Desktop Icon Backup Manager

User Manual

Software Version: ***1.7.5***

Development: **mapi68**

July 24, 2026

Desktop Icon Backup Manager

User Manual

Release Date: July 24, 2026

Software Version: 1.7.5

Copyright © 2026 mapi68

All rights reserved

This document and the software described herein are provided under license and may be used or copied only in accordance with the terms of such license.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the copyright holder.

Disclaimer: The information contained in this document is subject to change without notice. While every effort has been made to ensure accuracy, the author makes no warranty of any kind with regard to this material and assumes no responsibility for errors or omissions.

Support & Updates

Please visit the project repository:

github.com/mapi68/desktop-icon-backup-manager

Support the Developer

If you find this software useful, consider a donation:

ko-fi.com/mapi68

Contents

1 Introduction **1**
- 1.1 Key Features . . . 1
- 1.2 System Requirements . . . 2

2 Installation and Startup **3**
- 2.1 Installation . . . 3
- 2.2 First Run . . . 4
- 2.3 Quick Start Guide . . . 4

3 Main Interface **5**
- 3.1 Interface Overview . . . 5
- 3.2 Main Action Buttons . . . 5
  - 3.2.1 SAVE QUICK BACKUP . . . 5
  - 3.2.2 RESTORE LATEST . . . 6
  - 3.2.3 BACKUP MANAGER . . . 7
  - 3.2.4 SHOW/HIDE ICONS . . . 7

4 File Menu **8**
- 4.1 Scramble Desktop Icons (Random) . . . 8
- 4.2 Desktop Icons Visibility . . . 9
- 4.3 Export Backups . . . 10
- 4.4 Import Backups . . . 11
- 4.5 Remove All Backups . . . 11
- 4.6 Exit . . . 12

5 Settings Menu **12**
- 5.1 Start Minimized to Tray . . . 12
- 5.2 Auto-Save on Exit . . . 12
- 5.3 Auto-Restore on Startup . . . 13
- 5.4 Check for Updates on Startup . . . 14
- 5.5 Enable Adaptive Scaling on Restore . . . 14
  - 5.5.1 How Adaptive Scaling Works . . . 14
  - 5.5.2 When Scaling Is Applied . . . 15
  - 5.5.3 Log Messages for Scaling . . . 15
- 5.6 Minimize to Tray on Close . . . 16
- 5.7 Automatic Backup Cleanup Limit . . . 16
- 5.8 Auto-Hide Desktop Icons . . . 17
  - 5.8.1 Enable Auto-Hide Timer . . . 17
  - 5.8.2 Hide After... (Interval Selection) . . . 17
  - 5.8.3 Backup Before Auto-Hide . . . 17
  - 5.8.4 Notify Before Hiding . . . 17
  - 5.8.5 In-Window Countdown Banner . . . 18
  - 5.8.6 Timer Behavior . . . 18
- 5.9 Theme . . . 19
- 5.10 Window Position and Size Persistence . . . 19

6 System Tray Integration **20**
- 6.1 Tray Notifications . . . 20

7 Keyboard Shortcuts **21**

8 Help Menu **21**
8.1 Online User Manual . . . 21
8.2 Check for Updates . . . 21
8.3 Statistics Dashboard . . . 22
8.4 About Dialog . . . 22

9 Advanced Backup Management **22**
9.1 Backup Manager Window . . . 22
9.1.1 Search and Filter Bar . . . 22
9.1.2 Backup Table Columns . . . 23
9.1.3 Live Diff Preview Panel . . . 23
9.1.4 Visual Preview Panel (standard) . . . 24
9.1.5 Information Panel . . . 24
9.1.6 Available Operations . . . 24
9.2 Backup Comparison . . . 25
9.2.1 How to Use Comparison . . . 25
9.2.2 Comparison Engine Details . . . 26
9.2.3 Technical Details . . . 26
9.3 Backup File Format . . . 27
9.3.1 File Naming Convention . . . 27
9.3.2 Filename Parsing . . . 27
9.3.3 JSON File Structure . . . 27

10 Command Line Interface (CLI) **28**
10.1 Available Arguments . . . 28
10.2 Technical Implementation . . . 29
10.3 Example Batch Script . . . 29
10.4 Automation with Task Scheduler . . . 29
10.5 Batch Operations with JSON . . . 29

11 Best Practices **30**
11.1 Optimal Backup Strategy . . . 30
11.2 Backup Naming Best Practices . . . 30
11.3 Multi-Monitor Management . . . 31
11.4 Resolution Change Scenarios . . . 31
11.5 Regular Maintenance . . . 31
11.6 Backup Archiving Strategy . . . 32

12 Troubleshooting **32**
12.1 System Interaction and Permissions . . . 32
12.2 Common Issues . . . 33
12.2.1 Error: "Unable to find desktop ListView control" . . . 33
12.2.2 Icons Restored to Wrong Positions . . . 33
12.2.3 Backup File Not Found . . . 33
12.2.4 Program Crashes on Startup . . . 34
12.2.5 Scramble Operation Fails . . . 34
12.2.6 Settings Not Persisting . . . 35
12.3 Diagnostics . . . 35
12.4 Performance Issues . . . 35
12.4.1 Slow Save/Restore Operations . . . 35
12.4.2 High Memory Usage . . . 35

buy me a coffee

13 Frequently Asked Questions (FAQ) **36**
13.1 General Questions . . . 36
13.2 Feature Questions . . . 36
13.3 Technical Questions . . . 37

14 Technical Information **37**
14.1 Software Architecture . . . 37
14.2 Remote Memory Access Process . . . 38
14.3 Data Structures . . . 39
14.4 Threading Architecture . . . 39
14.5 Desktop Refresh Mechanism . . . 40
14.6 Settings Storage . . . 40
14.6.1 Settings File Location . . . 40
14.6.2 Complete Settings Structure . . . 41
14.7 Security and Privacy . . . 42

15 Development and Contribution **42**
15.1 Building from Source . . . 42
15.2 Translation/Localization . . . 43

16 Screenshots **44**

buy me a coffee

buy me a coffee

Desktop Icon Backup Manager

User Manual

1 Introduction

Desktop Icon Backup Manager is a professional tool designed to save and restore Windows desktop icon positions with advanced features for managing multiple layouts, adaptive scaling across different resolutions, and complete automation through configurable settings.

Windows has a long-standing bug — present in every version from Windows 7 through Windows 11 — that silently rearranges desktop icons after events such as a Windows Update, connecting or disconnecting a monitor, waking from sleep, or exiting a full-screen game. Desktop Icon Backup Manager solves this by letting you snapshot your exact layout at any time and restore it in a single click, with a live preview of every change before you commit.

1.1 Key Features

- **Quick Backup**: Save icon positions with optional descriptive tags
- **Profiles**: Built-in dropdown of 14 preset tag names (Work, Gaming, Presentation, Dev / Coding, Meeting, Home, Office, Laptop, Docked / External Monitor, Clean Desktop, Pre-Update, Pre-Reboot, Favourite, Test) for consistent, one-click tag entry; fully translated into all 26 supported languages
- **Inline Tag Editing**: Double-click any tag cell in the Backup Manager table to rename a backup in place — the change is written to the JSON file immediately; no dialog required
- **Colour-coded Tag Bar**: Each distinct tag is assigned a unique colour bar on the left edge of its row in the Backup Manager — consistent tags always share the same colour for instant visual identification without reading the text
- **Export Backups**: Export the selected backup or all backups to a ZIP archive or a destination folder for portability, off-site storage, or sharing between machines
- **Import Backups**: Import `.json` backup files or ZIP archives from another installation; existing files are automatically skipped to prevent overwriting
- **Backup Management**: Dedicated interface with search filtering and sortable table columns to select, restore, or delete specific backups
- **Live Diff Preview**: Color-coded overlay showing exactly which icons will move, appear, or stay in place *before* you confirm a restore
- **Visual Preview**: Mini-map display of icon layouts before restoring with interactive tooltips
- **Backup Comparison**: Compare *any two* backups to see exactly what changed (added, removed, moved icons)
- **Adaptive Scaling**: Automatic adjustment of icon positions across different resolutions

buy me a coffee

- **Show/Hide Desktop Icons**: Quickly toggle visibility of all desktop icons without affecting backup data — useful for clean desktop screenshots, presentations, or focus mode. Accessible via `Ctrl+H` shortcut, button, or File menu submenu
- **Auto-Hide Desktop Icons**: Automatically hide desktop icons after a configurable delay (30 seconds to 30 minutes, or a custom interval in minutes and seconds). Optionally creates an automatic backup before hiding. A live countdown banner is shown inside the main window (label, progress bar, and a one-click disable button) and in the system tray tooltip. Optional desktop notifications are emitted 1 minute and 10 seconds before the hide. The timer resets whenever icons are shown again (manually, after a restore, or after a scramble)
- **Automatic Cleanup**: Configurable limit on the number of backups to retain (5, 10, 25, 50, or unlimited)
- **Theme**: Choose between Light, Dark, or Use System Setting colour schemes, applied instantly with no restart required
- **Random Scramble**: Randomize icon positions with automatic preventive backup
- **System Tray Integration**: Quick access to main functions from the system tray
- **Automation**: Auto-save on exit and auto-restore on startup options
- **Update Checker**: Automatically checks for new versions at startup (configurable); tray notification and log entry when an update is available. Also accessible manually via Help menu
- **Command Line Interface**: Full CLI support for scripting and scheduled tasks
- **Multi-language Support**: Available in 26 languages — Arabic, Czech, Danish, Dutch, English, Finnish, French, German, Greek, Hindi, Italian, Japanese, Korean, Norwegian Bokmål, Polish, Portuguese (BR), Portuguese (PT), Romanian, Russian, Simplified Chinese, Slovenian, Spanish, Swedish, Traditional Chinese, Turkish, and Ukrainian. Auto-detected from Windows locale, or manually overridden

1.2 System Requirements

Minimum Requirements

- Operating System: Windows 7 or higher (fully compatible with Windows 11)
- Python 3.8+ (if running from source)
- Libraries: PyQt6, pywin32
- Disk Space: 50 MB for application + space for backups (typically 2-10 KB per backup)
- Permissions: Standard user permissions (no administrator rights required)

2 Installation and Startup

2.1 Installation

Method 1: Standalone Executable (Recommended)

1. Download the latest zip file from the Releases page:
   `desktop-icon-backup-manager_v1.7.5.zip`
   The archive contains the Windows executable and the PDF user manual.
2. Extract it in any folder (e.g. `C:\Tools\` or a USB drive — the program is fully portable)
3. Run `Desktop Icon Backup Manager.exe`
4. The `icon_backups` folder and `settings.ini` are created automatically next to the executable

Because all data files are stored relative to the executable, the program can be moved or copied to any location at any time without losing settings or backups. No entries are written to the Windows registry and no administrator rights are required.

Tip: Sync Across Multiple PCs with OneDrive

Create a new folder inside your OneDrive (e.g. `C:\Users\YourName\OneDrive\DIBM\`), then extract the release ZIP into it. Since `icon_backups` and `settings.ini` are always stored next to the `.exe`, OneDrive will sync them automatically across all your PCs. Run the `.exe` from that OneDrive folder on each machine. Avoid running the program on two machines at the same time to prevent file conflicts.

Method 2: From Python Source

1. Install Python 3.8 or higher
2. Install dependencies:

   ```
   pip install PyQt6 pywin32
   ```

3. Run the script:

   ```
   python main.py
   ```


buy me a coffee

2.2 First Run

On first run, the program automatically:

- Creates the `icon_backups` folder for backup files
- Generates the `settings.ini` configuration file in the application directory
- Verifies the presence and accessibility of desktop icons
- Places itself in the system tray (if enabled)
- Loads default settings and preferences

Important - Desktop Icons Must Be Visible

Make sure desktop icons are visible (Right-click desktop → View → Show desktop icons). If icons are hidden, the program cannot access their positions and will show an error: "Unable to find desktop ListView control."

2.3 Quick Start Guide

For users who want to get started immediately, here is the essential workflow:

1. **Save your current layout**: Click the green **SAVE QUICK BACKUP** button (or press `Ctrl+S`). Optionally type a descriptive tag in the field above the button, or use the **Profiles** dropdown to the right to pre-fill it with a preset name (Work, Gaming, Presentation, etc.). The backup is saved instantly as a small JSON file in the `icon_backups` folder next to the executable — no configuration needed. Each file records the position of every icon on the desktop along with the current screen resolution and monitor count, so the correct layout can always be matched to the right setup.
2. **Restore a layout**: Click the red **RESTORE LATEST** button to restore the most recent backup, or open the **BACKUP MANAGER** (blue button) to choose a specific backup. Before confirming, the Backup Manager shows a live diff preview — a colour-coded overlay of which icons will move, which are already in position, and which are in the backup but no longer on the desktop. If the saved backup was created at a different resolution, enable *Adaptive Scaling* in Settings to have positions recalculated proportionally before restoring.
3. **Enable auto-protection**: In the Settings menu, enable *Auto-Save on Exit* so your layout is always preserved when closing the program.

Recommended First-Time Setup

1. Save an initial backup with a meaningful tag (e.g., "Default Layout")
2. Enable *Auto-Save on Exit* in the Settings menu
3. Set a *Cleanup Limit* of 10–25 backups to manage disk space
4. Optionally enable *Start Minimized to Tray* for background operation

3 Main Interface

3.1 Interface Overview

The main interface is divided into seven main areas:

1. **Menu Bar**: Contains File, Settings, and Help menus with all advanced options
2. **Progress Bar**: Shows real-time progress during backup, restore, and scramble operations
3. **Tag Input Field**: Optional field for entering descriptive tags for quick saves
4. **Profiles Dropdown**: A combo box to the right of the tag field containing 14 built-in profile presets (Work, Gaming, Presentation, Dev / Coding, Meeting, Home, Office, Laptop, Docked / External Monitor, Clean Desktop, Pre-Update, Pre-Reboot, Favourite, Test). Selecting any entry instantly copies the name into the tag field; the combo resets to its placeholder so the same entry can be re-selected. All names are fully translated into the 26 supported languages.
5. **Action Buttons**: Four large, color-coded buttons for primary operations
6. **Activity Log**: Displays operation status, warnings, and errors in real-time with timestamps. Every entry is also silently written to `history.log` next to the executable (maximum 500 entries, auto-trimmed). The file persists across sessions and can be used for auditing or troubleshooting. The **Clear Log** button clears both the on-screen log and the `history.log` file
7. **Status Bar**: Shows current primary monitor resolution and quick operation messages

3.2 Main Action Buttons

3.2.1 SAVE QUICK BACKUP

Color: Green | Shortcut: Ctrl+S | Button icon: 💾

Immediately saves current desktop icon positions to a new timestamped backup file. You can add an optional descriptive tag in the field above the button for easier identification later. Use the **Profiles** dropdown to the right of the tag field to quickly pre-fill it with one of 14 built-in names (Work, Gaming, Presentation, Dev / Coding, Meeting, Home, Office, Laptop, Docked / External Monitor, Clean Desktop, Pre-Update, Pre-Reboot, Favourite, Test) — then edit freely before saving if needed.

Note on Ctrl+S vs Button Click

When using the `Ctrl+S` shortcut, the backup is saved with the tag *"Quick Backup (Shortcut)"*, regardless of what is typed in the tag field. To use a custom tag, type it in the field and then click the **SAVE QUICK BACKUP** button.

Operation Sequence:

1. Disables UI buttons to prevent concurrent operations
2. Displays progress bar
3. Scans all icons present on the desktop via Windows ListView control
4. Records exact X and Y pixel coordinates for each icon
5. Saves metadata including resolution, icon count, timestamp, and description

6. Applies automatic cleanup if limit is configured (deletes oldest backups)
7. Forces desktop refresh to ensure visual consistency
8. Re-enables UI and displays completion message

Technical Data Structure

The backup system captures a comprehensive snapshot of the desktop environment:

- **Display Metadata**: Records monitor count, screen names, individual resolutions, and the `devicePixelRatio` value.
- **JSON Schema**: Stores `icon_count`, an ISO `timestamp`, a `description` field, and exact X/Y pixel coordinates for every icon.
- **Coordinate Precision**: Icon positions are read through the Windows API message `LVM_GETITEMPOSITION`. The application declares itself Per-Monitor-v2 DPI-aware at startup, so both the saved coordinates and the saved resolution are in physical pixels, matching the monitor's physical dimensions regardless of Windows display scaling.

Example Log Output:

```
[14:30:15] Starting new timestamped backup...
[14:30:15]   (Tag: Work Setup Final)
[14:30:15] Monitor Resolution: 1920x1080
[14:30:15] Found 12 icons. Starting scan...
[14:30:16]   Saved 12 icons to backup file '1920x1080_20241211_143015.json'
[14:30:16]   (Description: Work Setup Final)
```

3.2.2 RESTORE LATEST

Color: Red | Button icon: ↺

Restores icon positions from the most recent backup file. Before restoring, a detailed confirmation dialog displays complete backup information.

Confirmation Dialog Information:

- Full backup file name
- Saved resolution with comparison to current resolution
- Total number of icons in the backup
- Tag/description (if available)
- Backup date and time in human-readable format (YYYY/MM/DD HH:MM:SS)

Restore Process:

1. Validates backup file existence and format
2. Disables window redrawing for performance
3. Reads saved icon positions from JSON file
4. Applies adaptive scaling if enabled and resolutions differ
5. Maps saved icon names to current desktop icons
6. Updates each icon position using Windows API
7. Re-enables redrawing and forces desktop refresh

buy me a coffee

8. Reports statistics (icons restored, icons skipped)

Smart Restore Behavior

If an icon exists in the backup but not on the current desktop, it will be skipped and reported. This prevents errors when restoring backups from different system states.

3.2.3 BACKUP MANAGER

Color: Blue | Shortcut: Ctrl+M | Button icon: ↺

Opens the comprehensive backup management window (default size: 1600×800, resizable) that provides:

- **Search/Filter Bar**: Real-time filtering by tag, resolution, or date
- **Sortable Table**: Click any column header to sort (description, resolution, icon count, timestamp). The Tag/Description column stretches to fill available space; Resolution, Icons, and Timestamp columns have fixed widths with centre-aligned content.
- **Live Diff Preview**: Before restoring, see exactly which icons will move (orange → red arrow), which are already in place (blue), and which are missing from the desktop (green)
- **Legend Panel**: A dedicated legend widget displayed *outside* the canvas, to the right of the information box, showing the colour coding used in the diff preview (blue = in place, orange/red arrow = will move, green = not on desktop)
- **Detailed Information**: Extended metadata display for the selected backup (file name, icon count, resolution, description, formatted timestamp), shown alongside the legend panel
- **Context Menu**: Right-click for quick restore, compare with latest, or delete actions
- **Flexible Comparison**: Compare the selected backup against *any* other backup (not just the latest) via the "Compare Two Selected..." button
- **Inline Tag Editing**: Double-click a cell in the *Tag / Description* column to edit the label directly in the table. Press `Enter` or click outside to save; press `Escape` to cancel. Columns 2–4 (Resolution, Icons, Timestamp) remain read-only and open the restore dialog on double-click as before
- **Export / Import**: Two buttons in the bottom toolbar (`Export Backups...` and `Import Backups...`) let you archive or transfer backups — see Section 4.3

3.2.4 SHOW/HIDE ICONS

Color: Purple | Shortcut: Ctrl+H | Button icon: 👁

Toggles the visibility of all desktop icons without affecting any backup data or icon positions. This feature is useful for creating clean desktop screenshots, presentations, or focus-mode work sessions.

Three Access Methods:

1. **Button**: Click the purple **SHOW/HIDE ICONS** button on the main window
2. **Keyboard**: Press `Ctrl+H`
3. **Menu**: `File > Desktop Icons Visibility > Show/Hide Icons`

Behavior:

- If icons are **currently visible**, clicking toggles them *hidden*
- If icons are **currently hidden**, clicking toggles them *visible*
- The operation is **instantaneous**
- Icon **positions are not affected** — only visibility changes
- **No backup is created** — this is a temporary display toggle

System Tray Integration:

You can also control icon visibility from the system tray menu (right-click the tray icon):

- **Show/Hide Desktop Icons** — toggles current state
- **Show Desktop Icons** — explicitly show (no-op if visible)
- **Hide Desktop Icons** — explicitly hide (no-op if hidden)

Important: Session Scope

Icon visibility state is **not persistent** — if you hide icons and then log out or reboot, icons will return to their normal state on the next login. This is by design: the visibility toggle is a temporary display preference, not a system-level setting.

Use Cases:

- **Screenshots**: Hide icons before capturing desktop images for documentation or tutorials
- **Presentations**: Display a clean, professional desktop during meetings or demos
- **Focus Mode**: Minimize visual distractions while working on specific tasks
- **System Maintenance**: Temporarily hide icons during updates, driver installation, or maintenance tasks

4 File Menu

4.1 Scramble Desktop Icons (Random)

This function completely randomizes the positions of all desktop icons across the available screen area, creating a chaotic but fun layout.

Mandatory Pre-Scramble Backup

Before scrambling, an automatic backup is **always** created with the tag "**Backup before Scramble**". This ensures you can always restore the previous layout. The scramble operation will abort if this backup fails.

Detailed Procedure:

1. Select `File > Scramble Desktop Icons (Random)`
2. Confirm the operation in the warning dialog
3. Program creates mandatory backup (first 50% of progress)
4. Desktop redrawing is temporarily disabled for performance

5. Each icon is assigned random X and Y coordinates within screen bounds
6. Positions use margin buffer (100 pixels) to prevent edge clipping
7. Desktop redrawing is re-enabled (remaining 50% of progress)
8. System refresh signals are broadcast

Technical Details:

- Uses `GetSystemMetrics` to determine virtual screen dimensions
- Random positions: `random.randint(margin, screen_dimension - margin)`
- Margin prevents icons from being placed too close to screen edges
- Operation completes even if some icons fail to move

4.2 Desktop Icons Visibility

The **Show/Hide Desktop Icons** feature allows you to quickly toggle the visibility of all desktop icons on your desktop without affecting any backup data. This is useful for:

- **Clean Screenshots**: Hide icons before capturing desktop screenshots
- **Presentations**: Present a clean, distraction-free desktop
- **Focus Mode**: Hide visual clutter while working on specific tasks
- **System Maintenance**: Hide icons temporarily during updates or system changes

Three Ways to Toggle Icons:

1. **Menu**: `File > Desktop Icons Visibility > Show/Hide Icons`
2. **Button**: Main window `SHOW/HIDE ICONS` button (purple)
3. **Keyboard**: `Ctrl+H`

Submenu Options:

- **Show/Hide Icons** (`Ctrl+H`): Toggle visibility — shows if hidden, hides if visible
- **Show Icons**: Explicitly show all icons (no-op if already visible)
- **Hide Icons**: Explicitly hide all icons (no-op if already hidden)

System Tray Access: You can also toggle icons from the system tray:

- Right-click the tray icon to access the context menu
- **Show/Hide Desktop Icons** — toggles visibility
- **Show Desktop Icons** — explicitly show
- **Hide Desktop Icons** — explicitly hide

Important Notes

- Icon visibility is **independent** of backups — toggling visibility does **not** affect saved layouts
- The state is **not** saved between sessions — icons return to their previous state on next login/reboot
- This feature uses Windows API to control the desktop icon view window; it does not delete or move any icons
- Some third-party desktop customization tools may interfere with icon visibility control

4.3 Export Backups

Exports the selected backup or all backups to an archive or folder. Accessible from:

- **Backup Manager** bottom toolbar: `Export Backups...`
- **Main menu**: `File > Export Backups...`

Export Workflow:

1. A dialog asks *what* to export:
   - **All backups** (shows the total file count)
   - **Selected backup only** (shows the filename)
2. A second dialog asks *how* to export:
   - **ZIP archive (.zip)**: All selected files packed into one compressed archive; a save-file dialog lets you choose the name and destination.
   - **Folder (copy .json files)**: A directory picker lets you choose a destination folder; JSON files are copied there directly.
3. A summary dialog reports the number of files exported, or a partial-export warning if any file could not be copied.

Use Cases

- **Off-site storage**: ZIP to a USB drive or cloud folder before a system reinstall
- **Migration**: Export all backups, then import on a new machine with `Import Backups...`
- **Sharing**: Send a ZIP containing your layout to a colleague with the same monitor setup
- **Archiving**: Keep a permanent copy of a perfect layout that you want to preserve independently of the cleanup limit

4.4 Import Backups

Imports `.json` backup files or ZIP archives from another location or installation. Accessible from:

- **Backup Manager** bottom toolbar: `Import Backups...`
- **Main menu**: `File > Import Backups...`

Import Workflow:

1. A multi-selection file dialog opens, filtered to `*.json` and `*.zip`. You can select multiple files of either type in a single operation.
2. For each selected file:
   - **.json**: Validated as JSON before being copied to the `icon_backups` folder.
   - **.zip**: The archive is opened; every `.json` entry inside is validated and extracted individually.
3. Files whose name already exists in `icon_backups` are **skipped** (never overwritten).
4. A summary dialog reports:
   - `[OK] Imported` — new files successfully added
   - `[SKIP] Skipped (already exist)` — duplicates left untouched
   - Errors (invalid JSON, corrupted ZIP, etc.)
5. The Backup Manager table refreshes automatically to show the newly imported backups.

Duplicate Detection

Duplicate detection is based on the **filename** only. If you rename a JSON file externally and re-import it, it will be treated as a new file and imported without being skipped.

4.5 Remove All Backups

Permanently deletes all saved backup files from the `icon_backups` folder.

Warning - Irreversible Operation

This action cannot be undone. All backups will be permanently deleted from the system.

Deletion Process:

1. Counts all JSON backup files; if none exist, shows an informational message and stops
2. Shows a confirmation dialog asking whether to proceed
3. Iterates through each backup file and attempts deletion
4. Logs the total number of successfully deleted files, and the total number of failures (if any)
5. Shows a summary dialog: "Success" if all files were deleted, or "Error" if some could not be removed

4.6 Exit

Closes the application gracefully with multiple configurable behaviors:

Exit Behaviors:

- Saves current window geometry (position and size) to settings
- Creates automatic backup if "Auto-Save on Exit" is enabled
- Respects "Minimize to Tray on Close" setting if window is closed via X button
- Saves all pending settings changes
- Properly closes system tray icon
- Terminates console window if running as PyInstaller executable

Exit Methods:

- Menu: `File > Exit`
- Keyboard: `Ctrl+Q`
- System Tray: Right-click icon → Exit
- Window Close (X button): Behavior depends on "Minimize to Tray" setting

5 Settings Menu

5.1 Start Minimized to Tray

When enabled, the application starts hidden in the system tray instead of showing the main window.

Use Cases:

- Users who want the program always available in background
- Systems with auto-restore enabled (no need to see the window)
- Reducing desktop clutter while maintaining quick access

Behavior:

- System tray notification: "Application started minimized to system tray"
- Notification duration: 2 seconds
- Double-click tray icon or use context menu to show window

5.2 Auto-Save on Exit

Automatically creates a backup of current icon positions when closing the program normally (not via Task Manager or system crash).

Strongly Recommended

This option is highly recommended to avoid data loss from unsaved layout changes. It provides a safety net for forgetful users and ensures your latest layout is always preserved.

Technical Implementation:

- Backup is created in `closeEvent` before application exit
- Tag: "Auto-Save on Exit"
- Respects configured automatic cleanup limit
- Does not require user confirmation
- Logs operation silently (visible only if window is open)
- If backup fails, application still exits

Smart Deduplication:

If a **Quick Backup** was performed manually within the last **`AUTO_SAVE_SKIP_SECONDS` seconds** before closing, the Auto-Save is automatically skipped to avoid creating a redundant identical backup. The application tracks the time of the last completed backup in memory for a reliable and instant check.

> **Note**
>
> This deduplication only applies when Auto-Save on Exit is enabled. If no manual backup has been performed in the current session, the Auto-Save proceeds normally as a safety fallback.

Interaction with Other Settings:

- Works independently of "Minimize to Tray on Close"
- Complements "Auto-Restore on Startup" for seamless layout preservation
- Cleanup limit prevents unlimited backup accumulation
- Skips redundant save if a Quick Backup was made within the last `AUTO_SAVE_SKIP_SECONDS` seconds

5.3 Auto-Restore on Startup

Automatically restores the most recent backup when the application starts.

Detailed Behavior:

- Restore occurs 1 second after startup (QTimer delay)
- Uses latest backup file by timestamp
- Applies configured adaptive scaling settings
- Shows full operation log in main window
- No user confirmation required
- If restore fails, application continues normally

> **Important Consideration**
>
> When combined with "Auto-Save on Exit", this creates an automatic save/restore cycle. While convenient, be cautious: if icons are in wrong positions when exiting, they will be automatically restored to those wrong positions on next startup. Use Backup Manager to restore specific known-good layouts if needed.

Recommended Use Cases:

- Single-monitor setups with stable icon arrangements

- Users who frequently restart their computer
- Systems where Windows randomly moves icons (common issue)
- Combination with laptop docking/undocking workflows

5.4 Check for Updates on Startup

When enabled (default: on), the application silently checks GitHub for a new version 10 seconds after startup.

Behavior when an update is available:

- A tray notification is shown: "A new version is available! (x.x.x)"
- Notification duration: 8 seconds
- A log entry is added to the Activity Log

Manual check: Select `Help > Check for Updates...` at any time to open the update dialog, which shows the installed version, the latest version from GitHub, and a "Download Update" button that opens the releases page in your browser.

Network Note

The check connects only to GitHub raw content to read a single plain-text `version.txt` file. No personal data is sent. If the network is unavailable, the check silently fails.

5.5 Enable Adaptive Scaling on Restore

Enables intelligent scaling of icon positions when restoring a backup created at a different screen resolution.

DPI Awareness

The application declares itself Per-Monitor-v2 DPI-aware. This has an important consequence on resolution values: the saved resolution is always a physical pixel dimension, regardless of the Windows display scaling factor (100%, 125%, 150%, 200%). For example, a 2560×1440 monitor at 125% scaling saves as `2560x1440`.
Because both the saved resolution and the icon coordinates are captured in the same coordinate system (physical pixels), adaptive scaling produces consistent results whether you run at 4K 100%, 2K 150%, or any other DPI configuration.

5.5.1 How Adaptive Scaling Works

The system calculates independent scaling factors for X and Y axes based on resolution differences:

$$\text{scale}_x = \frac{\text{current width}}{\text{saved width}}$$
$$\text{scale}_y = \frac{\text{current height}}{\text{saved height}}$$

Each saved coordinate is then transformed using these factors:

$$x_{\text{new}} = \lfloor x_{\text{saved}} \times \text{scale}_x \rfloor$$

$$y_{\text{new}} = \lfloor y_{\text{saved}} \times \text{scale}_y \rfloor$$

Practical Example - Upscaling:

- Backup saved at 1920×1080 (Full HD)
- Restore on 2560×1440 (2K/QHD) monitor
- Scaling factors: $\text{scale}_x = 1.333$, $\text{scale}_y = 1.333$
- Icon at (960, 540) → restored at (1280, 720) [screen center maintained]
- Icon at (100, 100) → restored at (133, 133) [relative position preserved]

Practical Example - Downscaling:

- Backup saved at 2560×1440
- Restore on 1920×1080 monitor
- Scaling factors: $\text{scale}_x = 0.75$, $\text{scale}_y = 0.75$
- Icon at (2000, 1200) → restored at (1500, 900)
- Icon at (200, 200) → restored at (150, 150)

5.5.2 When Scaling Is Applied

Scaling Activation Conditions

Adaptive scaling only activates when ALL these conditions are met:

1. "Enable Adaptive Scaling on Restore" is checked in Settings
2. Backup file contains resolution information (newer format)
3. Current resolution differs from saved resolution
4. Both resolutions can be successfully parsed

If any condition fails, raw coordinates are used without modification.

5.5.3 Log Messages for Scaling

When Scaling Is Active:

```
[14:45:20] Adaptive Scaling enabled: X=1.333, Y=1.333
```

When Scaling Is Disabled (with resolution mismatch):

```
[14:45:20] Warning: Resolution mismatch!
```

When Scaling Is Disabled (resolutions match): No scaling message is logged; restore proceeds silently with raw coordinates.

5.6 Minimize to Tray on Close

When enabled, clicking the window's X button minimizes the program to the system tray instead of completely closing it.

Complete Closure Methods:

- Use menu: `File > Exit`
- Keyboard shortcut: `Ctrl+Q`
- Right-click tray icon: Select "Exit"

User Notification:

- Tray notification: "Application minimized to system tray. Click or double-click to restore."
- Notification duration: 2 seconds
- Window geometry is saved before minimizing

Recommended Usage

Enable this option if you want to keep the program always running in the background for quick access to tray menu functions (Quick Save, Restore Latest) without the main window occupying taskbar space.

5.7 Automatic Backup Cleanup Limit

Configures the maximum number of backup files to maintain. When this limit is exceeded after a save operation, the oldest backups are automatically deleted.

Available Options with Values:

- **Disabled (Keep All)**: No limit, all backups retained indefinitely (value: 0)
- **Keep Last 5**: Maintains only 5 most recent backups (value: 5)
- **Keep Last 10**: Maintains only 10 most recent backups (value: 10)
- **Keep Last 25**: Maintains only 25 most recent backups (value: 25)
- **Keep Last 50**: Maintains only 50 most recent backups (value: 50)

Cleanup Operation Details:

1. Cleanup runs automatically after each successful save
2. Backups are sorted by timestamp, **newest first**
3. Files beyond the configured limit (i.e., the oldest ones) are deleted
4. Deletion is logged with filename for each file
5. The backup just created is never deleted
6. If limit is 0 (Disabled), cleanup is completely skipped

Example Log Output (Limit = 10, Current Count = 12):

```
[14:50:30] Cleanup needed: Current count (12) exceeds limit (10).
           Deleting 2 oldest file(s).
[14:50:30]   Deleted oldest backup: 1920x1080_20241201_100000.json
[14:50:30]   Deleted oldest backup: 1920x1080_20241202_120000.json
```

```
[14:50:30] Cleanup complete. Total deleted: 2 file(s).
```

Recommended Strategy by User Type

- **Casual Users**: Keep Last 5-10 (sufficient for most needs)
- **Daily Users**: Keep Last 10-25 (balances history and space)
- **Power Users**: Keep Last 25-50 (extensive history for experimentation)
- **Archivists**: Disabled (manual management, export important backups)

5.8 Auto-Hide Desktop Icons

Automatically hides all desktop icons after a configurable delay. Accessible from **Settings > Auto-Hide Desktop Icons** and from the **system tray** context menu (as a quick on/off toggle).

5.8.1 Enable Auto-Hide Timer

When enabled, a countdown starts each time desktop icons become visible (after a manual show, a restore, or a scramble). When the countdown reaches zero, the icons are automatically hidden.

- Stored as `autohide_enabled` in `settings.ini`
- Default: `false`
- The checkmark is synchronised between the Settings menu and the system tray toggle

5.8.2 Hide After... (Interval Selection)

Choose a preset or custom interval:

Presets: 30 seconds, 1 minute, 2 minutes, 5 minutes, 10 minutes, 15 minutes, 30 minutes.

Custom: Opens a dialog with two spin boxes (minutes 0–60 and seconds 0–59) for precise intervals such as 1 minute 33 seconds. Minimum accepted value: 10 seconds.

- Stored as `autohide_seconds` in `settings.ini`
- Default: `300` (5 minutes)

5.8.3 Backup Before Auto-Hide

When enabled, the application creates a silent backup immediately before hiding icons. This ensures you always have a fresh snapshot of the visible layout. The backup uses the tag *"Auto-Hide Backup"* and respects the configured cleanup limit.

- Stored as `autohide_backup_before_hide` in `settings.ini`
- Default: `true`

5.8.4 Notify Before Hiding

When enabled, the application emits a desktop notification (system tray balloon) as the deadline approaches, giving you a chance to interact with the desktop before the icons disappear.

- Two thresholds, fired once per timer cycle: **60 seconds** and **10 seconds** before the hide
- The notification title is *"Auto-Hide"* and the body reads *"Desktop icons will be hidden in 1:00"* (or *"... in 10s"*)

- Thresholds are reset every time the timer starts or restarts, so you get the full sequence again after any Show/Restore/Scramble
- Stored as `autohide_notify_enabled` in `settings.ini`
- Default: `true`

5.8.5 In-Window Countdown Banner

While the timer is running, the main window shows a compact **countdown banner** just below the menu bar. It contains:

- A label with the current remaining time, for example *"Auto-Hide in 4:37"*
- A thin horizontal progress bar that fills from right to left as the deadline approaches
- A small **stop button** on the right side that disables Auto-Hide with a single click, equivalent to unchecking *"Enable Auto-Hide Timer"* from the menu

The banner is visible whenever Auto-Hide is enabled. While the timer is actively counting down, it shows the remaining time together with the progress bar. When the feature is enabled but the timer is not currently ticking — for example at application launch if Auto-Hide was already persisted as enabled — the banner shows the configured interval without the progress bar, so the state is always obvious at a glance.

5.8.6 Timer Behavior

- The timer **starts** as soon as Auto-Hide is enabled (either from the Settings menu or from the tray). It also starts at application launch when Auto-Hide is already enabled and icons are visible
- The timer **resets** when icons become visible again — after a manual Show, a Restore, or a Scramble operation
- The timer **stops** when icons are hidden (manually or by the timer itself), or when auto-hide is disabled
- A live countdown is shown in the **main-window banner** and in the **system tray tooltip** (e.g., *"Auto-Hide in 3:42"*)
- When icons are hidden, the tooltip shows *"Desktop icons are hidden"*

Example Log Output:

```
[18:30:00] Auto-Hide enabled: icons will be hidden after 1 minute(s)
           33 second(s).
[18:31:33] Auto-Hide: creating backup before hiding icons...
[18:31:33] Auto-Hide: hiding desktop icons now.
```

Tip

Auto-Hide is particularly useful during presentations or screen-sharing sessions where you want a clean desktop. Enable it, set a short interval, and your desktop tidies itself automatically.

5.9 Theme

Controls the application's colour scheme. Accessible from **Settings > Theme**.

Three options are available:

Use System Setting The application follows the Windows colour mode (Light or Dark) configured in *Settings > Personalisation > Colours*. This is the default. If Windows switches mode, restart the application for the change to propagate.

Light Forces a light palette regardless of the Windows system setting.

Dark Forces a dark palette regardless of the Windows system setting.

Behaviour:

- The selected option is indicated by a checkmark in the **Settings > Theme** submenu.
- The palette is applied **immediately** — no restart is required.
- Stored as `theme_mode` in `settings.ini` (`system` / `light` / `dark`).
- Default: `system`.

> **Note**
>
> The Theme setting controls the application's own palette only. It does not change the Windows system colour mode or affect any other application.

5.10 Window Position and Size Persistence

The application automatically remembers your window position and size:

- Window geometry saved when closing normally
- Saved to `settings.ini` under `[MainWindow]/geometry`
- Automatically restored on next startup
- Default position: 100,100 with size 800×650 pixels
- Works independently of "Minimize to Tray" setting

Format in settings.ini:

```
[MainWindow]
geometry=@Rect(100 100 800 650)
```

> **Note**
>
> If the saved position is off-screen (e.g., after monitor disconnection), Qt automatically repositions the window to visible area.

6 System Tray Integration

Right-click on the system tray icon to access quick functions:

- **Quick Save**: Creates immediate backup with tag "Quick Save (Tray)"
- **Restore Latest**: Restores most recent backup (shows confirmation dialog with live diff preview before proceeding)
- **Show/Hide Desktop Icons**: Toggles visibility of all desktop icons
- **Show Desktop Icons**: Forces icons to be visible
- **Hide Desktop Icons**: Forces icons to be hidden
- **Auto-Hide Timer**: Checkable toggle to enable or disable the auto-hide countdown — synchronised with the Settings menu. When active, both the tray tooltip and the in-window banner display a live countdown
- **Show Window**: Brings main window to front and activates it
- **Exit**: Completely closes application (bypasses "Minimize to Tray" setting)

Icon Activation Behaviors:

- **Double-click**: Shows/hides the main window
- **Single-click**: No action
- **Right-click**: Opens context menu

6.1 Tray Notifications

The program displays system tray notifications for important events:

When Window Is Hidden:

- Operation completion messages (Save/Restore/Scramble successful)
- Critical errors during background operations
- Warning messages requiring user attention

Always Displayed:

- Application started minimized to tray
- Application minimized to tray from window close
- Quick operation confirmations from tray menu

Notification Duration:

- Standard messages: 2000 milliseconds (2 seconds)
- Warning/Error messages: 5000 milliseconds (5 seconds)

7 Keyboard Shortcuts

Shortcut	Action
`Ctrl+S`	Quick Save current layout
`Ctrl+M`	Open Backup Manager
`Ctrl+H`	Toggle Show/Hide Desktop Icons
`Ctrl+,`	Open Settings menu
`Ctrl+Q`	Exit Application
`F1`	Open Online User Manual

Table 1: Available keyboard shortcuts

8 Help Menu

8.1 Online User Manual

Opens the complete PDF user manual in your default browser.

URL: https://mapi68.github.io/desktop-icon-backup-manager/manual.pdf

Manual Contents:

- Complete feature documentation
- Step-by-step tutorials
- Troubleshooting guides
- Best practices and recommendations
- Technical reference information

8.2 Check for Updates

Opens the update checker dialog (also triggered automatically at startup if the option is enabled in Settings).

Dialog contents:

- **Installed version**: Read from the bundled `version.txt`
- **Latest version**: Fetched in a background thread from GitHub
- **Status**: "You are using the latest version", "A new version is available!" (with Download button), or "You are using a pre-release version"
- **Download Update**: Opens the GitHub releases page in your browser
- **Check Again**: Re-fetches the remote version without closing

8.3 Statistics Dashboard

Opens a dashboard dialog showing aggregated statistics about your backup history. All data is computed in real time from the backup files and persistent counters stored in `settings.ini`.

Top stat cards:

- **Total backups**: Live count of all `.json` files currently present in the `icon_backups` folder
- **Restores**: Lifetime count of all restore operations
- **Saves**: Lifetime count of all save operations (persisted in `settings.ini`)
- **Disk usage**: Total size of all `.json` files in the `icon_backups` folder
- **Avg icons**: Average number of desktop icons across all saved backups

Backups per month: A bar chart showing how many backups were created in each of the last six months. The tallest bar is highlighted in a darker shade.

Top resolutions: A ranked list of the most frequently used screen resolutions across all backups, shown as horizontal progress bars with percentages.

Most moved icons: A ranked list (gold / silver / bronze badges) of the desktop icons that changed position most often between consecutive backups. An icon is counted as "moved" when its coordinates shift by more than 4 px in either axis. Requires at least two backups to produce results.

Activity: Quick reference showing the date of the first and last backup, the total number of save and scramble operations, and the most frequently used tag/description.

Note: The total backup count is computed live from the `icon_backups` folder, so it stays accurate even after importing, exporting, or manually deleting files. The restore, save, and scramble counters are stored in `settings.ini` under the `[Statistics]` section because those operations leave no trace in the backup files.

8.4 About Dialog

Displays program information including:

- Program name and description
- Current version number
- Developer information

9 Advanced Backup Management

9.1 Backup Manager Window

The Backup Manager provides a professional interface for managing all saved backups with advanced features. The window is **resizable** to suit any screen size or DPI setting.

9.1.1 Search and Filter Bar

Location: Top of the window, immediately below the instruction label

Functionality:

- Real-time filtering as you type

- Case-insensitive search
- Searches across all displayed information: tag, resolution, icon count, timestamp
- Clear button (X) to quickly reset filter
- Placeholder text: "Search by tag, resolution, or date..."

Search Examples:

- Type "1920" → Shows only backups saved at 1920×* resolution
- Type "work" → Shows backups with "work" in their tag/description
- Type "2024/12" → Shows backups from December 2024
- Type "15" → Shows backups with 15 icons or saved on the 15th day

9.1.2 Backup Table Columns

The list is displayed in a **sortable QTableWidget** — click any column header to sort ascending or descending:

1. **Tag / Description**: User-provided description (stretches to fill available space). Each distinct tag is assigned a unique colour, displayed as a vertical bar on the left edge of the cell — the same tag always gets the same colour, making layouts instantly recognisable without reading the text.
2. **Resolution**: Monitor resolution at save time (e.g. "1920x1080")
3. **Icons**: Number of icons in the backup, right-aligned
4. **Timestamp**: Human-readable date/time in format "YYYY/MM/DD HH:MM:SS"

9.1.3 Live Diff Preview Panel

Location: Right side of the window, above the info panel.

New in this version: When you select a backup, the preview automatically fetches the *current* live icon positions from the desktop and overlays them against the saved positions, giving you a visual diff *before* you commit to restoring.

Aspect-ratio preserved canvas: The preview draws a virtual screen rectangle that preserves the aspect ratio of the saved resolution (letterboxed inside the widget with empty bands on the sides when needed). This ensures icon positions in the preview reflect their real relative position on the desktop, with no horizontal or vertical squashing.

Windows 11 taskbar mock-up: The bottom of the preview canvas shows a stylised Windows 11 taskbar, containing a centered cluster of Start, Search and Task view icons, plus a right-side system tray with Wi-Fi, volume and battery glyphs. The tray clock displays the *backup's own timestamp* — time on top (HH:MM) and date below (DD/MM/YYYY) — so you can see at a glance when the snapshot was taken. Icons whose saved y-coordinate falls inside the taskbar area are those that would be hidden behind the real Windows taskbar after restore.

The preview canvas draws a subtle grid to convey the desktop space, and each icon state is rendered with a distinct visual treatment:

- **Blue dot (with soft halo)** — Icon is already in the correct saved position; restoring will not move it.

- **Orange dot** $\xrightarrow{\text{dashed}}$ **Red dot** — Icon will move. The orange dot marks the *current* position; the red dot marks the *target* position from the backup. A dashed arrow connects them to show direction.
- **Green dot** — Icon exists in the backup but is not currently on the desktop (will be skipped during restore).

Legend: A compact 3-row panel to the right of the info panel uses custom pixel-drawn dot and arrow icons (not Unicode characters) so that every row has exactly the same height and spacing, regardless of system font or DPI. Rows are arranged with a fixed-height `QGridLayout` (26 px per row) to guarantee uniform vertical spacing.

Interactive Tooltips: Hover over any dot to see the icon name and its status (*"✓ already in place"*, *"↕ will move"*, or *"△ not on desktop"*).

Pro Tip: Live Diff

Use the Live Diff Preview to assess the impact of a restore at a glance. A mostly blue canvas means the backup already matches your current layout; many orange/red dot-pairs indicate significant movement. You can inspect and then cancel the restore dialog if needed.

9.1.4 Visual Preview Panel (standard)

In addition to the live diff, the standard preview is available via the single-backup dot-map in the information panel on the right, with interactive tooltip on hover:

Technical Implementation:

- Mouse tracking enabled on preview widget with `setMouseTracking(True)`
- Hover detection uses distance calculation: $\sqrt{(dx)^2 + (dy)^2} < 12$ pixels
- Icon names displayed via `QToolTip.showText()` at cursor position
- Automatic tooltip hiding when cursor leaves icon proximity

9.1.5 Information Panel

Displays detailed metadata for the selected backup:

```
File: 1920x1080_20241211_143015.json
Icons: 12
Resolution: 1920x1080
Description: Work Setup Final
Timestamp: 2024-12-11T14:30:15
```

9.1.6 Available Operations

Inline Tag Editing:

- Double-click a cell in column 1 (Tag / Description) to activate an in-place editor
- Press `Enter` or click outside the cell to confirm and save the new tag to the `.json` file
- Press `Escape` to cancel without saving
- A tooltip on the column header reads: *"Double-click to edit the tag/description"*
- Double-clicking columns 2–4 (Resolution, Icons, Timestamp) opens the restore dialog — those columns are read-only

- If the file cannot be written, an error dialog appears and the cell reverts to its previous text automatically

Restore Operations:

- Double-click any backup in the list
- Select backup + click "Restore Selected Layout" button
- Right-click backup → "Restore Selected" in context menu

Delete Operations:

- Right-click backup → "Delete Selected"
- Confirmation dialog shown before deletion
- Successful deletion refreshes the list automatically

Selection Behavior:

- Single-selection mode (one backup at a time)
- Selection changes update preview and info panels immediately
- "Restore Selected Layout" button disabled when no valid selection
- Keyboard navigation supported (arrow keys, Enter to restore)

9.2 Backup Comparison

The application includes an intelligent comparison system that lets you compare **any two backups** from the list, visualizing structural changes and icon movements between two different points in time.

9.2.1 How to Use Comparison

Method 1 — Compare with Latest (context menu):

1. Open the Backup Manager
2. Select any backup from the list (other than the latest)
3. Right-click → *Compare with Latest*
4. A color-coded report appears in the comparison dialog

Method 2 — Compare any two backups (button):

1. Open the Backup Manager
2. Select the **first** backup from the table
3. Click the "**Compare Two Selected...**" button at the bottom
4. A mini-picker dialog opens showing all other available backups
5. Select the **second** backup and click Compare
6. A color-coded report appears

9.2.2 Comparison Engine Details

The comparison engine (`BackupComparator.compare()`) performs several automated checks between the two files:

- **Icon Tracking**: Identifies icons added in the latest version (+), icons present only in the old backup (-), and common icons using set operations.
- **Position Audit**: For common icons, it detects even pixel-level differences in coordinates between the two save states.
- **Color-Coded Reporting**: Generates a report where differences are marked in green (+) for new additions, red (-) for missing items, and yellow (↔) for moved icons.

Version Audit

This tool is particularly useful for auditing changes after a system update or a resolution change. Even if the two backups were taken at different resolutions, the *Adaptive Scaling* logic will normalize the coordinates to show only "real" icon movements.

9.2.3 Technical Details

1. **Set Operations**:
   - Added icons: `names2 - names1` (set difference)
   - Removed icons: `names1 - names2`
   - Common icons: `names1 & names2` (intersection)
2. **Position Comparison**:
   - For each common icon, compares `pos1 != pos2`
   - Moved icons list generated from position mismatches
3. **Report Generation**:
   - Color-coded HTML output via `_colorize_comparison_report()`
   - Green (#4ec9b0) for added icons (+ prefix)
   - Red (#f48771) for removed icons (- prefix)
   - Yellow (#dcdcaa) for moved icons (↔ prefix)

Example Comparison Output:

```
=== COMPARISON RESULTS ===

Icon(s) Added: 2
Icon(s) Removed: 1
Icon(s) Moved: 3
Icon(s) Unchanged: 8

--- ADDED ICONS ---
  + New Shortcut.lnk
  + Project Folder

--- REMOVED ICONS ---
  - Old File.txt
```

```
--- MOVED ICONS ---
  <-> Documents
  <-> Downloads
  <-> Pictures
```

Safety Verification

If the "Enable Adaptive Scaling" setting is active, the system will automatically calculate the scaling factors if it detects a resolution mismatch during the comparison phase.

9.3 Backup File Format

Backups are stored as human-readable JSON files in the `icon_backups` subfolder.

9.3.1 File Naming Convention

```
[Resolution]_[Date]_[Time].json

Pattern: {width}x{height}_{YYYYMMDD}_{HHMMSS}.json
Example: 1920x1080_20241211_143015.json
```

9.3.2 Filename Parsing

The application uses a strict regex parser for backup identification:

- **Format Recognition**: The expected filename pattern is `Resolution_Date_Time.json`. Filenames that do not match this exact pattern are treated as malformed and flagged as "N/A" in the backup list.
- **Fail-Fast Validation**: Calendrically impossible dates (e.g. `20240230`) are also rejected, so only real, parseable timestamps are accepted.

9.3.3 JSON File Structure


```
{
    "timestamp": "2024-12-11T14:30:15.123456",
    "icon_count": 12,
    "description": "Work Setup Final",
    "display_metadata": {
        "monitor_count": 2,
        "primary_resolution": "1920x1080",
        "screens": [
            {
                "id": 0,
                "name": "\\.\DISPLAY1",
                "width": 1920,
                "height": 1080,
                "pixel_density": 1.0
            },
            {
                "id": 1,
                "name": "\\.\DISPLAY2",
                "width": 1920,
```


buy me a coffee

```
                "height": 1080,
                "pixel_density": 1.0
            }
        ]
    },
    "icons": {
        "This PC": [100, 200],
        "Recycle Bin": [100, 350],
        "Documents": [100, 500],
        "Downloads": [100, 650],
        "Pictures": [100, 800],
        "Music": [100, 950],
        "Videos": [250, 200],
        "Desktop": [250, 350],
        "Network": [250, 500],
        "Control Panel": [250, 650],
        "User Files": [250, 800],
        "Shortcuts": [250, 950]
    }
}
```


10 Command Line Interface (CLI)

The application supports advanced automation through CLI arguments. When `--backup`, `--restore`, or `--silent` is used, the program enters headless mode: the graphical user interface is not initialized, and the process exits immediately after the operation completes. Using `--silent` alone (without `--backup` or `--restore`) simply exits with code 0.

10.1 Available Arguments

`-backup` Creates a new backup of the current desktop icon layout and exits.

`-restore <filename|latest>` Restores the specified backup file. Use `latest` to automatically pick the most recent save.

`-silent` Runs the operation without displaying any GUI message boxes. Useful for scheduled tasks and scripting.

Example CLI Usage

```
desktop-icon-backup-manager.exe --backup --silent
desktop-icon-backup-manager.exe --restore latest --silent
desktop-icon-backup-manager.exe --restore "1920x1080_20241211_143015.json"
--silent
```

CLI Backup Tag

Backups created via `-backup` are automatically tagged *"Silent CLI Backup"* in the backup file's description field.

10.2 Technical Implementation

The CLI integration is fully compatible with the existing infrastructure:

- **Configuration Inheritance**: CLI mode automatically loads `settings.ini`, ensuring features like *Adaptive Scaling* and *Cleanup Limits* are applied even without a GUI.
- **Exit Codes**:
  - `0`: Operation completed successfully.
  - `1`: An error occurred (e.g., file not found or permission denied).

10.3 Example Batch Script

You can create a `.bat` file to automate your backup on Windows startup or via Task Scheduler:

```
@echo off
start "" "desktop-icon-backup-manager.exe" --backup --silent
```

10.4 Automation with Task Scheduler

You can create scheduled automatic backups using Windows Task Scheduler:

1. Open Task Scheduler
2. Create Basic Task
3. Name: "Desktop Icon Backup - Daily"
4. Trigger: Daily at preferred time
5. Action: Start a program
6. Program: Full path to `desktop-icon-backup-manager.exe`
7. Enable: "Run whether user is logged on or not"

Important Notes:

- Use `-silent` flag to suppress all GUI dialogs
- Task runs in background if *Start Minimized* is enabled
- Cleanup limits configured in `settings.ini` are respected
- The `icon_backups` folder is always created next to the executable, regardless of the working directory set by the scheduler

10.5 Batch Operations with JSON

For advanced users comfortable with JSON editing, backup files can be manipulated directly.

Merge Multiple Backups:

1. Open two backup JSON files
2. Copy icon entries from one "icons" section to another
3. Ensure no duplicate icon names (second will override first)
4. Save merged file with new name
5. Restore merged backup

Manual Position Adjustment:

1. Open backup JSON in text editor
2. Locate icon name in "icons" section
3. Modify [X, Y] coordinates (e.g., [100, 200])
4. Save file
5. Restore modified backup

Bulk Position Shift:

- Write a script to add an offset to all coordinates
- Useful for moving the entire layout left/right/up/down
- Python script example available in project documentation

11 Best Practices

11.1 Optimal Backup Strategy

Recommended Configuration for Most Users

- Enable "Auto-Save on Exit" for safety net
- Set cleanup limit to 10-25 backups (balances history and space)
- Create manual backups with descriptive tags before major changes
- Use descriptive tags like "Before Windows Update", "Gaming Setup", etc.
- Test restore operation occasionally to verify backups work

11.2 Backup Naming Best Practices

Effective Tag Examples:

- **Descriptive**: "Work Setup - Dual Monitor", "Gaming - Single Screen"
- **Temporal**: "Before Windows Update 2024-12", "End of Year Layout"
- **Purpose**: "Clean Desktop Minimal", "Development Environment"
- **Event-based**: "Pre-Hardware Upgrade", "After Monitor Replacement"

Avoid:

- Generic tags: "backup1", "test", "new"
- Too long descriptions (truncated after 52 characters in the list view)
- Special characters that might cause issues: \/ : * ? " < > |

Colour-coded tag bars

The Backup Manager assigns a unique colour bar to each distinct tag. Using consistent tag names — for example, always “Work” rather than “Work Setup”, “work”, or “Work Layout” — ensures that all related backups share the same colour, making them immediately recognisable in the table without reading the full text.

11.3 Multi-Monitor Management

For systems with changing monitor configurations:

1. Create separate backups for each configuration with clear tags
2. Tag examples: "1 Monitor - Laptop Only", "2 Monitors - Office Setup", "3 Monitors - Home Studio"
3. Disable auto-restore if frequently switching configurations
4. Always verify current monitor setup before restoring
5. Use Backup Manager preview to visually confirm layout
6. Consider creating backup before connecting/disconnecting monitors

Docking Station Workflows:

- **Backup "Laptop Undocked"**: Before connecting to dock
- **Backup "Laptop Docked"**: After connecting monitors
- Use Backup Manager to quickly switch between configurations
- Consider enabling adaptive scaling for similar resolutions

11.4 Resolution Change Scenarios

When Upgrading Monitor:

1. Create backup with descriptive tag before upgrade
2. Note old resolution in tag: "Old Monitor 1920x1080"
3. After upgrade, test restore with scaling enabled
4. Adjust manually if needed, then save new backup
5. Keep old backup for reference or if reverting

When Using Laptop with Different External Monitors:

- Create backup for each location: "Home Office", "Work Office", "Portable"
- Include resolution in tag if significantly different
- Test adaptive scaling effectiveness for each pair
- Keep laptop-only backup as fallback

11.5 Regular Maintenance

Weekly:

- Review recent backups in Backup Manager
- Delete any test or unnecessary backups
- Verify auto-save/restore working as expected

Monthly:

- Check `icon_backups` folder size
- Export important backups to external location

- Test restore operation to verify functionality
- Review and update cleanup limit if needed

Before Major Events:

- Windows Updates: Create manual backup with tag
- Hardware changes: Backup before and after
- System reinstall: Export all backups to safe location
- Monitor changes: Backup both before and after

11.6 Backup Archiving Strategy

Export Important Backups (via `File > Export Backups...` or the Backup Manager toolbar):

1. Select one backup or choose to export all, then pick ZIP or folder
2. Store the ZIP in cloud storage (Dropbox, OneDrive, Google Drive)
3. Keep a copy on a USB drive for critical configurations
4. Include in your regular system backup routine

Import on a New Machine (via `File > Import Backups...` or the Backup Manager toolbar):

1. Copy the exported ZIP or JSON files to the new machine
2. Open the Backup Manager and click `Import Backups...`
3. Select the ZIP or individual JSON files; duplicates are skipped automatically
4. The table refreshes and the imported layouts are immediately available

When to Export:

- Perfect layouts you want to preserve forever
- Before system migrations or reinstalls
- Company/work standardised layouts to share with colleagues
- Configurations that took significant time to create

12 Troubleshooting

12.1 System Interaction and Permissions

Technical Requirements

- **Memory Access**: The program uses `OpenProcess` and `VirtualAllocEx` to interact with `Explorer.exe` memory.
- **ListView Visibility**: If icons are hidden, the program cannot access positions, resulting in the *"Unable to find desktop ListView control"* error.

buy me a coffee

12.2 Common Issues

12.2.1 Error: "Unable to find desktop ListView control"

Cause: Desktop icons are hidden or the ListView control is not accessible.

Solution:

1. Right-click on empty area of desktop
2. Navigate to `View > Show desktop icons`
3. Verify checkmark appears next to "Show desktop icons"
4. Restart the program
5. If error persists, check Windows Explorer is running

Advanced Troubleshooting:

- Restart Windows Explorer via Task Manager
- Check if third-party desktop replacement software is interfering
- Verify no group policies are hiding desktop icons
- Run program as administrator (if standard permissions fail)

12.2.2 Icons Restored to Wrong Positions

Cause: Resolution change, different monitor configuration, or disabled scaling.

Solutions:

1. Enable "Enable Adaptive Scaling on Restore" in Settings menu
2. Create separate backups for each monitor configuration with descriptive tags
3. Verify current resolution matches saved resolution in backup details
4. Check backup was created on same physical monitor setup
5. Use Backup Manager preview to verify layout before restoring

Multi-Monitor Specific Issues:

- Ensure same monitors are connected in same positions
- Check Windows display arrangement matches saved configuration
- Primary monitor must be same as when backup was created
- Monitor orientation (landscape/portrait) must match

12.2.3 Backup File Not Found

Cause: Corrupted or deleted backup file, missing `icon_backups` folder, file permissions.

Solution:

1. Verify `icon_backups` folder exists in program directory
2. Check folder contains .json files
3. Verify read/write permissions on folder and files

4. Open JSON file in text editor to check for corruption
5. Restore from Windows backup if available
6. Check Recycle Bin for accidentally deleted backups

JSON Validation:

- Open file in text editor (Notepad++, VS Code)
- Verify valid JSON format (matching braces, proper quotes)
- Check file is not empty (minimum valid backup is ∼100 bytes)
- Use online JSON validator if unsure about format

12.2.4 Program Crashes on Startup

Possible Causes and Solutions:

Corrupted settings.ini:

1. Delete `settings.ini` file from program directory
2. Restart program (will recreate with defaults)
3. Reconfigure your preferences

Auto-Restore failure:

1. Disable "Auto-Restore on Startup" by editing settings.ini manually
2. Set `auto_restore_on_startup=false`
3. Restart program and investigate problematic backup

Missing dependencies (source installation):

1. Reinstall PyQt6: `pip install -upgrade PyQt6`
2. Reinstall pywin32: `pip install -upgrade pywin32`
3. Check Python version is 3.8 or higher

12.2.5 Scramble Operation Fails

Cause: Pre-scramble backup failure, insufficient permissions, or desktop access issues.

Solution:

1. Check Activity Log for specific error message
2. Verify sufficient disk space for backup creation
3. Ensure `icon_backups` folder is writable
4. Try manual backup first to test functionality
5. Close any programs that might lock desktop access

12.2.6 Settings Not Persisting

Cause: Write permissions, settings.ini location, or application not closing properly.

Solution:

1. Check settings.ini exists in application directory
2. Verify write permissions on settings.ini file
3. Use `File > Exit` or Ctrl+Q instead of Task Manager
4. Don't force-close during "Saving settings..." operations
5. Run as administrator if in protected directory

12.3 Diagnostics

For Advanced Diagnostics:

1. Execute the problematic operation
2. Copy complete log content (Ctrl+A in log area, Ctrl+C)
3. Check for specific error messages
4. Note exact timestamp of error
5. Verify settings.ini for incorrect configurations
6. Check Windows Event Viewer for system-level errors

12.4 Performance Issues

12.4.1 Slow Save/Restore Operations

Normal Expected Times:

- Save operation: 1-3 seconds for 10-20 icons
- Restore operation: 2-5 seconds for 10-20 icons
- Scramble operation: 3-6 seconds total (includes backup)

If Operations Are Slower:

- Check CPU usage (Task Manager) - high usage indicates issue
- Verify disk is not at 100% usage (slow HDD can cause delays)
- Scan for malware/viruses affecting system performance
- Close resource-intensive applications during operations
- Consider SSD upgrade if using mechanical hard drive

12.4.2 High Memory Usage

Normal Memory Usage:

- Idle: 50-80 MB
- During operation: 80-120 MB
- With large backup list: 100-150 MB

If Memory Is Excessive:

- Close and reopen program (clears accumulated memory)
- Check for memory leaks (usage grows over time without activity)
- Report issue with version number and system specs

13 Frequently Asked Questions (FAQ)

13.1 General Questions

Q: Does the program work with Windows 11?

A: Yes, fully compatible with Windows 11 and all versions back to Windows 7. The program uses standard Windows APIs that work across all versions.

Q: Can I use the program with virtual desktops?

A: Windows virtual desktops share the same physical desktop icon layout. Backups include all visible icons regardless of which virtual desktop is active.

Q: Are backups portable between different computers?

A: Yes, JSON files can be copied between computers, but positions may not be accurate if:

- Screen resolutions differ (use adaptive scaling)
- Monitor configurations differ (different number or arrangement)
- Some icons in backup don't exist on target system

Q: How much disk space do backups use?

A: Each backup typically occupies 2-10 KB depending on icon count. With 50 backups, total usage is generally less than 500 KB (0.5 MB).

Q: Can I manually edit backup files?

A: Yes, they are standard JSON text files. You can edit with any text editor. Be careful with JSON syntax to avoid corruption. Always keep backup copies before editing.

13.2 Feature Questions

Q: Why create backup before scramble?

A: This mandatory backup ensures you can always restore your original layout. Without it, scrambling would be destructive with no undo option.

Q: What happens to icons not in backup during restore?

A: Icons currently on desktop but not in backup remain where they are. Only icons that exist in both the backup and current desktop are moved.

Q: Does adaptive scaling work with multi-monitor setups?

A: Scaling uses primary monitor resolution. For multi-monitor setups, it's recommended to create separate backups for each specific configuration rather than relying on scaling.

Q: Can I run multiple instances?

A: No, running multiple instances simultaneously is not supported and may cause conflicts. Only one instance should run at a time.

Q: Does the program check for updates?

A: Yes. By default, the program checks GitHub for a new version 10 seconds after startup. If one is available, a tray notification appears and a log entry is added. You can also check manually via `Help > Check for Updates...`. Clicking "Download Update" opens the GitHub releases page. The check can be disabled via `Settings > Check for Updates on Startup`. The program never auto-installs updates.

13.3 Technical Questions

Q: Why does the program need Windows API access?

A: Desktop icon positions are managed by Windows Explorer's ListView control. The program must use Win32 API calls to read and modify these positions, as there's no higher-level interface available.

Q: Is administrator access required?

A: No, the program runs with standard user permissions. It only accesses your own user desktop, not system-wide or other user desktops.

Q: Can antivirus software interfere?

A: Some aggressive antivirus programs may flag the memory access operations as suspicious. Add the program to your antivirus whitelist if you encounter issues.

Q: Why monospaced font in Backup Manager?

A: Monospaced fonts (Consolas) ensure perfect column alignment without complex UI frameworks, making the backup list easy to scan and read.

Q: Can I backup icon arrangement to network drive?

A: The `icon_backups` folder must be local for proper operation. After creation, you can manually copy backups to network storage for archiving.

14 Technical Information

14.1 Software Architecture

Core Components:

- **PyQt6 Framework**: Modern cross-platform GUI framework providing widgets, layouts, and event handling
- **pywin32 Library**: Python bindings for Windows API, enables low-level desktop access
- **QThread**: Asynchronous operations prevent UI freezing during lengthy operations
- **QSettings**: INI-based configuration persistence across sessions
- **QSystemTrayIcon**: Native system tray integration with context menus

Key Classes:

- `DesktopIconManager`: Core save/restore/scramble logic; also exposes the method used by the live diff:
  `get_current_icon_positions()`
- `BackupComparator`: Standalone module (`core/comparator.py`) for diff reports between any two backups

- `IconWorker`: QThread worker; accepts a shared `DesktopIconManager` instance to avoid redundant desktop handle detection
- `MainWindow`: Primary application window and UI orchestration
- `BackupManagerWindow`: Backup selection dialog with sortable `QTableWidget` (resizable)
- `IconPreviewWidget`: Standard single-backup dot-map preview
- `_DiffCanvas`: Custom `QWidget` that paints the live diff overlay (grid, dots, dashed arrows, arrowheads)
- `_ColorDot`: Pixel-drawn circle widget used in the legend to guarantee uniform size across all DPI settings
- `_ArrowIcon`: Pixel-drawn orange → red arrow widget that illustrates the "will move" legend entry
- `_LegendPanel`: `QFrame` with a fixed-row-height `QGridLayout` (26 px/row) ensuring perfectly uniform vertical spacing for all legend entries
- `DiffPreviewWidget`: Composes `_DiffCanvas` + `_LegendPanel` into the final panel shown in the Backup Manager

14.2 Remote Memory Access Process

To read icon positions, the program performs complex low-level operations:

Detailed Process:

1. **Find Desktop ListView**:
   - Locate "Progman" window (Program Manager)
   - Find "SHELLDLL_DefView" child window
   - Locate "SysListView32" control (icon container)
   - Fallback: Enumerate all windows if not found in standard location
2. **Access Remote Process**:
   - Get Explorer process ID via `GetWindowThreadProcessId`
   - Open process with `PROCESS_ALL_ACCESS` rights
   - Allocate 4096 bytes in remote process memory
3. **Query Icon Data**:
   - Send `LVM_GETITEMCOUNT` message for icon count
   - For each icon: Send `LVM_GETITEMPOSITION` for coordinates
   - For each icon: Send `LVM_GETITEMTEXT` for icon name
   - Read data from remote memory to local process
4. **Cleanup**:
   - Free allocated remote memory with `VirtualFreeEx`
   - Close process handle properly

Constants Used:

```
LVM_GETITEMCOUNT    = 0x1004  # Get total icon count
LVM_GETITEMTEXTW    = 0x1073  # Get icon text/name (Unicode)
LVM_GETITEMPOSITION = 0x1010  # Get icon X,Y position
LVM_SETITEMPOSITION = 0x100F  # Set icon X,Y position
LVS_AUTOARRANGE     = 0x0010  # Auto-arrange flag
MEM_COMMIT          = 0x1000  # Commit memory allocation
MEM_RELEASE         = 0x8000  # Release memory
PAGE_READWRITE      = 0x04    # Read/Write permissions
```

14.3 Data Structures

LVITEMW Structure (ctypes.Structure):

```
class LVITEMW(ctypes.Structure):
    _fields_ = [
        ("mask",       ctypes.c_uint),    # 0x0001 = LVIF_TEXT
        ("iItem",      ctypes.c_int),     # Icon index
        ("iSubItem",   ctypes.c_int),     # 0 (main item)
        ("state",      ctypes.c_uint),    # 0
        ("stateMask",  ctypes.c_uint),    # 0
        ("pszText",    ctypes.c_void_p),  # Pointer to text buffer
        ("cchTextMax", ctypes.c_int),     # 512 (max text length)
        ("iImage",     ctypes.c_int),     # Image index (unused)
        ("lParam",     ctypes.c_void_p),  # App-defined value
    ]
```

Position Data:

```
Format: Two 32-bit integers (8 bytes)
- X coordinate (signed int)
- Y coordinate (signed int)
Unpacked with: struct.unpack('ii', data)
```

14.4 Threading Architecture

Why Threading Is Essential:

- Icon scanning can take 2-5 seconds for many icons
- Without threading, UI would freeze completely
- User cannot cancel or see progress without threading
- Windows messages would not be processed during operations

Worker Thread Design:

- **log_signal**: Emits log messages back to main thread
- **progress_signal**: Emits 0-100 progress values
- **finished_signal**: Emits completion status and optional data
- Main thread updates UI in response to signals
- Worker thread performs all Win32 API calls

buy me a coffee

14.5 Desktop Refresh Mechanism

After modifying icon positions, the program forces desktop refresh using `SendMessageTimeout` to avoid blocking:

```
# Re-enable window redrawing
win32gui.SendMessage(hwnd, WM_SETREDRAW, 1, 0)

# Invalidate entire ListView area
win32gui.InvalidateRect(hwnd, None, True)

# Broadcast system-wide setting change (with timeout to avoid hang)
win32gui.SendMessageTimeout(
    HWND_BROADCAST,
    WM_SETTINGCHANGE,
    0,
    "IconMetrics",
    SMTO_ABORTIFHUNG,
    5000
)
```

Why This Is Necessary:

- Redrawing is disabled during restore for performance
- InvalidateRect forces visual update of moved icons
- WM_SETTINGCHANGE notifies all applications
- Without this, icons may not appear moved until manual refresh

14.6 Settings Storage

The application stores all configuration in a human-readable INI file located in the program directory.

14.6.1 Settings File Location

The program determines the settings file path dynamically based on execution context:

```
app_path = Path(os.path.abspath(sys.argv[0])).parent
settings_file_path = app_path / "settings.ini"
```

Location Behavior:

- **Standalone Executable**: Settings saved next to the .exe file
- **Python Script**: Settings saved in the script's directory
- Format: Qt INI format (`QSettings.Format.IniFormat`)
- Encoding: UTF-8 with automatic escaping for special characters

14.6.2 Complete Settings Structure

```
[General]
start_minimized=false
auto_save_on_exit=false
auto_restore_on_startup=false
check_updates_on_startup=true
adaptive_scaling_enabled=false
close_to_tray=false
cleanup_limit=0
autohide_enabled=false
autohide_seconds=300
autohide_backup_before_hide=true
autohide_notify_enabled=true
theme_mode=system
geometry=@Rect(100 100 800 650)

[Statistics]
total_restores_performed=0
total_saves_performed=0
total_scrambles=0
```

Settings Explanation:

`start_minimized` Launch application hidden in system tray

`auto_save_on_exit` Automatically backup layout when closing

`auto_restore_on_startup` Restore latest backup on launch

`adaptive_scaling_enabled` Enable resolution scaling during restore

`close_to_tray` Minimize to tray when clicking X button

`check_updates_on_startup` Check GitHub for a new version 10 seconds after launch (default: true)

`cleanup_limit` Maximum backup files to retain (0 = unlimited)

`autohide_enabled` Enable the auto-hide desktop icons timer

`autohide_seconds` Auto-hide countdown duration in seconds (default: 300 = 5 minutes)

`autohide_backup_before_hide` Create a backup before the timer hides icons (default: true)

`autohide_notify_enabled` Show a desktop notification 1 minute and 10 seconds before the timer hides icons (default: true)

`theme_mode` Application colour scheme: `system` (follow Windows setting), `light`, or `dark` (default: `system`)

`geometry` Window position and size in Qt rect format

`total_restores_performed` Lifetime count of successful restore operations (used by Statistics Dashboard)

`total_saves_performed` Lifetime count of successful save operations (used by Statistics Dashboard)

`total_scrambles` Lifetime count of scramble operations

Manual Configuration

Advanced users can edit `settings.ini` directly with any text editor. Changes take effect on next program launch. Invalid values are automatically reset to defaults.

14.7 Security and Privacy

Privacy Commitment

- No data is ever sent online or to external servers
- All backups are stored locally on your computer
- No telemetry, analytics, or usage tracking
- No network connections except the optional update check (connects only to GitHub to read `version.txt`; disabled via Settings)
- Access only to your own desktop (not other users)
- Source code is available for inspection

Data Stored:

- Icon names and positions only
- Screen resolution metadata
- User-provided descriptions/tags
- Application preferences (local settings.ini)

No Sensitive Data Collected:

- No file contents or documents
- No passwords or credentials
- No browsing history or personal information
- No system information beyond screen resolution

15 Development and Contribution

15.1 Building from Source

Requirements:

```
Python 3.8+
```

Setup:

```
git clone https://github.com/mapi68/desktop-icon-backup-manager.git
cd desktop-icon-backup-manager
pip install -r requirements.txt
python main.py
```

Building Executable with PyInstaller:

```
pip install pyinstaller
pyinstaller --onefile --windowed --icon=icon.ico \
    --name="desktop-icon-backup-manager" \
    main.py
```

15.2 Translation/Localization

The program uses Qt's translation system for internationalization. The language is auto-detected from the Windows locale and can be manually overridden.

16 Screenshots

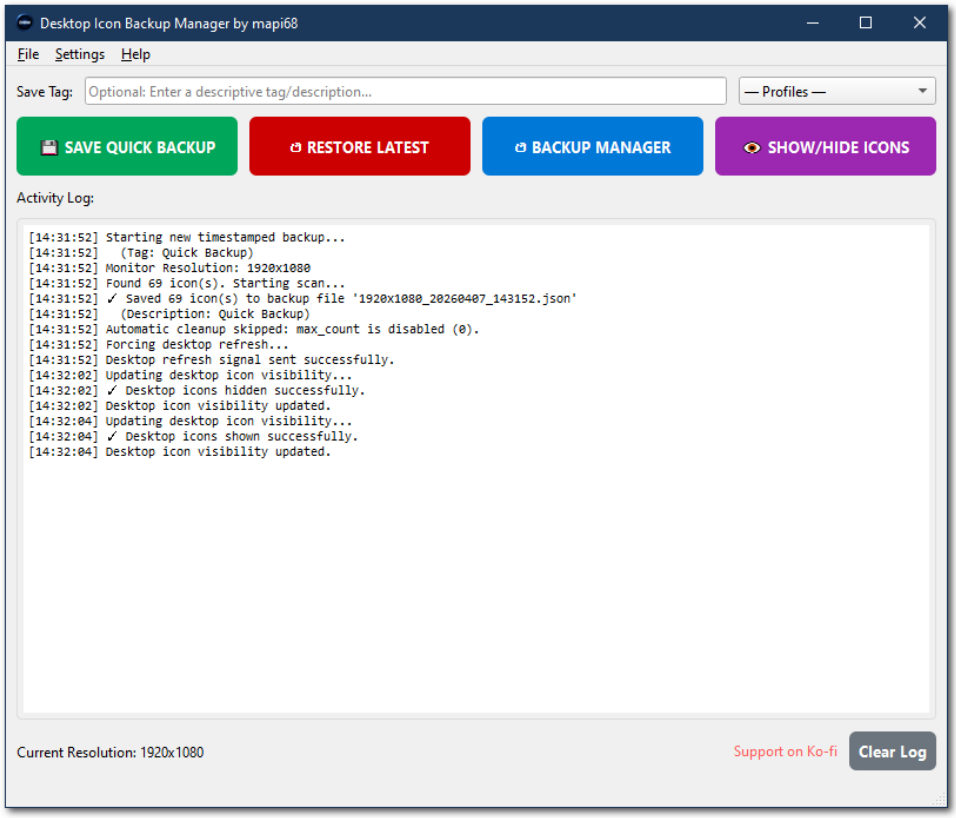

Figure 1: Main interface showing the activity log and three main action buttons

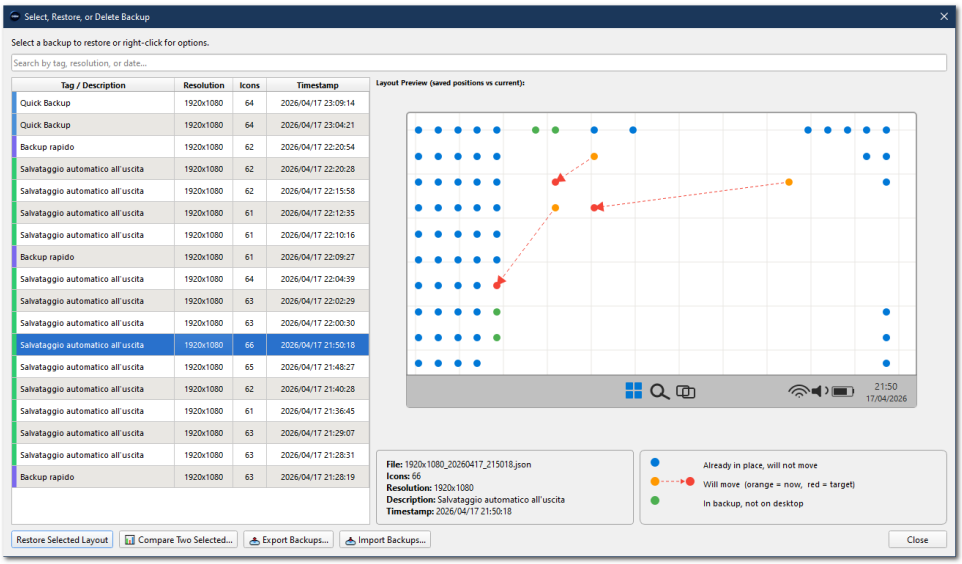

Figure 2: Backup Manager window with list of saved backups and layout preview

buy me a coffee

Figure 3: Loading Desktop Icon Backup Manager...

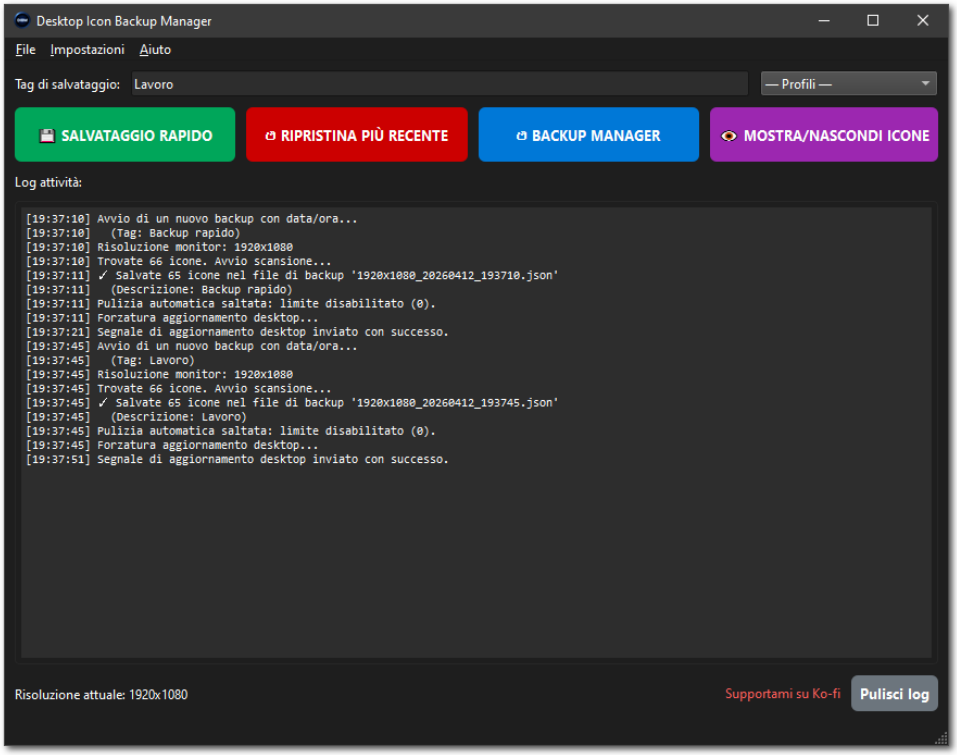

Figure 4: Desktop Icon Backup Manager featuring dark mode and Italian support

buy me a coffee

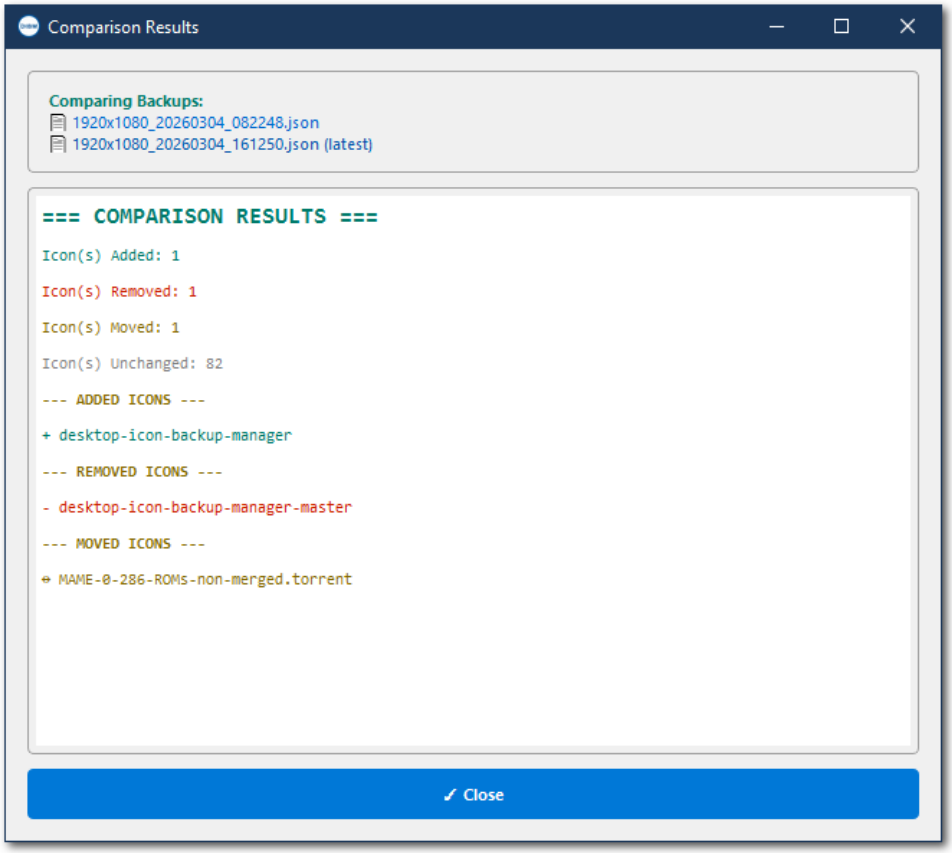

Figure 5: Detailed view of the comparison interface

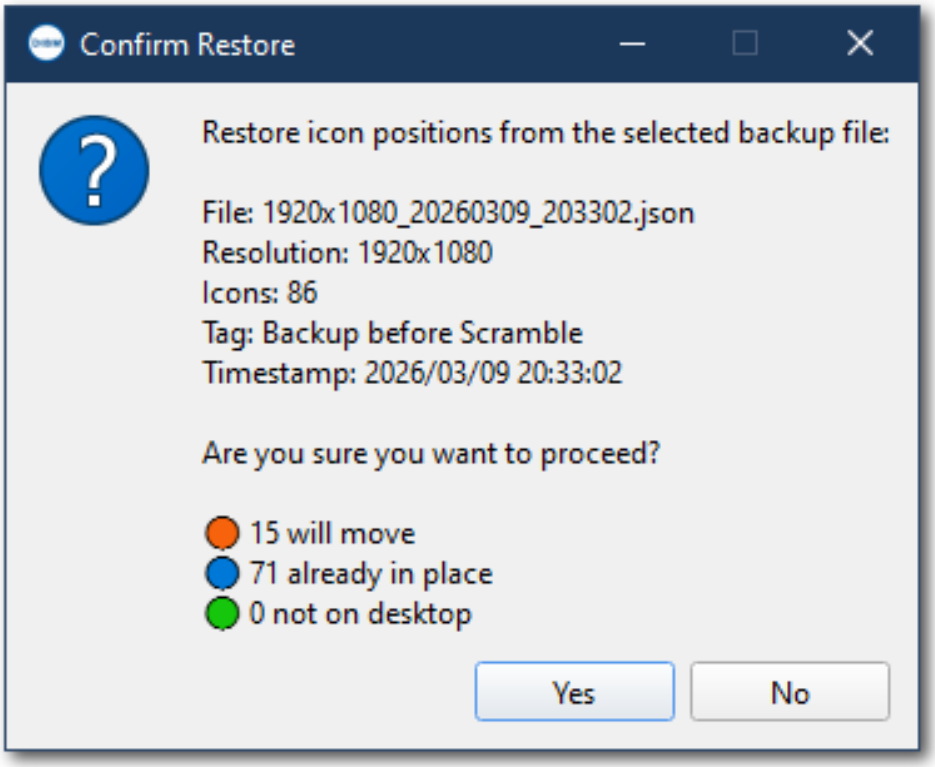

Figure 6: Confirmation dialog before restoring a backup

buy me a coffee

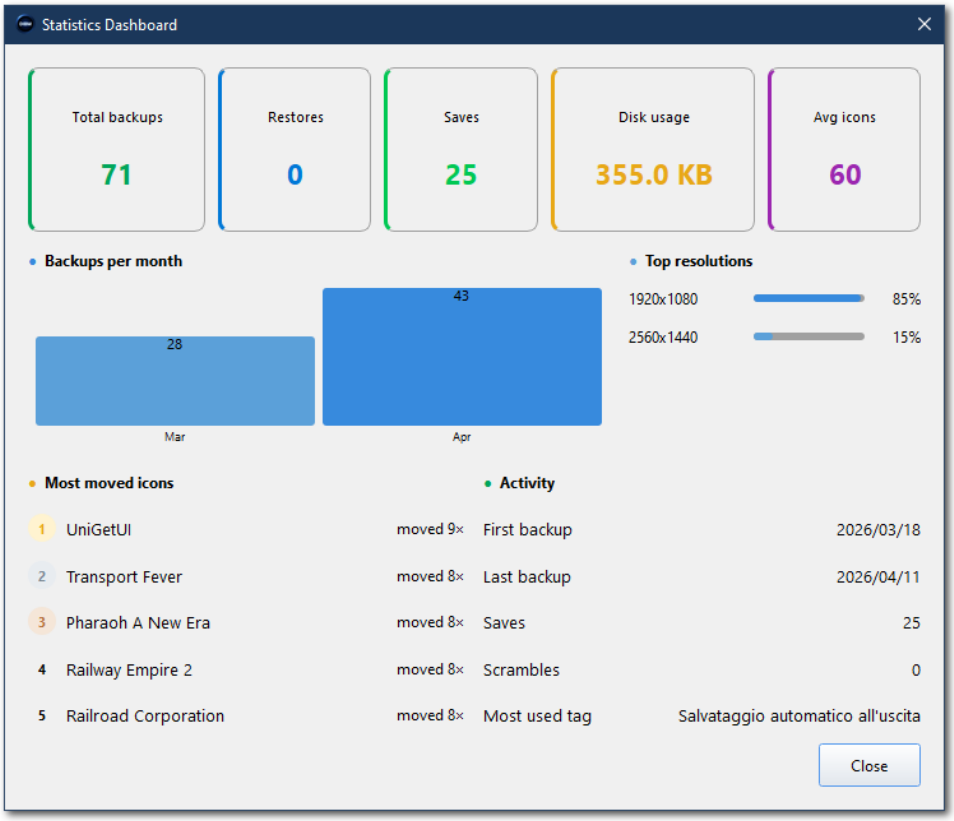

Figure 7: Statistics Dashboard showing backup history, resolutions, and most moved icons

Support the Developer

Desktop Icon Backup Manager is free and open-source.

If you find it useful, consider buying me a coffee!

---

ko-fi.com/mapi68

https://ko-fi.com/mapi68

Source code & updates:

github.com/mapi68/desktop-icon-backup-manager

Copyright © 2026 **mapi68** • July 24, 2026