to recycle bin, quick delete (files are deleted without being moved to recycle bin), zero delete (files are overwritten by 0s, useful to optimize compressibility of disk images, if the host system is a virtual machine), secure delete (multiple pass overwriting of data).

Send by mail (only for formats supported through 7z backend) attach the resulting archive to a new email message in the default mail client (if supported); it doesn't work with multi volume archives (the options is hidden if Volume size is not "Single volume").

TAR before option allows to consolidate all input objects in a single TAR archive, temporarily saved in the output path, which will then (in a second pass) be compress/encrypt/split using the format specified in archive type combo box (after that second pass the temporary TAR archive will be deleted).
That option allows easily merging the advantages of TAR format (of mainstream and standard usage on most Unix systems) with features of other archive formats. It is especially useful as it allows to select compression-only formats (like gz, bz, quad) to compress archives of multiple objects (resulting i.e. in tar.gz or tar.bz or tar.br, tar.zst) quite transparently for the user, but however it can be used in conjunction with any format (forming i.e tar.7z, tar.paq, tar.pea and so on).
The application will try to check if "TAR before" option is needed; however the user can check/uncheck this option anytime before launching the archiving process.

More options (mostly related to advanced features of the selected compression format) are available in the Advanced tab, and less frequently modified settings are available on Options > Settings, Archive manager page.

The status bar contains, on the left, information of free space on current output disk, input size, and estimated output size.
On the right are features OK and Cancel buttons: cancelling an operation also means discarding changes in option switches, while confirming with OK saves changes.

The arrow on the left of OK button shows **pre-set compression profiles and favourite formats** menu.
Compression profiles shows a pre-set list of most common file/archive management operations, in order to show inexperienced users how to solve common problems without the need of searching for appropriate options (i.e. how to add a password, how to keep the output below attachment size limit, etc), but also to offer experienced users a single handy switch to perform common tasks without the need of setting one option at time.
Favourite formats can be customized in Options > Settings > Create archive, by default it displays most commonly used formats.

Advanced

Set archive manager options points to Options > Settings, Archive manager page, where it is possible to change the general behaviour of the application – while the options in archiving / conversion screen are more focused on current task and are more likely to get frequently edited.

Use advanced filters checkbox shows if advanced filters are set or not; clicking the unchecked box will show the advanced filter's form to set inclusion/exclusion filters, clicking the checked box will deactivate advanced filters.

Halt system when task completes checkbox schedule shutting down the system when compression/conversion operation is completed.

Add ".enc" suffix to encrypted files adds the suffix before file extension, so encrypted files are easier to recognize – and can be easily grouped with filters for batch processing.

Append timestamp to archive name will append current date and time to the name of the archive, it is useful for archiving and backup purpose. A dropdown menu set the syntax of the timestamp, default: FILE.yyyymmdd.hhmmss

Convert non-canonical archive types (default off) treat such files as archives during Convert procedure (extract content and recompress in the new format), unchecking the option let the files unchanged – which is recommended as such formats may contain extra fields (e.g. digital signature, associated data etc) which are out of the scope of the application

Custom extension allows to replace standard output archive extension with the one entered (with or without dot) in the text field.

This screen also contains fine-grained options for each supported format, including compression level.
Once advanced options for a given format are set, they are remembered, so it is usually not needed to use Advanced tab and all of archive creation can be done form Archive tab.

*Hint: most of **multimedia file formats** (like jpg, png, mpg, avi, mp3...) are already compressed with algorithms that are either lossy and/or strongly optimized for the specific data structure, so compressing them with general purpose lossless compression algorithms, even most powerful ones, are likely to provide only marginal benefits in terms of size, compared with huge benefits which can be obtained compressing other types of files (bmp, tiff, doc, xls, txt, html...).*
Consequently, it is recommended to use fastest compression settings, or fast algorithms (i.e. gz and zip's deflate), or even "Store" option to don't compress, to archive those types of files in a computationally efficient way; in this way it is possible to consolidate and optionally encrypt files that are sent to the archive, without spending much time for compression.

Backend executables can be used by PeaZip in the native console mode or, by default, through a graphic wrapper form (see "PeaLauncher" chapter), displaying additional information usually not given by the console application (exit code explanation, input size, elapsed time in ms, speed in B/ms) and allowing to save a task report (console output plus additional information); this behaviour can be customized in Options > Settings.

7Z / p7zip - *Performances scales very well on multicore machines; following formats are recommended:*

7Z when high compression is desired; offers powerful encryption

ZIP to provide archives which all Windows user can read with integrated "compressed folders" utility, or with most of mainstream file/archive managers

TAR to provide archives which most Unix (Linux, BSD …) users can read with applications usually bundled by default

WIM to create disk images for Windows environment

GZ, BZip2, XZ to compress single files or uncompressed containers, like TAR; GZ is the fastest, BZIP is slower but compresses better, XZ utterly improves compression but requires more memory and computing power.

Selecting a format supported through 7z executable (7Z, Bzip2, GZip, self-extracting 7Z, TAR, WIM, XZ, ZIP) it will be displayed the 7z option subpanel, featuring Options, Compression and Encryption group box.
"Compression" group box allows choosing compression level and algorithm and to fine tune compression options, which are format specific.

Last used compression level and method is remembered by PeaZip for 7z backend, but other custom options (dictionary, word, passes, solid block size) will be remembered only for the current session (until the archive type is changed or edited) and next times the last used compression level will be proposed with its default settings.

"TAR" option allows to choose the standard, GNU or pax
"Timestamp" sets timestamp precision for ZIP and TAR/pax archives.
From this group of settings is it also possible to set archive timestamp from more recent file in archive, and not to change last accessed date/time.
 "Create self-extracting archive" options create a Win32 executable (.exe) that will self-extract archive content (archived and compressed in 7z format); the receiver will then not need any application to extract the archive since all what is needed for extraction is embedded in the archive itself. As drawback the resulting file will be about 80KB bigger than the raw archive and the executable must be in a single volume (Volume size option will be greyed). Checking this option the user can choose between a console and a graphical interface (default) for the self-extracting application.

"Encryption" group box contains encryption related options:
- "Algorithm" allows to choose encryption algorithm; 7Z format supports AES, while ZIP supports AES and ZipCrypto algorithm. AES is always used with 256 bit keys. ZipCrypto is a weak algorithm but may be useful if the user need to generate encrypted .zip archives compatible with some outdated applications not supporting new WinZip standard AES-based AE encryption.

"Options" group box allows choosing other format specific options:
- "Function" replicate the same control in main Options tab.
- "Paths" - non persistent, kept for current session only - allows to save to archive **relative paths** of input items (default), or to preserve **full path** or **absolute path** (includes drive letter) information to allow extraction to original locations (option featured in archive extraction, advanced tab).
- "Other" edit box **allows to freely enter additional parameters** for the archiving task. This string is inserted by default after all the parameters set by the GUI and its syntax is not checked, so use with caution.
- "Threads" combo box allow specifying the number of **threads** to try to generate for parallelising and speeding up the execution of the application (possible only for LZMA, LZMA2, Deflate, Deflate64 and BZip2 algorithms) on Hyper threading-enabled / multi-core / multi-CPU environments; on Windows systems single processor systems will be recognized and will use "no multithreading" option by default while multi processor system will use "generic multithreading" option by default; on non-Windows systems "no multithreading" is the default option.
- "Maximum memory usage" allows to limit the **memory** used by the task.
- "Compress files open for writing" allows to add files to the archive even if **open for writing** by other applications, very useful when running backup tasks (otherwise, if the option is not checked, open for writing files will be skipped); the last used setting is remembered.
- For WIM (Windows Imaging Format) files it is possible to set to include or exclude **NT security information and NTFS Alternate Data Streams** in archive, convert, and update operations.
- For TAR archives on Linux systems "**Store owner/group ids**" and "**Store owner/group names**" allows to store those additional information in archive.
- "Store hard / soft links as links" allows **not to follow links** when creating the archive, which will contain only the link object rather than the linked content

ARC - *Recommended when high compression is desired; offers powerful encryption and, optionally, recovery records. Performances scales very well on multicore machines.*
In FreeARC's options subpanel, it is possible to adjust compression level, specify file grouping strategy for solid archives, create recovery records to attempt archive's repair in case of corruption, encrypt the archive with various encryption algorithms (AES,

Serpent, Twofish, and Blowfish), archive with absolute or relative paths (default), create self extracting archives selecting from many sfx modules.

Brotli - *Recommended when very fast compression and extraction are required*
Selecting Google's Brotli algorithm, compression level from 1 (faster) to 9 (slower, more powerful) can be selected from main archive creation interface.
Being Brotli a compression-only algorithm, "TAR before" switch will be automatically activated for handling multiple input files.
"Maximize Brotli compression using more memory" in advanced options uses large window mode, which can improve compression ratio for large files.

Custom / RAR - *Allows to select an external compressor/decompressor, including Rar.exe, to support extra file types*
Selecting "Custom" format, it will be displayed a subpanel (in Advanced tab) allowing to specify archiver/compression utility to use to perform the task, alongside parameters (free editing) and syntax (the way parameters, input list and output name should be organized on the resulting command line).
Last 8 used custom executables are remembered and can be chosen from a popup menu rightclicking on executable's selection control.
Please note that exact syntax of the command for a custom executable may need to be utterly adjusted, this can be done in "Console" tab, which allows importing and free editing of the task definition.
Checking **RAR** checkbox in Advanced options tab, switches to using WinRar's Rar.exe syntax to **create RAR archives**, with ready to select RAR-specific options (create recovery records, use BLAKE2 checksum, use RAR5 standard, etc).
If WinRar is installed in a predictable path, PeaZip will automatically configure itself to find Rar.exe, anyway clicking "Manually set RAR binary" allows the user to specify where the binary is, in order to make easier to use this function especially on non-Windows systems resorting to Wine or similar mechanisms to run Rar.exe.
Unchecking RAR checkbox allows free selection and editing of custom compression executable as explained before.

Pea - *Recommended when powerful encryption and integrity check are desired; provides fast compression*
Pea archive format provides optional triple cascade encryption (AES, Twofish, and Serpent 256 bit in EAX mode), it is developed from the author of PeaZip and it is supported through Pea executable; for more information you can refer to the documentation about Pea on help and FAQ pages on PeaZip official website.

QUAD/BALZ/BCM - *Recommended when it is desired to provide average to good compression, with fast uncompressing*
Selecting QUAD/BALZ/BCM, the option subpanel will allow choosing compressor engine, and compression parameters (more can be set manually in "Console" tab). QUAD, BALZ, and BCM are simply compression algorithms not providing archival functions, and likewise other pure compression formats (GZ, BZ2, LPAQ) they can benefit of "TAR before" switch for handling multiple input files.
BCM is supported out of the box, while other formats needs "Additional Formats Plugin".

Split - *Recommended to split a single large file to the desired size, without attempting compression*
Options for file split are limited to optional integrity check algorithm to be performed on the file; integrity check information will be saved on a separate file, allowing files splitted by PeaZip to be joined by other applications (like Unix split, Hjsplit, FileTools etc).

UPX - *Recommended to developers to reduce the size of executables*
UPX compression is intended mainly for developers needing to reduce the size of executables before distribution.
It can accept only a single executable file at time and is not intended to be used as a general purpose compression utility; in fact misapplying Strip and/or UPX on non suitable executables (i.e. already stripped executables) may easily lead to unusable output executables.
It is possible to omit either UPX compression (selecting "do not compress" in "Compression" combo box) or Strip pre-processing (unchecking "Strip before UPX").
By default it is created a backup copy of the executable before Strip/UPX (option "Keep executable's backup"), named as the executable with .backup extension appended.
Please note UPX compression hampers further compression, i.e. if executables are part of a package which is then re-compressed before distribution, final result may be worse than without prior UPX compression of single executables – this does not apply to stripping of executables, which simply removes unnecessary data and does not change the possibility of compress it afterwards.

Zstd - *Recommended when very fast compression and extraction are required*
Selecting Facebook's Zstd compressor, the compression level from 1 (faster) to 19 (slower, more powerful) can be selected from main archive creation interface.
Being Zstandard a compression-only algorithm, "TAR before" switch will be automatically activated for handling multiple input files.
"Maximize Zstandard compression using more memory" in advanced options tab uses large window mode, which can improve compression ratio for large files.

*PAQ - *Recommended when highest possible compression is desired (experimental; speed and memory usage makes it not recommendable for general purpose use on current generation machines)*

PAQ is a very powerful compression scheme and it's presently in research state; different versions and branches exist and you should use the very same implementation to compress and uncompress data, PeaZip uses PAQ8O for compression.

It brings unmatched compression ratio, better than any other compressor; the downside is that the algorithm has very high memory and computing power requirements for today's machines, resulting to be very slow if compared to mainstream compression algorithms, so the user should carefully consider if the speed/compression trade-off would be advantageous case by case.

To obtain best results while compressing many small files you should also consider consolidation them before (i.e. using tar) since PAQ would store filenames (and sizes) in uncompressed form.

ZPAQ is faster and lighter than PAQ, at the cost of a slightly inferior compression ratio.

Note about Windows EXE files

PeaZip can create EXE files as SFX (self-extracting archives) both employing 7Z and ARC compression.

It is possible to open / browse and to edit EXE files as archives, but – due to the multiple standards available for the format - not all types of EXE files can be read (for browsing) or written (for editing).

Whenever the user attempts to edit the content of an EXE file PeaZip assumes the file is a 7Z SFX archives, because this structure can be bot read and written by the application.

As a role of thumb, to edit the content of an EXE file - which is not an SFX archive - it is possible to rename the file form .exe to .zip extension and to revert the extension back to .exe once done editing.

However please note that acting as a resource editor for EXE files is beyond the scope of the application, so edited EXE files are not guaranteed to work correctly – conversely, editing 7Z SFX EXE file is fully supported.

Console, Schedule

Console screen allow to export the task defined in the GUI as a script, which can be interactively edited, tested and saved.
Schedule screen allows a simplified one pass UI to save the task definition and schedule it in Windows' system task scheduler (Vista and newer systems).

Saving and scheduling an archive creation script is very similar to **define a backup task**.
Most relevant options for backup usually are:
- Archiving action by default is "Create new archive": alternatively action **Update** will spare time and disk occupation, only adding new files and overwriting older matching files; **Update only files already in archive** action will only "freshen" the archive, overwriting older matching items and ignoring (not adding) any item not already in the archive
- From advanced option "Compress files open for writing" allows to backup files open for writing by other applications, otherwise those files will be skipped
- From advanced options it is possible to set to include or exclude NT security information and NTFS Alternate Data Streams, for WIM format only
- **Append timestamp to archive name** will avoid naming conflict with other backup copies in same output path

Please note that exporting GUI task as script, if **Convert** archive flag is set, calls PeaZip binary itself rather than backend binaries.

This is needed because archive conversion, as performed by PeaZip, needs extra steps which backend binaries may be not able to perform, in example:
- dynamically create a temporary fresh new work folder in which extraction step will take place (some binaries will not force creation of non-existing dirs)
- avoid to add unwanted extra level of directory nesting in the resulting archive (this would be complicated and not always possible to automate relying on backend binaries only)

This solution limits the conversion to either run with current PeaZip's default options (not requiring interaction, less flexible but preferable for batch use), or to prompt the full GUI to let the user customize the task on the fly (requiring interaction before the actual process takes places).
In Options > Settings > Archive manager tab, Archive group, the option "Archive conversion scripts allow user's interaction" (default on) sets this behaviour.

It is obviously also possible to use PeaZip to create an archive conversion script directly calling backend binaries instead:
- export as script the archive(s) extraction step from Extract screen, taking care to avoid naming conflict with existing files
- export as script the archive(s) creation step from Archive screen, pointing to the extracted content as input
- take care of removing the un-compressed content when no longer necessary, and optionally remove original input file

In this case it will be up to the user to manage possible issues and errors, but of course this will also let the user more room to fit specific needs better than the automated archive conversion routine of PeaZip- i.e. arrange extraction and archiving scripts so each input is extracted, re-compressed, and unneeded files immediately deleted so the process will require less free disk space.

For more information about creating scripts and define scheduled tasks, see "Save and schedule tasks" chapter.

Keyboard shortcuts and mouse controls for archive extraction and creation

Operations:

Archive / Extract (as Ok button)	Ctrl+Alt+Enter
Cancel (as Cancel button)	Shift+Backspace

Options:

Toggle extract to new folder on/off	Ctrl+N

Archive types

Change archive type to 7Z	Shift+Ctrl+7
Change archive type to BZip2	Shift+Ctrl+B
Change archive type to GZ	Shift+Ctrl+G
Change archive type to 7Z self extracting	Shift+Ctrl+S
Change archive type to TAR	Shift+Ctrl+T
Change archive type to ZIP	Shift+Ctrl+Z

File tools:

Checksum/hash selected files	?
Compared selected file with…	=

Archive layout-related: (the archive list is selected)

Add file(s)	Ctrl+A
Add folder	Ctrl+F
Add from search dialog (drag to archive)	(context menu only)
Load archive's layout	Ctrl+L
Save archive's layout	Ctrl+S
Open object with default application	Enter or doubleclick
Remove selected object from archive's layout	Cancel or Ctrl+R or Ctrl+Backspace
Refresh	F5 or refresh icon on the left of layout's titles row

Extract without confirmation (from browser and extraction interface):

Extract all to… (open selection dialog)	Ctrl+E
Extract all here	Ctrl+Alt+E
Extract all to Desktop	Shift+Ctrl+E

Mouse controls

Doubleclick: open selected object with associated application or browse folder
Rightclick: activate "create layout" context menu;

Save and schedule tasks

From "**Console**" screen, featured in archiving/conversion and extraction UIs, it is possible to transform the task defined in the GUI interface into a command line script that can be edited, independently from the task definition in the GUI frontend.
To reset the generated script discarding changes, click the **backward arrow icon** on the left of "Launch" button.
When performing partial (selective) extraction from archive, the same icon can be right-clicked to modify the command line for operating on all, displayed or selected objects - if partial extraction of content is supported for the current archive type (7z, arc, rar, tar, zip…).

The task definition script can be edited, launched for testing, or saved as a text file for future use, i.e. study, scripting, analysis etc.
By default, PeaZip prompts to save script in its "Saved scripts" folder (unique for each user) that can be opened from Tools in main menu.
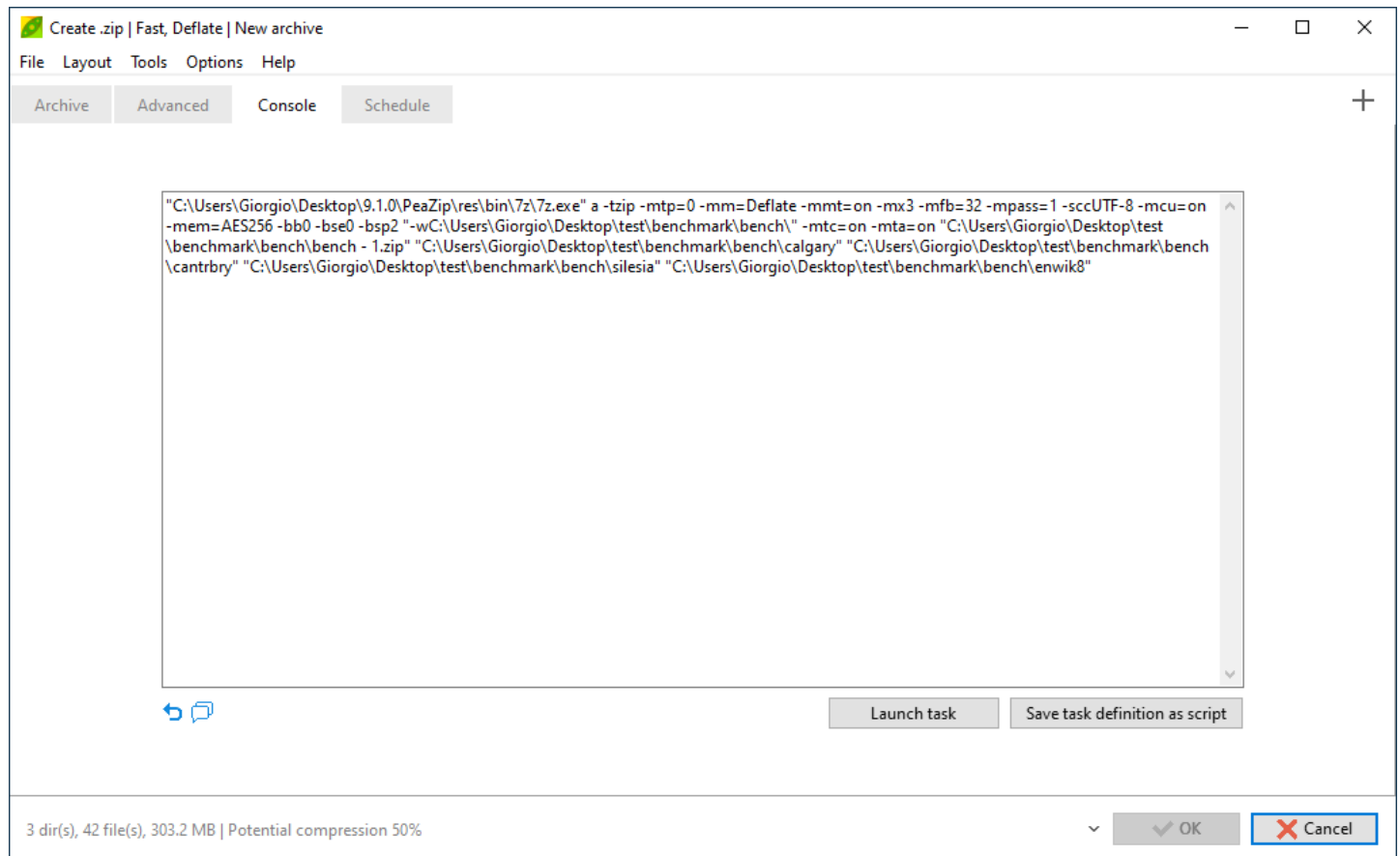
Image 25: Console tab: the command line corresponding to the task generated in the GUI

The checkbox "**7z / p7zip use pipe TAR/UNTAR**" allows to set (independently for archiving and extraction operations) the script generation engine to use stdin / stdout pipe to chain tar and compression/extraction routines.
In this way there is no need to temporarily save the intermediate tar archive when operating on compressed tar files.
Tasks defined in this way cannot be directly launched, but can be saved as script files to run separately.
This option apply only for formats supported by 7z / pzip (7Z, BZ, GZ, XZ, ZIP), and it is used only to generate scripts in Console tab.

Image 26: Schedule tab

"**Schedule**" screen is also featured in archiving/conversion and extraction interfaces, only on Windows Vista and newer systems.

It is meant to help creating, in a single pass, a schedule to run the script generated from the task defined in the GUI interface.

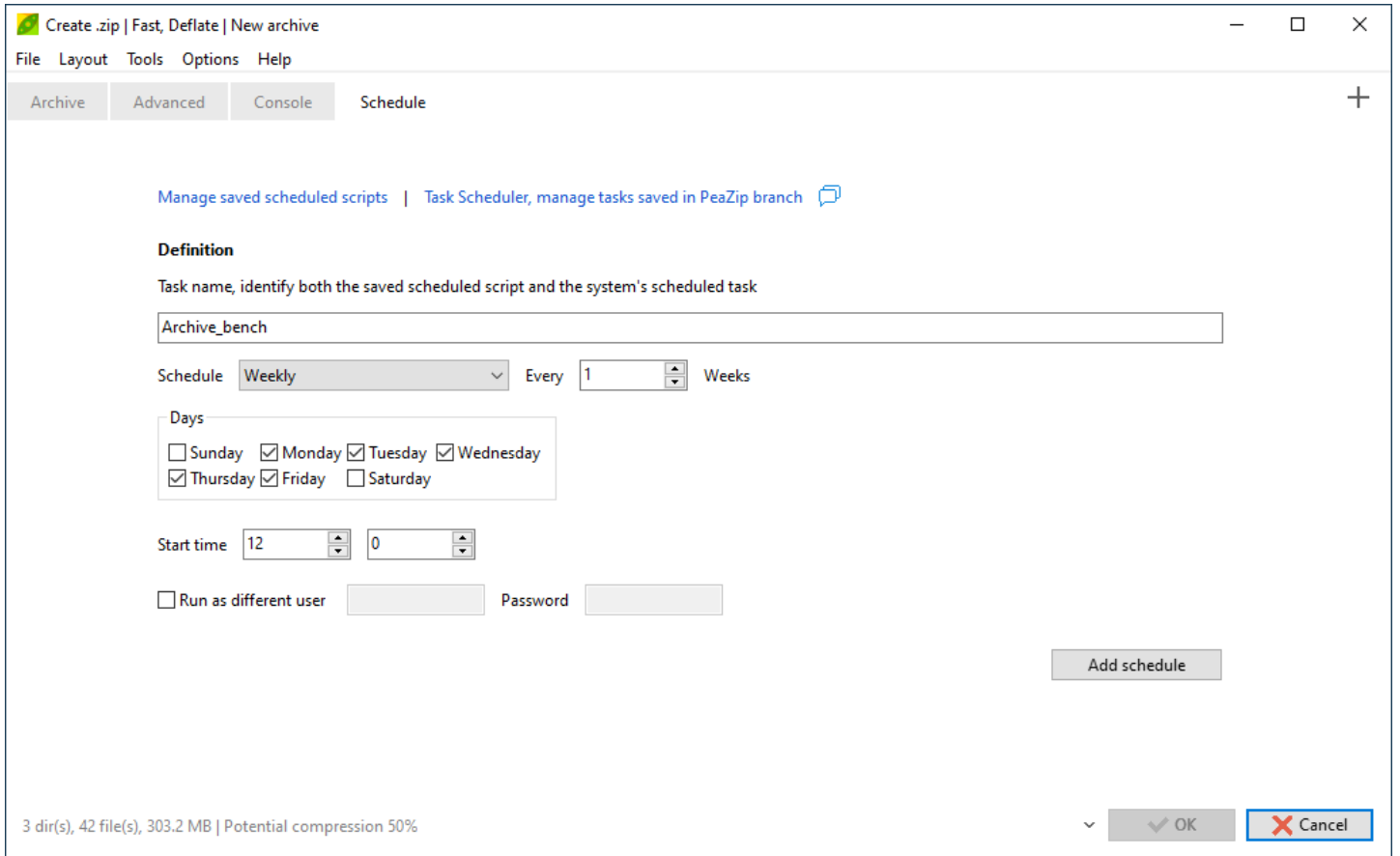The schedule is identified by an unique, editable name, which is used also as file name for the saved scheduled script.

"Manage saved scheduled scripts" opens PeaZip's Saved scheduled scripts folder so script files can be edited to modify the task definition (what does the task do), copied, deleted etc.

"Task Scheduler, manage tasks saved in PeaZip branch" opens system's Task Scheduler, PeaZip's tasks are collected in PeaZip subfolder of scheduled tasks library (tree on the left of Task Scheduler window).

From system Task Scheduler it is possible to manage PeaZip's scheduled tasks, i.e. for fine tuning the schedule (when does the task run), or deleting obsolete tasks.

Schedule dropdown menu allows to set the schedule on an hourly, daily, weekly, or monthly base, or once, starting at specified start time.

Optionally, it can be provided different user/password for running the task.

PeaLauncher

PeaLauncher is a GUI component that graphically displays information about the underlying running task(s), performed by multiple console-based backend binaries used by PeaZip to create, extract, test or list multiple archive formats.

It allows to inspect task progress, check estimated remaining time, pause or cancel tasks from buttons on the bottom, send the launcher to task bar (where it can be controlled from a context menu, or resumed to foreground), save the task definition as command line, and save task report after execution.

Title bar displays number of the current task / total number of tasks submitted.

Estimated % progress of current task is graphically represented by the progress bar on the bottom of the application window; if multiple tasks were submitted a second progress bar is shown under this one, representing the overall progress of the sequence of tasks.

On Windows 7 or more recent Windows systems the global progress is also shown in application's instance icon on system taskbar.
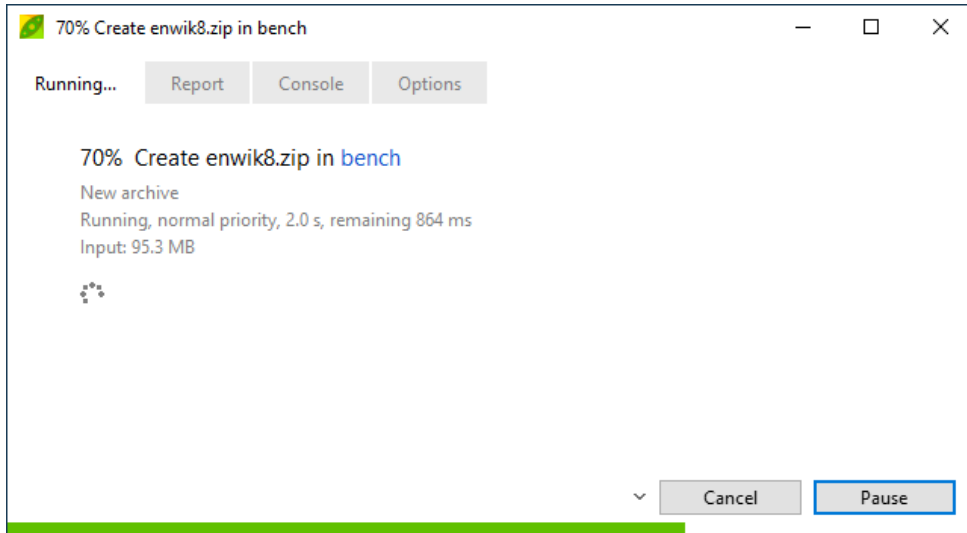

Image 27: PeaLauncher instance

By default, the window closes if no errors are detected, but remains always open for test, info and list tasks which requires the user to read the task's report; it is possible to always keep the task's window open after task termination so it is easier to inspect task's log and command line, selecting "Always keep open" in PeaLauncher combo box in Options > Settings.

The form is organized in four tabs

- **Status** (default), showing essential information and graphic about the process. The dropdown menu on the left of Cancel/Pause buttons allows to change process priority and stop all subsequent submitted tasks.
- **Report**, showing detailed feedback from the process (through stdout)
- **Console**, displaying the exact command executed – for inspection or learning purpose
- **Options**, allows optional actions to be to set at the end of the current list of tasks: open the output path, shut down the system.

The graphical wrapper is not invoked if tasks are running in console interface, either because they need to run in console (UNACE, UPX) or because the user selected to use "Console interface" in program's options; however list and test always run in graphic wrapper mode using pipes to give readable task log.

In console mode, the console will automatically close when task is completed without prompting any message if the task completes without errors, otherwise an error report will popup.

PeaLauncher can run also in "GUI + console" mode (see Settings chapter) displaying both the graphical interface and the underlying task running in its native console interface, showing its native progress indicator, usually more detailed than GUI's one.

Settings and Help system

Options > Localization brings to Options > Settings, General tab to set application's language and System context menu entries language (Windows); the application will be restarted with chosen localization.

Options > File associations and system menu integration (Windows only) runs a wizard to select types of files associated with PeaZip and menu entries in system's context menu and SendTo menu.

Options > Settings brings to main tab of Settings and Help interface, which allows to configure behaviour and aspect of PeaZip application. Set advanced archiving options and Set advanced extraction options displays operation-specific settings, same as shown in archiving/extraction screens.

Help > Help (F1) and About brings to respective tabs.
About tab shows the version of embedded binaries, shows installed plugins and allows to selectively remove plugins based on closed source components (UNACE and UNRAR). Please note rar extraction is supported out of the box by PeaZip, and the UNRAR plugin based on WinRar code is offered only as alternative extraction engine.

Reset button asks for removing current PeaZip configuration (includes recent archives history), and separately for removing other customization files: Applications (to reset custom file associations inside PeaZip), Bookmarks, and Password manager.
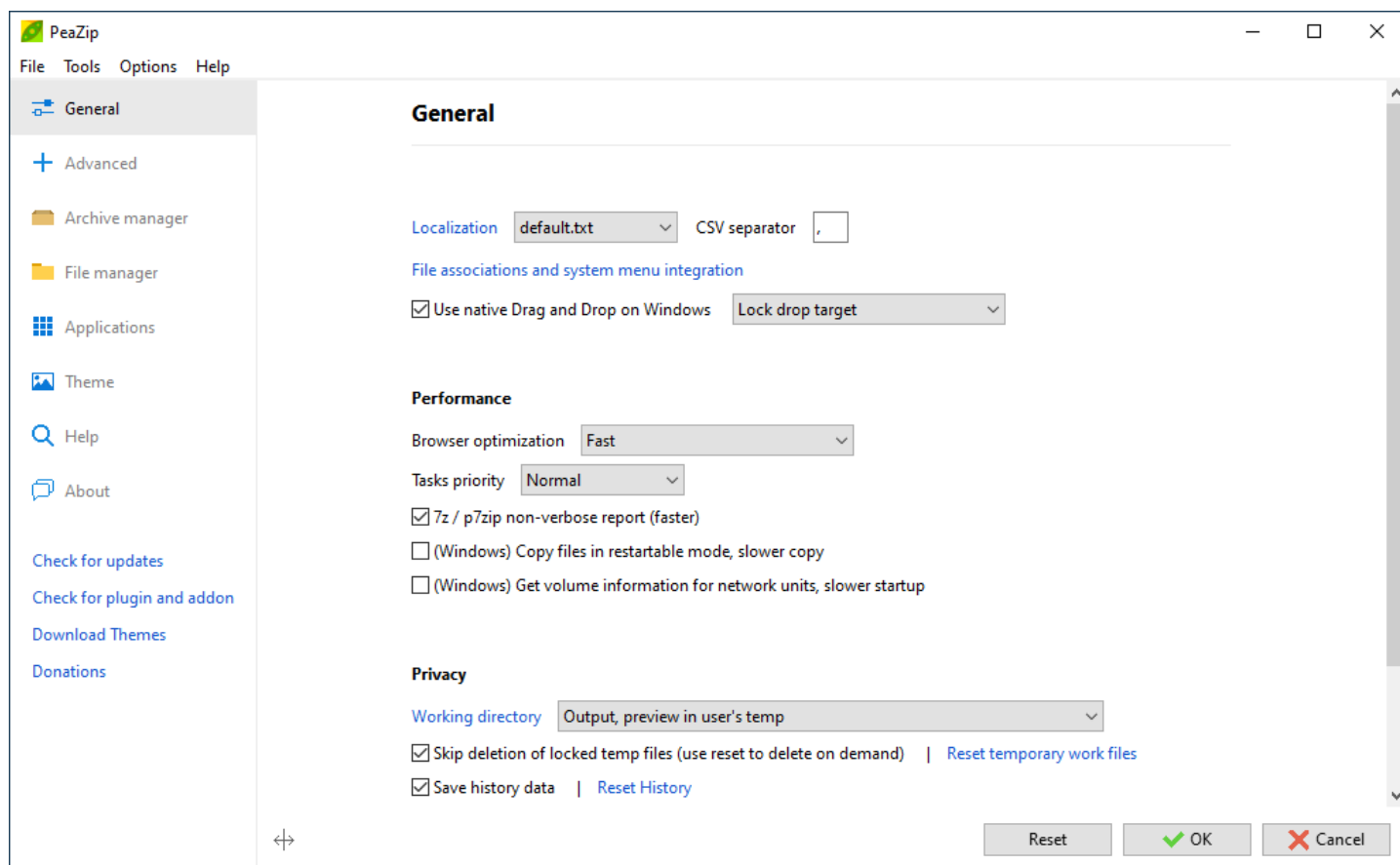
Image 28: Settings, with Help and About sections

General

Localization change application's language (and system context menu items languages in Windows installable version) and restarts PeaZip. Language can be chosen from the dropdown menu, or picked from a file selection dialog clicking on the "Language" link on the left of the dropdown menu.

It is also possible to set the **CSV separator** used by PeaZip when saving some reports as .csv files: "," is the international standard, while ";" is common in Latin countries.

Check for Translations button links to online Translations page, where newer translations can be downloaded, and containing more information about hoe to localize PeaZip.

System integration link replicate the entry with same name featured in main menu > Options.

System integration option is available also for PeaZip Portable and allows to integrate the software in the system if desired.
That creates system context menu entries for PeaZip in the registry and, technically, the application cannot be longer defined as "portable" in this way. In this case PeaZip Portable needs to be copied to the default installation directory Program files\PeaZip, or registry entries must be modified manually (which is generally not recommended) to point to actual PeaZip path.

Otherwise the simplest way to <u>integrate PeaZip Portable in the system</u> *is to associate it with the file types it will manage (e.g. 7Z, RAR, ZIP), using "Open with" from Windows rightclick context menu, choosing "Select other apps" and then selecting peazip.exe from the application's path.*
(PeaZip)\res\share\batch\SendTo\ path contains sample SendTo menu items which can be customized to be used also with PeaZip Portable, to make common actions available as links or in SendTo menu, without recurring to registry.
To remove system integration it is possible to run it and chose "No system integration" in the first installation screen, or "none of the above group" for files, context menu and SendTo menu, or else running unins.exe in PeaZip's folder (generally unins001.exe) for complete removal of any integration with the system.*

Use native drag and drop on Windows (default) if unchecked uses custom drag and drop handler. Both options shares same security, speed and stability advantages on extraction from archives, discussed in Drag and Drop chapter. Native drag and drop has the additional advantage of automatically remove unneeded levels of directory nesting on extraction, while Custom drag and drop shows its own overlay information window about items being handled by the operation.
Native drag and drop enables option for locking and/or hiding target window during drag&drop operations

"Performances" group collects settings with significant impact on program's performances:

Archive browsing optimization
- "Always pre-parse archives" pre-parse full archive content running a full list task, only for archive types supported through 7z/p7zip backend. This allows to warn user in advance of possible issues, independently from the content of the archive's TOC.
- "Very slow" to "Normal" pre-parse only archives containing less than a certain number of items (files, folders) in order to reduce CPU time and memory usage of the pre-parsing stage, improving app's performances: 1,0 million items (very slow, large archives) to 256 K (normal, small archives).
- "Fast" applies the time-consuming pre-parse step only if errors are auto detected in the listing process, with same limitation previously reported for small archives.
- "Do not pre-parse archives" open without parsing content in advance; best speed but some archives may not display properly.
Pre-parsing is always applied if errors are detected opening an encrypted archive.
A progress bar is shown at the bottom of the status bar during pre-parse operation, if the status bar itself is set visible.

Tasks priority globally sets priority for archiving and extraction processes launched by PeaZip, from idle to real time.

7z / p7zip non verbose report (faster), checked by default, uses a less verbose logging mode for backend 7z/p7zip tasks, which usually results in better performances

(Windows) **Copy files in restartable mode** (off by default) when activated, on Windows, copies files with Robocopy in restartable mode, which is slower but can be useful on networks to automatically retry and continue the operation in case of errors.

(Windows) **Get volume information for network units** (off by default) when activated checks for size, free space and other information about network mapped units. If the network is slow or a mapped unit gets disconnected, the operation may require many seconds, so if a slowdown is detected this option is automatically unchecked, but is always possible to change the value manually as desired.

"Privacy and security" group contains setting relevant to security and managing temporary work data:

Verify updates, binaries, Plugins… link points to the About tab (see below) collecting information and tools about application's binaries update and integrity status

Verify documentation, Issue Tracker… link points to the Help tab (see below) collecting documentation about the application and known issue. Notably, the Issue Tracker page also provide access to CVE database entries related to PeaZip.

Working directory set the temporary working directory for archive creation, update (add/delete), edit (modify archived files on the fly), and preview operations
- Output: improves **performances**, avoiding to copy data from system's temp to intended output destination directory – which may be on a different unit, doubling the write to disk operations. Also, this is safer under some security models as no data is decompressed / decrypted other than on the intended media.

- Output, system temp for preview (default)
- System temp: temporary files are written to system temp path, as conventionally done by most software. This option improves **stability** as it avoids possible issues in case connection with output path is lost or re-set, or if output path is periodically synced. It is safer under different security models, as it is easier to identify temp work files and prevent them to get copied/synced from system or third party software.
- Custom: allow to specify a dedicated path for work data – in example, work path can be directed to a ramdisk in order to be sure the temporary data can be easily cleared even after an event (application or system crash, files locked by third party software) not allowing PeaZip to take care of its own temp files.
- None: do not write temp files except when strictly needed by actions requiring it, in this case use system temp.
- None (disable actions requiring temp files): more restrictive than previous option, always disables use of temporary files which results in also disabling actions requiring temp files: preview, edit archived files, drag&drop extraction and the interactive extract then move mode (Extract then move option in extraction screen will be greyed).

EXCEPTIONS:
- extraction stage of archive conversion always works in output path; files are not hidden, even if they are meant to be usually deleted by default after conversion completion
- Windows Drag and Drop extraction uses working directory, but if it is set to use output path it will roll back using system temp if the output path cannot be determined during on drop event
- Unchecking "Use native Windows Drag and Drop" in this page result in using PeaZip's custom drag and drop implementation, which has some limitations compared to Windows' one but does not need to rely on temporary files, so it can be employed with Working directory set to last value None (no exceptions).

Skip deletion of locked temp files does not interactively ask for user intervention if a file is locked and cannot be deleted, in example when closing the application while a file in archive is being previewed.
- **Reset temporary work files** forces deletion of temporary work files and directories created by the application. If it is not possible to delete the folder, it will be opened for manual inspection. Closing other active instances of the program from Task Manager (or restarting the system) is recommended if temporary files cannot be removed otherwise. PeaZip takes care automatically of its temporary files, and under normal circumstances this reset switch can be ignored.

Save history data (default on) keeps track of recently created archives, visited paths and archives, and search terms. Disabling the option history will not be tracked, and menu of recent files/folders and recent searches will be not available
- **Reset History** delete records of latest paths and visited or created archives

Reset Applications delete configuration of custom editors/players/scripts/websites etc that can be associated to override default system's associations in "Open with…" tab, and parse the system to build a list of default application for "Open with…"

Reset Bookmarks delete bookmarked paths, archives, search filters…

Set/change PM master password (highly recommended, if Password Manager feature is used) define the password needed to access to the Password Manager, containing the list of passwords saved by the user
Reset Password Manager delete the list of saved passwords and re-sets the PM master password

Privacy policy dropdown menu is meant to help users working with sensible data to not forget setting a password for operating on the data (either for archiving or extracting it); the directive can be hardcoded to mode 1 or 2 at compile time, disabling and overriding this setting
- default value don't ask user to set a password at startup of the application, which works as usual
- "Ask to set password at startup" prompts the password / keyfile dialog when the application starts, so any following tasks (read or write archive) will be performed using that password (unless changed); it is simply a reminder and can be dismissed closing the password prompt
- same as previous, mandatory, password dialog cannot be skipped

Advanced

"Text encoding" group contains character encoding related options.

ZIP filenames encoding allows to set the code page used to encode filenames in .zip/.zipx archives: local code page then UTF-8 for symbols not in local code page (default), force UTF-8 for all non-ASCII symbols, force local code page, set a custom code page – useful to open zip archives built employing with a different code page, if filenames cannot be interpreted correctly with other options.
This option is also available from main menu, file manager context menu, and archiving / extraction context menus in form of a list of known codepages.

Please note that selecting an unsupported custom code page results in browsing, test, extraction, and archiving tasks always ending in error, most likely a memory allocation error.
If that happens change the custom code page or reset the application to return to the default settings.
In any case tasks can be saved as scripts and run on other machines if the current host does not support the desired code page.

7z / p7zip CONSOLE character encoding allows to set the code page to be used in command passed to 7z/pzip backend, which supports most of the common archive formats (default UTF-8).

Archive browser interface's character encoding option (default on): if flagged, displays extended characters in archived object's names as UTF-8 text, otherwise replaces unrecognized characters with "?" jolly character. This may improve commands syntax if, for any reason (i.e. limitations due to guest OS, archiving software creating the archive being read, or archive format limitations) the code page used when creating the archive cannot be successfully translated on the current machine.

 "Path" group contains links to relevant application's paths

Configuration contains configuration conf.txt, last known good configuration conf-lastgood.txt, bookmarks bookmarks.txt and custom applications list for "Open with" menu custedit.txt, and the persistent randomness collector file rnd

PeaZip's folder main folder of the program

User's SendTo menu folder (Windows only) help in creating, editing and removing SendTo menu links, and to copy those entries from one user to another, or from reference examples in SendTo.zip file in program's directory.

Desktop points to the path defined as user's desktop; "Custom" allows to set an arbitrary path to be used as desktop for the application, if preferred.

Start from set the directory PeaZip file browser will start in: last visited (default), computer root, desktop, home, custom – let the user specify the directory where the application will always start from.

"Backend binaries" group allow to choose the way the backend applications behaves

Use binaries in system paths allows to use binaries in system's paths e.g. as defined by $PATH, if available.
"No" (default) uses binaries provided by PeaZip package; 7"z / p7zip" uses only system 7z binary; "All" assumes all backend are available in system paths.
Please note that in order this feature to work the system need to have installed suitable packages that are syntax- and features-compatible with PeaZip's requirements.

Maximum argument size (non-Windows systems only) sets the maximum total size limit for the command lines generated by the application; can be customized from 32KB to 2MB, with 128KB being the default value

7z / p7zip alias allows to define an alternative name for the 7z / p7zip binary, e.g. 7za, 7zz, 7zs etc.

7z / p7zip syntax level, default Last, allows to limit 7z syntax to 21.07 release (avoiding switches introduced later), in order to improve portability of the app to platform not having last 7z build available, or when it is preferred to adhere to an older syntax to maintain compatibility with legacy systems

7z / p7zip UNRAR5, checked by default, uses 7z/p7zip routines for extraction of RAR5 format (even if UNRAR5 Plugin is installed); uncheck if you want to use RarLab-based UNRAR5 Plugin as alternative RAR5 extraction engine. Note this setting is overridden by "Free Software compliance" directive (on the bottom of this group) set to > 0.

Extrac32 (Windows only) uses system's integrated Extract32 to unpack CAB files; also this setting is overridden by "Free Software compliance" directive.

Show which options are volatile / context dependent (default off) displays in italic font the options which are not permanently set, as the application needs to toggle them automatically - meant for clarity and learning purpose.

User interface type sets is backend binaries are used as CLI applications, or directed through a graphic wrapper
- Console: native console interface, giving detailed and real time progress indication;
- GUI, default: use a graphical wrapper with pipes, giving a very detailed task log and allowing to pause, resume and change priority of the task;
- GUI + Console: like the previous, but also showing native console interface: gives a responsive UI and at the same time plenty details and real time progress indication.

Conditions when **PeaLauncher task's window stays open to show task report**, blocking further tasks

- Always stop to inspect task report (useful for debugging only, to get better insight about tasks)
- Stop to inspect report for error, list, test, auto-test: stops the sequence of tasks and keeps PeaLauncher window open only in case of errors, or for each single test, info and list tasks requiring the user to read the task's report
- Stop to inspect report for error, list, test (default), works as the previous setting except for auto-testing archives after creation; in this case the batch will only stop in case of errors (consistently how batch archive creation usually works) and not at each report (the way explicit archive testing works).
- Stop to inspect report for error, list: works as previous settings but doesn't keep open the report window for test operations unless an error is detected, this setting is **recommended to run test over a large batch of archives**, while previous settings is recommended is user prefers to read each single report.
- Never stop to inspect task report: guarantee to proceed with next task up to sequence completion, regardless the kind and the results of the tasks

Free Software compliance policy help end users, administrators, and distributors to enforce usage of free-as-in-freedom software, allowing to prevent any call to non-free software under OSI definition (which can then be safely removed from PeaZip package); to further enforce it, the directive can be hardcoded to mode 1 or 2 at compile time, disabling and overriding this setting

- "permissive" (default) allows any available backend, allows to handle any supported archive format
- "flexible" allows only Free Software components (royalty-free UNACE and UNRAR5 pluging will be disabled), handling any archive formats they support
- "strict" allows only Free Software components, allows only to handle open archive formats (not encumbered by patents / copyright either for writing or reading)

Archive manager

This screen collects archive browsing, extraction, and creation options which are not likely to be changed often - which are meant to be set once for all depending on user's preferences or needs, rather than depend on the task or on the chosen format.

Set password / keyfile options opens password prompt to set password / keyfile related options, see Set password chapter.

Auto extract TAR archive from compressed TAR.* files / EXTRACTION (on by default) extracts the content of a single TAR archive named as its compressed container, and deletes the intermediate TAR archive, resulting in atomic extraction of compressed tar file. Supported for all compression formats.

Auto extract TAR archive from compressed TAR.* files / CONVERSION (off by default) works as the previous switch, but for extraction step of archive conversion procedure. Supported for all compression formats.

Auto open TAR archive inside compressed TAR.* files (on by default) allows to directly browse compressed tar archives as it was a single atomic operation for the user: the tar archive is automatically decompressed in a temporary path (exactly as during previews of compressed files) and then opened in a separate instance of PeaZip. Supported for all compression formats except PEA.

Auto update modified files in archives (default on) asks for updating files that are changed while being previewed, for common archive file types handled through 7z/p7zip backend (such as 7z, tar, zip…). From PeaZip context menu, More, Force update of modified files the user can try to manually force this action.

Browse non-canonical archive types (default on) on double-click in PeaZip file manager opens for browsing files such as containers, installers, disk images, and other file types derivate from supported archive formats; unchecking the option such files are opened with associated application

Edit non-canonical archive types (default off) allows to save edits to file formats derivate from archive types, not recommended as such formats may contain extra data fields that are out of the scope of an archiving software

Load layout at startup (default off) allows to load a saved layout - a list of input items and inclusion / exclusion filters - when starting the application

Try to open archives containing errors (default on) allows to browse archives even if errors are detected, can be useful to handle bogus or corrupted archives

Default action on input configure the action to perform when a file associated with PeaZip is double-clicked from the system (or single or group of associated files opened with Open from context menu), or otherwise a single input file is sent to PeaZip. Possible actions are:
- Open (default),

- Open as archive, forcing opening the file as archive
- Extract… shows full extraction screen, where it is possible to set password and options before confirming extraction
- Extract all here, which immediately starts extraction with default options
- Extract here (smart new folder).

Tree defines how to show archive treeview in Navigation panel on the left of the file browser
- Sync archive tree, keep visited nodes (default)
- Do not sync archive tree, treeview is not automatically updated, as in older versions of the application
- Sync archive tree, simple, collapsing all nodes except for the one currently being browsed

The "After archiving / extraction" group displays action to be taken after completion of tasks.

Close PeaZip when task completes (default off) option schedule closing the application after current operation is completed – the application will anyway remain open in case it is needed to display error messages.

Do not ask confirmation for delete after archiving / extraction (default off).
If deletion operation is set after archiving / extract all operation, this option allow to skip confirmation request.
Use with caution: deletion is not performed if errors are detected in the process, but there are difficult to predict cases where user's supervision is recommendable over automatic checking.

Open output path when task completes option (default off) automatically opens the output directory where the archive was created in at the end of the process.

Test archives after creation, if supported by the format (default off) a test operation on new archives is scheduled after creation, if supported by the format. Unlike on-demand test (as with Test button) automated test result does not stays on screen except in case of errors, in order to not block a queue of multiple archive creation processes - this behaviour can be changed from Settings, Advanced tab, "Conditions when PeaLauncher task's window stays open"

The "Archive" group displays options related to archive creation

Default format (default last used) sets the default archive/compression type.

Archive conversion scripts allow user's interaction (default on) allows to interactively customize conversion tasks saved as scripts when launched, otherwise the task will run as batch requiring no interaction and applying current application's settings

Enumerate folder options (default on) checks number and size of files and directories in the archive layout, for a better estimate of input size and possible output size (see "Potential compression" information in status bar). The operation can be slow if a large number of files and directories is involved.

Store all timestamps (default on) saves creation and last access timestamps for items in archives supporting that feature (7Z, RAR, ZIP/ZIPX), if the option is unchecked only modified timestamp will be saved.

Switch to zipx extension for non-Deflate zip archives will automatically use zipx extensions for zip archives if a different compression algorithm than Deflate is used (as done by WinZip), in order to mark them as incompatible with legacy archive managers

Set advanced archiving options set archiving and conversion -specific options from Archiving screen

The "Extract" group contains option related to archive extraction.

Auto close after extraction if no browsing actions took place (default off), useful in example if a downloaded file was opened for quick inspection before extraction

Always ignore paths… directives (default all off), extracts file(s) to output not re-creating the directory structure as is in the archive. Please note those switches does not apply when extracting directories, and are overridden by **extract then move** option (set in extraction screen).
- Extract and open with… operations
- Extract displayed objects (objects shown in current browsing or search filter)
- Extract selected objects

Enable "Extract and open with" submenu (default off) shows the submenu in application's context menu.

"Extract and open with" submenu contains "open with" (PeaZip, associated application, custom…) functions concatenated after extraction likewise functions in "Preview with" menu. Unlike for preview functions, the extraction does not take place into a temporary path, so it should be used carefully.

Smart new folder (default on) modifier for "Extract to new folder" option, let the software decide if creating an extra new folder for extraction, works as "Extract here (smart)" context menu entry.

Smart extraction mechanism guarantee a single item is created from extraction, to avoid "tarbomb"-like effect in which extraction of archive results in destination path being flooded by unexpected files and folders.

At the same time, unlike plain "Extract to new folder", smart extraction guarantee no extra directory nesting level is added, so if extraction results in a single file or folder, the content is moved outside the extra new folder, which is then removed (if moving the data fails, the content is kept in the new folder).

Skip testing for encryption (default off) will suppress password requests encountering encrypted archives.

Extract everything for special file types allows (for 7Z-supported archive types) to optionally extract the entire archive (or current path) to preview special filetypes which usually needs external data to properly run, in example .html, .bat, and .exe files
- First dropdown sets when to apply this rule: Never, Ask (interactive), Always (default)
- Second dropdown sets how to apply the rule: Extract entire archive (default) or Extract displayed items only (items in current path, or matching current search filters) which may be enough in some cases and can be faster than extracting the entire archive
- Formats to be treated as special cases can be edited in the text field, space separated extensions with or without dot.

Set advanced extraction options set extraction-.specific options from Extraction screen

File manager

"Browser" group collects options for the file browser.

File size sets the multiplier for file size: binary (1024) for IEC KiB standard, decimal (1000) for IEC kB, none for exact byte size.

Checksum/hash files sets the algorithm used to check, on demand, files in the file browser (by default CRC32).

Use quick duplicate finder makes the application to use a quick approximate function (aforementioned checksum or hash algorithm over small portions of the file, taken in various places) to find duplicate files rather than running the algorithm on the full file content.

"File tools" group collects options for the separate file analysis & management utility.

Secure delete sets the level of secure deletion procedure; fastest overwrite input file(s) with random data, very slow overwrites with all 0, all 1, and then 3 times with random data.

Checksum/hash files options allows to **select algorithm(s) to be used for file checksum/hash analysis**, define the **list of preferred algorithms** (the list is enabled selecting "Preferred algorithms" in the algorithm selection dropdown menu), and **how to display the result** (hexadecimal, hex LSB, Base64).

Applications

This screen allows to organize **custom applications** (or scripts, or commands, or websites) to be used to open files overriding system's file associations, saved to a separate configuration file "custedit.txt".

Applications can be sorted dragging up or down the application's number in the list; double click on cells to edit or remove applications and descriptions, rightclick for more options.

"Reset Applications" button wipe current application lists (basic and advanced) and try to find on the system (and configure for use in PeaZip) some common utilities and anti-virus scanners.

- "Basic" group (up to 16 applications or scripts) allows editing custom application with ease, selecting or typing the application or command to be used to open a file (or preview a file inside read-supported archives). It is also possible to enter parameters after command or application name in the same string; after the "Executable or command" string it will be passed a blank space and the input file name.

Note: in this group can be defined also websites/webservices to pass the filename to as parameter; in those cases (detected by PeaZip if the name of the app starts for www. or http:// or https://) only the filename is passed, not the full qualified name (path+filename) as for local applications.

- "Advanced" group allows instead a bit more complex syntax, providing a string to be entered before, and one after, the input file name (and it's up to the user decide if spaces between strings and file name are needed). In the last column of the table it is possible to enter the syntax for the application when no file is passed as parameter, as this case may require a different syntax, or even invoking a different executable.

Note: by default antivirus / antimalware scanners are defined in "Advanced edit" set, because some of them requires a quite complex syntax, but it is only a convention. Syntax's examples of pre-set applications can be used as model to start customizing entries in both sets; clicking an entry in "Advanced edit" set will show a line displaying complete command entered, with the pseudocode "%f" representing the input file name position in the command string.

Theme

Themes set appearance and icons for PeaZip application, and for Pea executable.
Theming values can be edited from this tab, and saved in the current configuration, to override correspondent theme settings - which can be restored clicking on "Reset" link for each variable.

Check themes button shows the online Themes page, where PeaZip themes can be downloaded, and containing more information about how to use, edit, and publish custom themes.

Theme drop down menu allows to change application's icons (including archives and folder icons) and look& feel to better blend with target system, choosing one of the pre-set themes which comes packed with PeaZip, or a custom theme (theme package has extension ".theme.7z").
Custom themes can be downloaded from "Check for Themes" button on the top of the panel, pointing to Add-Ons page on PeaZip website.
Clicking on the Theme label link opens the folder containing theme icons, which can be edited as plain graphic files – read more in this chapter about how customizing and saving themes.

Application accent sets a custom color for highlighted graphic items in the application.
"Preset" link, on the right of Application accent color picker, allows to quickly customize (from a simple context menu) both application accent and text accent variables accordingly to classic color combinations that can be found in reference operating systems' UI, e.g. BeOS, Mint, Ubuntu, Windows…

Text accent sets the custom color for text links and can be set independently from Application accent color.

Color sets app's background color, applied to most of the visual items (forms, listviews, treeviews): by default it set equal to system's background color to match system's and other apps' appearance, but can be changed to suit user's preferences.
"Presets" link on the right of the color picker allows to quickly reduce luminosity of the main color (Low light or Dark modifiers) or add non-neutral tints (Graphite, Mocha, Plum, Senape) to main color; Reset link restores original theme color.

PeaZip uses system colors, automatically adapting to dark themes and Windows Dark Mode.

Color temperature allows to render app's background and buttons / tabs color tonalities colder or warmer, and it is zeroed by default on a neutral grey.

Contrast sets the contrast level between background and highlighted design elements.

Spacing sets the file manager's and tables vertical spacing between elements/rows, alongside zooming it can be used to improve usability for users with special needs, i.e. visual impairment, touch screens, small or distant screens

Tool bar group sets the background color for the tool bar (window, button, or accent color, in multiple shades), and optionally sets centered buttons instead of left aligned.

Address bar sets color for the address bar (window, button, or accent color, in multiple shades) except for Address field

Breadcrumb bar sets color for the field containing the breadcrumb (or textual address), it can be selected independently from the color of the parent control Address bar in order to make the field stand out or blend with the address bar at will.

Highlight tabs shows the rows (or columns) containing tabs in window, button or accent color.

Tabs style displays tabs, e.g. in archiving & extraction screens, as buttons, links, or classic tabs, each of those styles can be either centred or left aligned.

The "Advanced" link shows more theming options explained below.

Alternate grid color shows grids in dual tone

Transparency maxes the application transparent; there is a maximum level of transparency hard coded for the application to avoid to make it unusable or totally invisible.

Small icons size, by default 16 px, allows to scale the small icons up to 32 px. This setting applies to small icons only.

Zooming allows to resize app's graphics independently from the default scaling, useful to display larger icons. This setting applies to all icons and graphic elements.

The user can create new themes clicking on "**Create Theme from current settings**" link, it will be created a theme package on the desktop named name.theme.7z; themes can be distributed as independent packages.
Please note that the file readme.txt is inherited from the original theme, please edit it with relevant information and credits about the new theme if you plan to release it.

To further customise themes, or to remove unwanted themes, "**Manage Themes**" opens the path containing themes, that is in user's application data for installable versions or inside PeaZip's folder for portable versions.

PeaZip 7.x and more recent accepts as icon sources arbitrarily sized graphic saved as 32 bit or 8 bit PNG with transparency.
Icons will be scaled at runtime to be rendered fitting system's icon size (or approximate size that can be efficiently rendered), up to 10x zoom - in example a small icon can be rendered ranging from 16px to 160px.
Configuration file theme.txt can be edited as UTF-8 text.

App icon, **context menu icons**, and **associated file types icons** are not part of the theme, as they are system-dependent, and can be customised separately.
On Windows, icons are stored in (PeaZip)\res\share\icons\ path and can be edited with a suitable icon editor or replaced with custom icon files.
On Linux systems icons (in PNG format) are stored accordingly the distribution and the desktop environment policies, i.e. \opt\kde3\share\icons\ or \usr\share\icons\
Custom, user-provided icons and other resources can be found on Add-Ons page on PeaZip official website.

Help

This screen contains a localized mini-help, and links online and offline documentation, issue tracker etc.

About

This screen shows detailed information about the current version of PeaZip and binaries provided by the package.
From this page it is possible to
- Check for online updates and Plugins
- Verify current version of the application, binaries, plugins…
- Verify hash of binaries against known values, as in main menu > Tools > **Verify hash of binaries**
- Quickly access to binaries folder, to manually add, delete, and verify binaries and plugins
- Remove (if present) non-Open Source plugins (unrar5 and unace) with a single click.

HINT: to install a PeaZip plugin

1) Download the desired plugin for PeaZip from the official website
2) Open the plugin file, which is a TAR archive: the plugin TAR archive can be opened with PeaZip itself, or using any other suitable archive manager utility
3) Extract the directories provided by the plugin package in PeaZip's binaries folder (peazip)/res/bin/

This will add to the application the needed executables to manage the desired archive format(s) - please note some operating systems may require to manually set files as executables.

Binaries and plugin folder can be opened from the button on the bottom of "About" screen, and it is usually is located in
/usr/lib/peazip/res/bin/ (Linux)
/Applications/peazip.app/Content/MacOS/bin/ (macOS)
C:\Program Files\PeaZip\res\bin\ (Windows)

Advanced editing of settings

Program's configuration is stored in conf.txt file and last known good configuration (successfully loading) is saved in conf-lastgood.txt and it is automatically used to restore previously working state if conf.txt fails to load.
If even last working configuration fails, PeaZip will load pre-set values and save a new standard configuration file.

Configuration is saved by default in /res/conf folder in application's path An alternative location, either as absolute or relative path, can be set in second line of altconf.txt file in the same folder: "**same**" string in altconf.txt specify the data has to be saved in program's res path (best for portable packages) and "**appdata**" specify the data has to be saved to %appdata%\PeaZip\ path in user profile (Windows) or /.PeaZip/ path in user's home on Unix systems, to guarantee write access to data from current user's profile and allow different users to store different and private profiles for PeaZip.

In this way only the invariant data (including binaries, graphic and default themes, language files etc), which needs to be accessed only for reading during normal program's usage, is stored in program's path, that could be even set as read-only.

All variable data files (conf.txt, rnd, custedit.txt and bookmarks.txt), which can need to be accessed for writing during normal program's usage, will be stored in res folder (by default) or any other path specified in altconf.txt.

If the altconf.txt file is deleted, it is needed to re-create, or copy it from PeaZip or PeaZip Portable packages, in order to determine if the software will work as installable or portable version – in the meantime PeaZip will load pre-set values and run in volatile mode without saving configuration (which does not allow user preferences to be remembered).

Supported formats

See also Learn more page on PeaZip official website.

Main families are:

Through **Pea** executable (LGPL, Windows and Linux)
- Full support
  - PEA: security focused, flexible integrity check and optional two factor authentication with passphrase and keyfile (AES, Serpent, Twofish, 128 and 256 bit, used in EAX mode authenticated encryption), fast compression comparable with Zip/Gzip, native multi volume spanning.
  - Split: compatible volume spanning (file split/join) function with optional integrity check

Through Igor Pavlov's **7z** (LGPL, Windows) and Myspace's POSIX 7z (LGPL, Linux)
- Full support
  - 7z, 7z sfx: feature-rich archiving format, strong AES encryption, awesome compression ratio and optionally auto extracting archives (sfx, Win32 executables)
  - Bzip2 (BZ, BZ2, TBZ, TBZ2): single file compressor, adequate speed and good compression ratio
  - Gzip (GZ, TGZ): fast single file compressor, adequate compression ratio
  - TAR: mainstream archiving and backup format for Unix platforms, not providing native compression
  - WIM (and SWM) Windows image format, Microsoft's disk image format
  - XZ: single file compressor, based on LZMA2 algorithm, providing excellent compression ratio
  - ZIP: mainstream archiving and compression format for Windows platform; support covers also Deflate64, LZMA and PPMd compressed archives, and AES-encrypted archives
- Browse/extract support
  - ARJ, LHA, LZH: popular archiving format on DOS and early Windows platform
  - CPIO, Z, TAZ, TZ: archive/compression formats for Unix platforms
  - LZMA: single file compressed with LZMA algorithm, introduced with 7z format
  - RAR/RAR5: popular archiving and compression format, with advanced encryption and error recovery features
    - 7z/p7zip frontend, since release 15.06, manages both legacy rar files (up to RARv2.9/RAR4 format) and new RAR5 files; users can chose if handling RAR5 using 7z/p7zip (default) or RarLab's royalty-free "unrar" backend (only available as separate plugin due the non-OSI compatible nature) checking "7z/p7zip RAR5" flag in General Options.
  - Various archive types based on ZIP or its modifications: Java archives (JAR, EAR, WAR), PAK, PK3, PK4 (modified zip archives, used to store data by some games like Quake3, Quake4, Doom3), SMZIP, UP3 (U3 portable application's package format), XPI (Mozilla package format for addons of Firefox, Thunderbird etc), and many more
  - MBR, FAT and NTFS filesystems
  - Various disk images: ISO (standard disk image format), IMA, IMG, IMZ, UDF
  - Various executable file types: NSIS (Open Source Windows installer format), some Windows executables, MSI Microsoft's proprietary installer format for Windows
  - Various (non-archive) compressed file formats, like
    - CAB compressed archive format
    - Compound (used in some of MS Office formats: DOC, XLS, PPT)
    - OpenOffice file types: container files for text, database, image and multimedia data: ods, ots, odb, odf, odg, otg, odp, otp, odt, ott, oth, odm, oxt.
    - Other containers as Gnumeric .gnm files, CHM, CHW, HXS compressed help files, SWF, FLV…
  - Various Linux installers: DEB (Debian-based), PET/PUP (Puppy Linux), RPM (Redhat-based), SLP (Stampede Linux)
  - Various Macintosh formats: DMG/HFS package/disk image format
  - LZ (Linux only due to p7zip fork)

Through **PAQ/LPAQ/ZPAQ**, Matt Mahoney et al. (GPL, Windows and Linux)
- Full support for PAQ8O, ZPAQ, LPAQ8. PAQ is an experimental compressor; at cost of high computing time and memory usage provides best known compression ratio for most data structures. LPAQ and ZPAQ are relatively lighter and faster but provide lower compression.

Through Ilia Muraviev's **BCM** (Public Domain)
- Full support for BCM: high performance Burrows-Wheeler Transform -based compressor

Through **Strip** (GNU binutils) and **UPX** (GPL Markus F.X.J. Oberhumer, László Molnár and John F. Reiser)
- Strip reduces size of executables stripping symbols, and UPX apply compression: this allows cutting the size of binaries of different types (exe, elf, etc…), i.e. for distributing smaller packages.

Through Bulat Ziganshin's **FreeARC** (GPL)
- Full support of ARC/WRC, FreeArc's sfx: experimental archive format, featuring efficient compression (high ratio and good speed), and advanced features like strong encryption and recovery records

Through **Brotli** (Google / Jyrki Alakuijala, Eugene Kliuchnikov, Robert Obryk, Zoltán Szabadka, Lode Vandevenne, MIT License)
- Full support of BR format featuring fast data compression and extraction

Through **Zstd** (Facebook / Yann Collet, Przemysław Skibiński, dual license BSD License and GPLv2)
- Full support of Zstandard ZST / TZST format featuring fast data compression and extraction

[external] If **WinRar** is installed, switching to Custom format in archive creation screen automatically configures PeaZip to send command to Rar.exe in order to create RAR archives.

[external] Windows Makecab can be set in custom format (pre-configured in custom executable field dropdown menu) to create CAB packages

[external] Windows Extrac32 can be set as alternative engine for extracting CAB packages, from Options > Settings > Advanced, Backend group.

PeaZip UnRar5 Plugin (SEPARATE PLUGIN)

No longer needed since PeaZip 5.9.0 release. Provided as alternative RAR5 extraction engine, and for older versions support. Contains Eugene Roshal (win.rar GmbH) RarLab's unrar: royalty-free, source is available but unrar restriction (limitation to freedom of use of the code, disallowing to use it to create a rar compressor) makes it non-OSI compatible, featured as alternative to integrated RAR5 support, currently only needed on Linux as p7zip does not jet support rar5 format
- Browse/extract support for RAR files detected using RAR5/v5.0 revision structure; due the different syntax between RarLab's unrar and 7z/p7zip's unrar, following differences apply when handling RAR5 format through this engine:
  - o recursivity with advanced filters is always enabled
  - o renaming options always rename the file being extracted

PeaZip UnAce Plugin (SEPARATE PLUGIN)

Through UNACEV2.DLL 2.6.0.0 (Windows) and UNACE (Linux): royalty-free, proprietary, Marcel Lemke, ACE Compression Software Please note that after unacev2.dll vulnerability discovery in 2019 it is recommended to stop using ACE format and convert existing archives to other formats, and to remove the plugin as soon as it is no longer needed.
- Browse/extract support for ACE: popular compression format, used mainly on Windows systems

PeaZip Additional Formats Plugin (SEPARATE PLUGIN)

Through Ilia Muraviev's **QUAD** (GPL) and **BALZ** (Public Domain)
- Full support for QUAD (high performance ROLZ-based compressor which features high compression ratio and fast decompression) and BALZ (similar design, enhances overall performances compared to QUAD)

Through **PAQ/LPAQ/ZPAQ**, Matt Mahoney et al. (GPL, Windows and Linux)
- Read only support for older PAQ8 formats F/JD/L and for older LPAQ1 and LPAQ5: lighter and faster version of PAQ, at the cost of some compression (single file compressor)

PeaUtils

PeaUtils is an alternative GUI, provided by pea executable, focused only on launching various integrated file utilities features.

Function context menu provide a choice between supported utilities, i.e. apply single or multiple checksum and hash algorithms, secure delete files, secure deletion or zero deletion of free space on a drive (Windows only), list files (can be saved to file), display environment variables.

Input file(s) can be selected via a standard file selection dialog, or can be simply dragged on the application; a link to system search is provided, so it can be started to find the file to be dragged on the application.

PeaUtils can be accessed from Tools menu in main application's menu, or starting without parameters the executable pea in PeaZip's path, or with command line "pea peautils [n]" where n is a number from 0 to 11, an optional parameter pointing the GUI to start with a specific function (the number is the order of the function in the dropdown list).
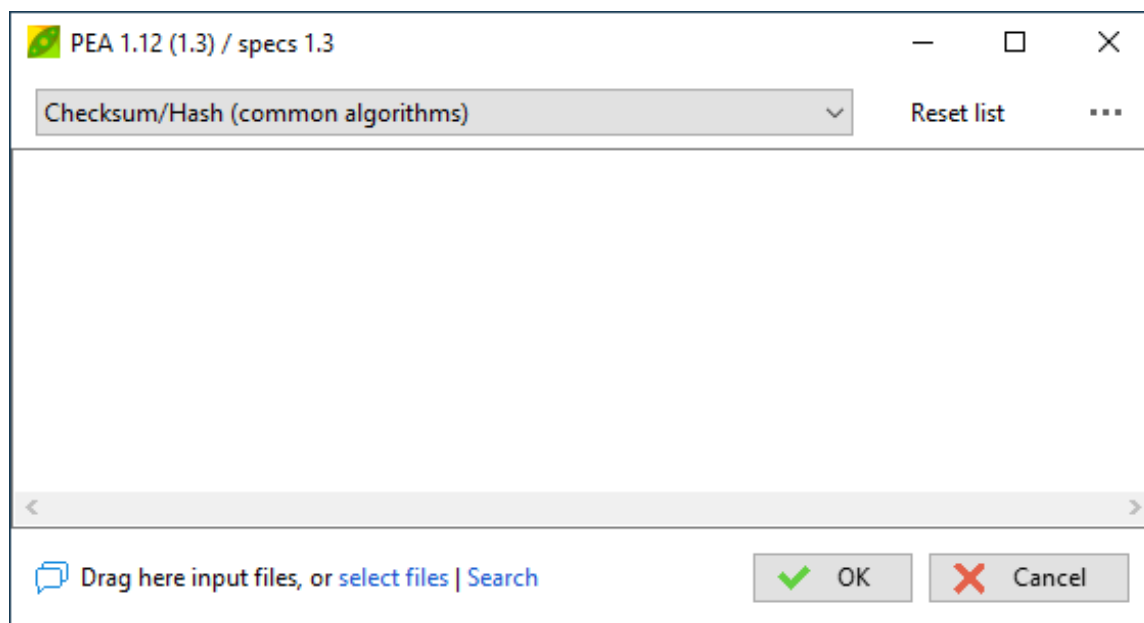

Image 29: PeaUtils

Customisation and scripting, command line syntax examples

PeaZip portable doesn't need installation and doesn't modify the host system, however program's most used functions can be integrated, under Windows, in SendTo and context menu and, under Linux, in freedesktop-compliant desktop environments.

In **(peazip)/res/share/batch** directory are featured script examples and SendTo links, and in freedesktop_integration subfolder are provided .desktop files (Gnome, KDE, Xfce, Cinnamon…) and Nautilus scripts (Gnome) for integration with Linux desktop environments.

In the same way it is possible to extend the integration automatically provided by installable packages, creating quick links in system's menus or in scripts for most of the program's internal functions.

PeaZip acts as a frontend for many Open Source archiving executables, each with its own syntax developed and maintained by respective third parts Authors, see "Supported formats" chapter for a comprehensive list of technologies employed in PeaZip.
For the syntax for each backend executable please refer to the original Authors and the original documentation (i.e. in 7-Zip's documentation and websites), however one of the scopes of PeaZip is offering a simple and homogeneous command line interface to mask the complexity and the differences in command line syntax of underlying applications.

While through PeaZip's GUI it is possible to use underlying applications with great granularity (and save command lines for any further use from Console tab in extraction and archiving screens), **PeaZip itself is made accessible through command line** to offer an easy access to most common functions.

The switches can be used in **scripts**, and for system integration: **links**, **SendTo items**, Linux/BSD **.destop files**, macOS **Automator scripts**… special switches, specified later, are meant for use in Windows registry entries, which are subject to unique limitations.

The full list of strings accepted as quick link to PeaZip functions when passed as first parameter is:

    -add2archive: add to a new archive and open PeaZip GUI to set archive's type and options;
    -add2archive-add as -add2archive, but add to existing archive (if found)
    -add2archive-update as previous, but update older files only

    -add2pea: add to a new .pea archive;
    -add2crypt: add to a new encrypted .pea archive;
    -add27z: add to a new .7z archive;
    -add27zmail: add to a new .7z archive and attach it to a mail (requires compatible mail client) *
    -add2separate7z: add each input to a separate new .7z archive;
    -add2sfx7z: add to a new self extracting 7z archive (.exe);
    -add2sfx7zmail: add to a new self extracting archive and attach it to a mail (requires compatible mail client) *
    -add2zip: add to a new .zip archive;
    -add2zipmail: add to a new .zip archive and attach it to a mail (requires compatible mail client) *
    -add2separatezip: add each input to a separate new .zip archive;

    -add2arc -add2sfxarc -add2rar -add2brotli -add2bzip2 -add2gzip -add2tar -add2wim -add2xz -add2zstd -add2zpaq
    Directly add files and folders to archive in the specified format, with latest options used for that format.
    Please note -add2zpaq switch sets last used *PAQ format (default is zpaq); -add2rar requires WinRar rar.exe or equivalent to be available in the host system.
    TAR is automatically applied when folders and/or multiple files are sent to formats which supports single file compression (Brotli, BZip2, Gzip, XZ, Zstd).
    If no input file or folder is provided after the switch, the archive creation screen remains open for input.
    Same switches in -add2multi* form can be used for integration in Windows registry.

    -add2convert: add items to archive conversion task, see Create / convert archives chapter for more details about archive conversion procedure; this switch will prompt full app's GUI to allow to refine the task further
    -add2convimm: as the previous switch, but will immediately start archive conversion with current app's settings

    -add2preset N, add items to archive as specified in Nth preset. Compression presets are numbered from 0 to 15, are fully customizable, and can be found in (peazip)/res/share/presets directory, see "Create / convert archives" chapter for more information about the topic.
    -add2custom (full qualified path of custom compression settings file) add items to archive as specified in the specified custom compression setting file.
    Used in combination with "Immediate execution" option, -add2preset and -add2custom switches allows to immediately start compression without requiring confirmation, otherwise they allows to further customize and fine tune the task.

-add2wipe: securely delete selected file(s);

-add2compare: byte to byte compare two files;

-setcomment commentstring archive1..n set the text in commentstring as comment for specified archives
-setcommentf commentfile archive1..n set the text loaded from file commenfile as comment for specified archives
-removecomment archive1..n remove comment from specified archives

Comments can be set for ZIP/ZIPX files, and for RAR files if Rar.exe is featured in the system, working even on multiple archives of mixed mixed types at once; the comment can contain UTF-8 characters and up to 64 KB of text for all formats.
Comment can be set for multi-volume RAR files, but not for multi-volume ZIP/ZIPX files.

-ext2browse: open (and browse if applicable) the archive(s) in PeaZip GUI
-ext2browsepath: browse the selected folder (or its path, if a file is selected) in PeaZip

-ext2here, -ext2folder (alias -ext2smart), -ext2newfolder: extract archive(s) to its own path, "folder" (alias "smart") variant creates new folder if necessary as with "Smart new folder" option enabled, "newfolder" variant always creates a new folder named after the archive (as with "Smart new folder" option disabled)
-ext2neutral directly extracts archives in current path applying the "new folder" policy as defined in PeaZip app (non, new folder, smart new folder, or force new folder)
-ext2downloads, -ext2desktop, -ext2documents directly extract archives in Downloads, Desktop, or Documents path
-ext2bookmark(1..8) extract archives in the path of the bookmarked item number 1 to 8
Those options apply PeaZip's new folder policy, as in -ext2neutral switch.

Available options for those operations:
-i ignore "Delete original archives after extraction" directive, if set in the application
-o next parameter declares output path, otherwise assume path of first input item as base output path
-p next parameter declares password, otherwise if password is needed it is asked interactively
Those switches execute single or multiple tasks as a single sequence of queued tasks: in this way multiple batch extraction tasks can run in a single window without stealing the focus each time the next task in sequence is started

Legacy (pre 8.2 release) equivalents of extraction switches are now deprecated, and executes same functions without presenting a unified queue:
-ext2here_, -ext2folder_ (alias –ext2smart_), -ext2newfolder_
-ext2to, ext2tosmart, -ext2tofolder: extract archive(s) interactively asking for output directory;
-ext2simple, ext2simplesmart, -ext2simplefolder: allows to specify output path as first parameter;
Variant of this group of switches with "i" suffix (i.e. –ext2simplei) ignores delete after extraction option set in the application
-ext2list: list archive(s) content, to quickly look what is in the archive;
-ext2test: test archive(s) content;

-ext2main: extract archives from main applications "Archive extraction" interface (alias: -ext2full);

-ext2commandprompt: open the command prompt in the selected folder (or in its path, if a file is selected)

-ext2openpwtest requires password as paramstr(2) (always delimited) and archive's full name as paramstr(3): try to list the archive and terminate with exit code 0 if archive is not encrypted, 1 if it is encrypted, 2 if it is non readable (corrupted or with encrypted file list), -1 if archive is not found.

-peazipreset: reset PeaZip configuration, as Options > Reset button; can resolve unexpected problems with configuration, restarting the application with factory settings

mail functions require a compatible mail client, like i.e. Outlook and Outlook Express, as default mail client of the system

-add2archive, -ext2main and -ext2browse open the PeaZip GUI, to allow further user's interaction

Direct archiving functions -add2pea, -add2crypt, -add27z, -add27zmail, -add2separate7z, -add2sfx7z, -add2sfx7zmail, -add2zip, -add2zipmail, -add2separatezip
and direct extraction functions -ext2here, -ext2folder (alias -ext2smart), -ext2newfolder
can delete input files as atomic operation (asking for confirmation) if this policy is set in PeaZip delete after archiving and delete after extraction checkboxes in archiving and extraction screens respectively.

As it could not be handy to have an interactive request prompt in a batch script, an alternative version for those actions **not applying deletion policy after archiving/extraction (ignore deletion)** is available

- appending "i" letter to the –add2… group of switches (zip and 7z formats only) e.g. -add27zi, -add2zipi;
- as -i option for -ext2… group of switches.

On Windows platforms PeaZip can also send multiple input to a single instance through **registry entries** using temporary files; temp files are deleted automatically (otherwise the mechanism would not work) but in case of any kind of problem there is a reset switch in Options > Settings first tab to remove any possible leftover:

-add2multi works like -add2archive

-add2multi7z works like -add27z, but stays in the archiving interface until the user click "Ok" button; it can be modified in -add2multi7zfastest, *normal, or *ultra to directly set the compression to fastest, normal or ultra levels presets (the compression method used is the one defined by the user); *mail send archive by mail, *encrypt create encrypted 7z archive

-add2multizip works like -add2zip, but stays in the archiving interface until the user click "Ok" button; likewise the previous switch it can be modified to *fastest, *normal, or *ultra compression level; *mail send archive by mail

-add2multisfx works like -add2sfx7z, but stays in the archiving interface until the user click "Ok" button

-add2multibrotli -add2multibzip2 -add2multigzip -add2multitar -add2multiwim -add2multixz -add2multizstd -add2multizpaq directly add files and folders to archive of specified format wiuth default compression options

-ext2multi works like - ext2main

While the instance accepting files is open, it is possible to continue adding input from the same directory selecting objects and using context menu's entry, as an alternative to dragging the object to the application window that might be less handy if the desktop is crowded by windows.

Example of command line syntax for archiving: *peazip –add2zip file1 file2 directory3*
which will add specified objects (in the example file 1 and 2, and all content of directory 3) to a .zip archive, auto named after the 1st object (in this case will be named file1.zip and will be saved in the same path of file1); using –add27z instead of –add2zip will perform the same task but will result in a .7z archive (-add2pea will result in a .pea archive, -add2sfx7z will result in an self extracting executable and so on).

Example of command line syntax for extraction: *peazip -ext2here archive1*
will extract archive1 in the same path; using -ext2newfolder archive1 will be extracted to a new folder named "archive1" in the same path of archive1.

Another example for extraction: *peazip -ext2here -o destinationdirectory -i -p thepassword archive1 archive2 archive3*
will extract archive1, archive2, archive3 to destinationdorectory output path using thepassword and ignoring delete after extraction directive if set in the application.

For all extraction command it is also possible to pass directory names as input, directories will be scanned and found archives will be processed.

Translations

In "Translations" directory, there is a package named peazip-x.y.about_translations.zip containing all files and information needed for translators to localize the application.

Language files are contained in res/share/lang folder, and the instructions to translate main application language file is contained in "readme_about_translations.txt".

Instead, if you are interested in translating Windows context menu entries (for W7 and above) please refer to instructions in "readme_about_windows_context_menu.txt".

Language files are plain **UTF-8 with BOM** encoded text files which can be freely edited using any suitable text editor.

To create a new translation file you can:

1 - make a copy of default.txt (in PeaZip's path in /res/share/lang subfolder) or of any other language file if you prefer starting from a language other than English;

2 - edit lines 2 to 6 of the document to enter language name (both in English and in the original language for better readability), PeaZip's version (major.minor) the translation is aimed to, translator's and last last revisor's name and last revision date;

3 - translate the text after the "variable_name: " part (don't add, move or delete lines, don't change the "variable_name: " part);

4 - optionally, translate the mini-tutorial after "=== about text group ===" line (free editing, it is loaded and displayed "as is" as application's mini-tutorial); it is very important to improve usability of the program for non-English speaking users.

5 - name your language file: it is recommended to use the standard two letter international code to identify the country https://en.wikipedia.org/wiki/ISO_3166-1_alpha-2

As alternative tool to help translators, you can find "translations.xls" to help in creating and maintaining localizations, simply compiling column D.

The spreadsheet contains 2 pages to be compiled: "PeaZip text group", and "PeaLauncher text group".

The pages need to be completed and pasted (column E) in the language file; the "About text group" can be freely edited in the language file itself.

The spreadsheet shows variable name (column B), corresponding text string in english (column C) and a blank, yellow column (D) for typing the translated text strings.

On the right, a column E (blue) will show the "variable_name: " part assembled with the translated string: the content of this area can be copied and paste to replace the text in "=== PeaZip text group ===" and "=== PeaLauncher text group ===" sections (the spreadsheet features TWO pages, one for each of the two groups).

Lines must be pasted in the original order (it is sufficient to sort them by column F).

USE OF THER SPREADSHEET IS OPTIONAL, you can directly modify any existing language file if you prefer.

At each new version of PeaZip all bundled language files are mass-updated, with missing text lines added in English; to update a localization, it's enough to update the English text lines.

For a better result it is also recommended to check all the language file using them in the application, to see if the update is coherent with linguistic style used by the translator of the current version.

For languages spoken in different ways in different countries (i.e. English, Spanish, Portuguese...) it is recommended to fork the translation, creating i.e. en-us, pt-br etc

To contribute translations:

MAIL
Language files can be sent to giorgio.tani.software@gmail.com, to be evaluated for inclusion in future updates or publication in Translations repository
https://github.com/peazip/PeaZip-Translations

GIT
Language files and translated .reg files can be updated on Git repository so commits can be evaluated and merged in the online code base
https://github.com/peazip/PeaZip/tree/sources/peazip-sources/

All translated language files should be considered as released under GFDL, GNU Free Documentation License, as they have to be considered derivate work from the application's language file which is released under GFDL.

Notes

Creation of RAR archives
No free software application can create RAR archives, unless the operation is performed invoking WinRar binary itself, because the RAR format is proprietary, moreover, the UnRAR license explicitly disallows to reverse engineering the file format definitions implemented to build functional RAR creation software.
If WinRar is installed in the system you can use PeaZip's "Custom/RAR" type to invoke rar.exe to create / update / edit / repair RAR archives through PeaZip's GUI, which takes care to apply the correct syntax.
From Advanced tab you can disable using RAR, in this case you will be able to manually configure the custom executable and its syntax.

Extraction of RAR5 archives
Since PeaZip 5.9.0 release the non-free UnRar5 Plugin is no longer needed neither on Windows nor on Linux to extract RARv5 files, but the plugin (contains Eugene Roshal RarLab's unrar) is still provided on Add-ons page as alternative RAR5 extraction engine, and for older versions support.

ACE archives and unacev2.dll vulnerability
Supporting ACE archives requires manually download and install the separate UNACE plugin, based on WinAce unacev2.dll, which is provided for legacy support only.
PeaZip does not contain unacev2.dll out of the box, and even if the UNACE plugin is installed PeaZip will not call unacev2.dll to open ACE files disguised for other formats.
After 2019 vulnerability discovery in unacev2.dll (which is closed source and no longer maintained) it is recommended to stop using the ACE format, migrate existing legacy ACE files to better supported formats, and remove UNACE plugin directory (or delete unacev2.dll) as soon as it is no longer needed: Help > About screen reports if UNACE plugin is installed and features a button for quick removal of the plugin.

Office, Open Office, MSI, EXE files…
Many file formats are actually containers composed by resource objects which can be browsed and extracted using PeaZip.
However since those file types are not commonly used as archives themselves (for browsing or for extraction of resources), PeaZip by default don't handle them as archives, unless explicitly asked with "Open as archive" function. Consequently, double clicking on files of those types within PeaZip starts the associated application rather than opening them as archives.

UTF-8 support
PeaZip GUI is fully UTF-8 compliant (since release 2.2.0), and internal filename handling functions can work with UTF-8 filenames since release 3.0.1 in Linux, and since release 5.8.0 in Windows (on Windows platform, internal filesystem and string management functions were made fully Unicode aware since 6.0.0 release compiled with Lazarus 1.6/FPC3.0).
PeaZip can generate UT-8 archiving/extraction scripts ("Console" tab).
For more information check UTF-8 support status in Lazarus and in PeaZip sources comments.

SendTo and context menu
Launching the program through links in SendTo, or any other link to the application, or through scripts, allows receiving multiple input arguments for each instance of the program, while entries in Windows' context menu, by design, can receive a single input argument for each instance.
PeaZip's Context Menu items can handle multiple input argument through a temporary files list, the temporary files are automatically deleted (otherwise the mechanism would not function), and in any case it is possible to force deleting them in Options > Settings > Reset peazip-tmp (or resetting PeaZip).
For a similar Windows' design choice, on pre-Windows 7 systems, program's entries in Windows context menu are not grouped in a submenu since it will involve much more complex approach than creating separate registry entries; registry method is now (for Windows 7 and later) recommended over the previous COM method.
SendTo menu are simple link files, they can be freely renamed, or have the icon changed, and can be copied or moved to one user to another as needed, they are located in C:\Documents and Settings\(username)\SendTo (Windows XP) or C:\Users\(username)\AppData\Roaming\Microsoft\Windows\SendTo (Vista/7), assuming C: as system's disk.
All available SendTo links are featured in SendTo.zip file in PeaZip's path, more can be built using the switches explained in "Customisation and scripting".
Context menu entries are written in the registry; the exact location, different in 7 and previous systems, can be read in the InnoSetup scripts in "installer" dir of source package. They can be modified using a registry editor, but manually modifying registry is not generally recommended to users.

Compilation, build and porting
PeaZip, and Pea are written in FreePascal (highly compatible with Delphi and ObjectPascal languages) and require Lazarus IDE to be compiled; Windows setup scripts (.iss files) are developed using Inno Setup.
To compile PeaZip binaries open the .lpi file of the desired binary (i.e. peach.lpi for peazip binary) and select "build all".

FreePascal supports multiple widgetsets (Win32, WinCE, GTK1, GTK2, Qt, Carbon, fpGUI) to allow compilation of GUI applications for the various supported systems and to create different "flavours" of the application for platforms supporting multiple widgetsets (i.e Linux).

PeaZip's sourcecode is cross platform, platform-specific code portions are contained in conditional compilations blocks.

Deploying the application to other targets than MSWINDOWS, LINUX and BSD may require adaptation of those platform specific areas (and possibly other fine-tunings).

PeaZip will also need various backend compression and archiving applications to be reachable in expected directories within the application's path, please refer to the structure of any of the precompiled packages, either installable or portable, to see what third parts binaries must be included, and refer to respective Authors for ports of those utilities.

Being PeaZip programmed as frontend/backend application, missing or unwanted backend binaries can be omitted, at the cost of losing the ability of handling supported formats; for the same reason, backend binaries can be freely replaced with 64 bit counterparts or with updated versions (which will work fine as long as they follow the same syntax).

PeaZip code should be fairly easy to port on Delphi and other Pascal dialects; the underlying **Wolfgang Ehrhardt crypto library** is explicitly written to be portable to most or all Pascal dialects, however due to some ASM parts some of its features may be x86 processor specific.

Wolfgang Ehrhardt, one of most influential Delphi developer, passed away in early 2019 [1], [2].
His legacy is mathematic and cryptographic code of impressive quality, generously released as Open Source. This is a huge loss for Delphi community and a sore loss for me. His was the first, and the best, cryptographic library I used in Lazarus/FreePascal, and a natural choice for a project as PeaZip.
May his work live long and continue teaching and inspiring new generations of programmers!

Qt issues
PeaZip for Linux is available as alternative to GTK2 version, compiled for Qt5 (64 bit packages) Qt4 (legacy 32 bit packages)
Lazarus/Freepascal support for Qt widgetset is less mature than support for GTK2, so if experiencing problem with Qt version it is recommended to use GTK2 version, that can run on KDE as well (as Qt version can run on Gnome as well) because PeaZip is meant to be as desktop- and widgetset- independent as possible.
Some issues happened in the past with Qt builds are:
- in some cases the drag and drop from system to the Qt version of the application will not work
- in some cases PeaZip will run only starting it with –style=cleanlooks parameter (this is not inherited when PeaZip starts other processes, which consequently may not run)

Install/uninstall on Linux
PeaZip is meant to be desktop independent as possible, so it is not limited to be used i.e. under KDE or Gnome, and does not requires different packages to run on different desktop environments.
PeaZip/res/share/batch/ contains sample scripts and examples to be used in all freedesktop compliant DE.

On PeaZip's official website are featured generic Linux packages, meant to be compatible with as many distributions and versions as possible, providing the system supports the widgetset (GTK2 * or Qt **):
TAR.GZ (PeaZip Portable) does not need to be installed, it can be used on any Linux family extracting the package and running peazip binary in program's directory.

Third parts packages explicitly targeted to a specific distribution/version are highly recommended for best integration with a specific Linux distribution and version, still the Portable packages are recommended for deploying the application on a network path, or removable device to be used in an hybrid environment, or if you just want to test the application without modifying the host system.

Please note that PeaZip for Linux 64 bit tries to provide all main binaries in x86_64 form every time it is possible, but ia32-libs are required if it is desired to run 32 bit -only binary (e.g. FreeArc compressor / decompressor) on 64 bit systems.

If a package does not run on a specific system please launch peazip binary on a console to get a more detailed error report.

* On some Linux distributions the GTK2 version may require to install some components of gtk/gdk libraries, like libgdk_pixbuf, those components are quite common and can be usually found in repositories of each distribution.

** Qt widgetset packages requires Qt5 (Qt 4.3 for legacy 32 bit Linux packages), and needs libQt5Pas.so.1.2.6 often referred as libQt5Pas.so.1 (libqt4intf.so and libqt4pas5 for 32 bit) or equivalent binary installed in /usr/local/lib or equivalent directory, such as /usr/lib or /usr/lib32.
You can get up to date missing libraries using distribution-specific ways, i.e. apt-get install package:architecture, or copy the file manually and run ldconfig.