

USER GUIDE

# PowerToys User Guide

Installation, Usage & Quick Reference

*A Cyber Panda Solutions LLC product*

---

# About This Document

## What Is This?

This is a comprehensive, source-code-verified **User Guide** for **Powertoys**, generated by DocCompiler.ai using Claude Opus 4.6 with 1M-token context. Every command, configuration option, and behavior documented here was verified against the actual source code.

## The Definitive Reference

This document is designed to be the single source of truth for Powertoys. Rather than piecing together scattered README fragments, outdated wiki pages, forum posts, and version-specific screenshots, you can rely on this as a verified, structured reference backed by automated code review.

## Optimized for AI Assistants

This document is structured for consumption by AI assistants. Feed this PDF into your preferred LLM for accurate, grounded answers about Powertoys — instead of having the model guess or hallucinate from incomplete context.

## Version & Updates

Generated on **2026-05-07** (version **3.1**). This document reflects the source code at commit `fde1599`. The always-updated version is permanently available at [doccompiler.ai/microsoft/PowerToys](https://doccompiler.ai/microsoft/PowerToys). Do not print or download for offline use — this document is updated when the source code changes.

## Our Mission

DocCompiler.ai aims to be the definitive wiki for the world's most important open-source projects. Every document is overkill by design: exhaustive, fact-checked, and built to replace the scattered tribal knowledge that slows teams down.

---

DocCompiler.ai is a product of Cyber Panda Solutions LLC. All documents are AI-generated from source code analysis and should be reviewed before use in production environments. Questions or corrections: [jesse.green@doccompiler.ai](mailto:jesse.green@doccompiler.ai)

# Table of Contents

## Microsoft PowerToys -- User Guide

1. Quick Start
  - Default Shortcuts at a Glance
2. Installation
  - 2.1 System Requirements
  - 2.2 Installation Methods
  - 2.3 First Launch
  - 2.4 Updating
  - 2.5 Uninstallation
3. Configuration
  - 3.1 General Settings
  - 3.2 Per-Module Settings
  - 3.3 Settings File Locations
  - 3.4 Backup and Restore
  - 3.5 Enterprise Management (GPO)
4. Core Usage
  - 4.1 Advanced Paste
  - 4.2 Always on Top
  - 4.3 Awake
  - 4.4 Color Picker
  - 4.5 Command Not Found
  - 4.6 Command Palette (CmdPal)
  - 4.7 Crop and Lock
  - 4.8 Cursor Wrap
  - 4.9 Environment Variables
  - 4.10 FancyZones
  - 4.11 File Explorer Add-ons
  - 4.12 File Locksmith
  - 4.13 Hosts File Editor
  - 4.14 Image Resizer
  - 4.15 Keyboard Manager
  - 4.16 Light Switch
  - 4.17 Mouse Utilities
  - 4.18 Mouse Without Borders
  - 4.19 New+
  - 4.20 Peek
  - 4.21 Power Display
  - 4.22 Power Rename
  - 4.23 PowerToys Run
  - 4.24 Quick Accent

- 4.25 Registry Preview
- 4.26 Screen Ruler
- 4.27 Shortcut Guide
- 4.28 Text Extractor (OCR)
- 4.29 Workspaces
- 4.30 ZoomIt
- 5. Common Tasks
  - 5.1 Setting Up a Multi-Monitor Workspace
  - 5.2 Batch Processing Files
  - 5.3 Presentation Mode
  - 5.4 Developer Productivity
  - 5.5 Extracting Text from Images and Screenshots
  - 5.6 Taming Keyboard Shortcuts
  - 5.7 Working with Colors (Design Workflows)
  - 5.8 Creating a Custom File Template Library
  - 5.9 Investigating Locked Files
  - 5.10 Keeping a Script Running Without Sleep
- 6. Troubleshooting
  - 6.1 PowerToys Won't Start
  - 6.2 Module Not Working After Update
  - 6.3 Keyboard Manager Remapping Errors
  - 6.4 Command Not Found Not Working
  - 6.5 FancyZones Windows Not Snapping
  - 6.6 Mouse Pointer Crosshairs Not Tracking
  - 6.7 Find My Mouse Activating Unintentionally
  - 6.8 Mouse Without Borders Connection Issues
  - 6.9 Awake Not Keeping System Awake
  - 6.10 Image Resizer DLL Load Failure
  - 6.11 Cannot Disable Mouse Utilities
  - 6.12 PowerToys Run Not Finding Applications
  - 6.13 Color Picker Not Activating
  - 6.14 Advanced Paste AI Features Not Working
  - 6.15 Settings File Corruption
  - 6.16 Shell Extensions Not Appearing in Context Menu
  - 6.17 Text Extractor Returns Incorrect Text
  - 6.18 Elevated Application Compatibility
  - 6.19 High Memory or CPU Usage
  - 6.20 Workspaces Not Restoring Correctly
  - 6.21 Quick Accent Not Showing Characters
  - 6.22 FancyZones Layout Editor Won't Open
  - 6.23 Peek Preview Not Rendering
  - 6.24 New+ Templates Not Appearing
  - 6.25 Registry Preview Not Opening .reg Files

## 7. FAQ

[General](#)

[Module-Specific](#)

# Microsoft PowerToys -- User Guide

Microsoft PowerToys is a free, open-source collection of over 30 utilities for Windows power users, developed by Microsoft under the MIT license. It runs as a background service (PowerToys Runner) that loads individual utility modules on demand. This guide covers everything you need to install, configure, and use every PowerToys module effectively.

## 1. Quick Start

Get productive with PowerToys in under five minutes:

1. **Install** via WinGet (recommended):

```
powershell
```

```
winget install Microsoft.PowerToys --source winget
```

2. **Launch** PowerToys from the Start menu -- it starts minimized to the system tray.
3. **Try these immediately:**
  - Press **Alt+Space** to open PowerToys Run (quick launcher)
  - Press **Win+Shift+V** to open Advanced Paste
  - Hold the **Win** key for 1 second to see the Shortcut Guide overlay
  - Press **Win+Shift+T** to extract text from any screen region via OCR
4. **Access Settings** by right-clicking the PowerToys icon in the system tray and selecting **Settings**, or by searching "PowerToys" in the Start menu.

### Tip

Pin the PowerToys system tray icon so it stays visible. Right-click the taskbar, go to **Taskbar settings > Other system tray icons**, and toggle PowerToys on.

Every module can be independently enabled or disabled from the Settings dashboard. Start with the modules you need and explore others over time.

## Default Shortcuts at a Glance

Module	Default Shortcut
Advanced Paste	<a href="#">Win+Shift+V</a>
Always on Top	<a href="#">Win+Ctrl+T</a>
Color Picker	<a href="#">Ctrl+Break</a>
Command Palette	<a href="#">Win+Alt+Space</a>

Crop and Lock (thumbnail)	Win+Ctrl+Shift+T
Crop and Lock (reparent)	Win+Ctrl+Shift+R
FancyZones Editor	` Win+Shift+ ``
Mouse Jump	Win+Shift+D
Mouse Highlighter	Win+Shift+H
Mouse Crosshairs	Win+Alt+P
Find My Mouse	Double-press Ctrl
Peek	Ctrl+Space
PowerToys Run	Alt+Space
Screen Ruler	Win+Shift+M
Shortcut Guide	Hold Win
Text Extractor	Win+Shift+T
Workspaces	Win+Ctrl+W

All shortcuts are customizable in each module's settings page.

## 2. Installation

### 2.1 System Requirements

Requirement	Minimum
Operating System	Windows 10 v2004 (build 19041) or later / Windows 11
Architecture	x64 or ARM64
Runtime	.NET 8 Desktop Runtime (bundled with installer)
WebView2	Microsoft Edge WebView2 Runtime (bundled)
Disk Space	~500 MB
RAM	4 GB (8 GB recommended for AI features in Advanced Paste)

### 2.2 Installation Methods

#### WinGet (recommended)

```
powershell
```

```
winget install Microsoft.PowerToys --source winget
```

## Expected output:

```
Found Microsoft PowerToys [Microsoft.PowerToys] Version 0.89.0
This application is licensed to you by its owner.
Microsoft is not responsible for, nor does it grant any licenses to, third-party packages.
Successfully installed
```

## Microsoft Store

Search for "Microsoft PowerToys" in the Microsoft Store app, or visit the [Microsoft Store listing](#) directly.

## GitHub Releases

Download the latest `.exe` installer from [github.com/microsoft/PowerToys/releases](https://github.com/microsoft/PowerToys/releases). Choose the correct architecture:

File	Architecture
<a href="#">PowerToysSetup-x64.exe</a>	Intel/AMD 64-bit
<a href="#">PowerToysSetup-arm64.exe</a>	ARM64 (Surface Pro X, etc.)

## Chocolatey

```
powershell
choco install powertoys
```

## 2.3 First Launch

After installation, PowerToys launches automatically and places an icon in the system tray. On first run:

- The Settings window opens with the **General** tab
- All modules are enabled by default
- PowerToys is configured to run at startup via a Task Scheduler task under `\PowerToys\`

## 2.4 Updating

PowerToys checks for updates from GitHub releases every 24 hours. When an update is available:

1. A notification appears in the system tray
2. If **Download updates automatically** is enabled in General settings, the installer downloads in the background (skipped on metered connections)
3. Click **Install** when prompted, or install manually from Settings > General

```
powershell
# Update via WinGet
winget upgrade Microsoft.PowerToys --source winget
```

## 2.5 Uninstallation

```
powershell
winget uninstall Microsoft.PowerToys
```

Or use **Settings > Apps > Installed apps**, find Microsoft PowerToys, and click **Uninstall**.

#### Note

Uninstalling does not remove your settings. To fully clean up, delete the settings directory at `%LOCALAPPDATA%\Microsoft\PowerToys\` after uninstalling.

## 3. Configuration

### 3.1 General Settings

Open PowerToys Settings from the system tray icon or by pressing **Alt+Space** and typing "PowerToys Settings."

Key general settings:

Setting	Description	Default
Run at startup	Launch PowerToys when Windows starts	On
Download updates automatically	Auto-download new versions	On
Theme	Light, Dark, or Windows default	Windows default
Run as administrator	Elevate PowerToys for compatibility with admin apps	Off

### 3.2 Per-Module Settings

Each module has its own settings page accessible from the left sidebar in Settings. Common patterns across modules:

- **Enable/Disable toggle** at the top of every module page
- **Activation shortcut** customizable to any key combination
- **Excluded applications** list (by executable name) for modules that should ignore certain apps

### 3.3 Settings File Locations

Settings are stored as JSON files on disk:

File	Path
General settings	<code>%LOCALAPPDATA%\Microsoft\PowerToys\settings.json</code>
Module settings	<code>%LOCALAPPDATA%\Microsoft\PowerToys\{ModuleName}\settings.json</code>

Example -- viewing FancyZones settings:

```
powershell
```

```
Get-Content "$env:LOCALAPPDATA\Microsoft\PowerToys\FancyZones\settings.json" | ConvertFrom-Json | Format-Li
```

## 3.4 Backup and Restore

To back up all PowerToys settings:

```
powershell
```

```
$source = "$env:LOCALAPPDATA\Microsoft\PowerToys"  
$dest = "$env:USERPROFILE\Documents\PowerToys-Backup"  
Copy-Item -Path $source -Destination $dest -Recurse -Force
```

To restore, copy the backup folder back to the same location and restart PowerToys.

## 3.5 Enterprise Management (GPO)

PowerToys supports Group Policy for enterprise environments. Administrators can control module availability, update behavior, and feature flags through GPO templates. See Admin Guide for full configuration reference.

# 4. Core Usage

This section documents every PowerToys module with activation shortcuts, features, and usage examples.

## 4.1 Advanced Paste

**Shortcut:** [Win+Shift+V](#)

Advanced Paste transforms clipboard content into different formats using AI-powered conversion. It goes far beyond the standard Windows paste.

**Paste Formats:**

Format	Description
Paste as Plain Text	Strip all formatting
Paste as Markdown	Convert to Markdown syntax
Paste as JSON	Structure clipboard data as JSON
Custom AI Action	Free-form prompt to transform clipboard content

**How to use:**

1. Copy content to clipboard normally ([Ctrl+C](#))
2. Press [Win+Shift+V](#) to open the Advanced Paste window
3. Select a paste format or type a custom AI instruction
4. The transformed content is pasted into the active application

**Direct paste shortcuts** can be configured for each format to skip the UI entirely. Set these in Settings > Advanced Paste.

### Important

AI-powered features (custom actions) require an OpenAI API key configured in Settings > Advanced Paste > AI Settings. Without it, only the standard format conversions (plain text, Markdown, JSON) are available.

## 4.2 Always on Top

**Shortcut:** `Win+Ctrl+T`

Pins the currently focused window so it stays above all other windows. A colored border appears around pinned windows to indicate their status.

### Features:

- Toggle on/off with the same shortcut
- Customizable border color and thickness
- Optional sound effect on toggle
- Works with any window, including UWP apps
- Excluded apps list for windows that should never be pinned

**Usage:** Focus the window you want pinned, press `Win+Ctrl+T`. A colored border (default: blue, 3px) appears to confirm. Press `Win+Ctrl+T` again to unpin.

## 4.3 Awake

**Activation:** System tray icon (coffee cup)

Keeps your computer and/or display awake without changing Windows power settings permanently.

### Modes:

Mode	Behavior
Off (Passive)	Use Windows power settings normally
Keep awake indefinitely	Prevent sleep until manually turned off
Keep awake temporarily	Prevent sleep for a set duration
Keep awake until expiration	Prevent sleep until a specific date/time

**System tray usage:** Right-click the Awake icon (coffee cup) in the system tray to switch between modes or set a timer.

### Awake CLI

Awake includes a standalone CLI (`PowerToys.Awake.exe`) for scripting and automation:

```
powershell
# Keep awake indefinitely with display on
PowerToys.Awake.exe --display-on
powershell
```

```
# Keep awake for 2 hours (7200 seconds)
PowerToys.Awake.exe --time-limit 7200

powershell

# Keep awake until a specific time
PowerToys.Awake.exe --expire-at "2026-05-08T09:00:00"

powershell

# Bind to a process -- stop keeping awake when the process exits
PowerToys.Awake.exe --pid 12345

powershell

# Use PowerToys configuration instead of CLI flags
PowerToys.Awake.exe --use-pt-config
```

### CLI Options Reference:

Flag	Short	Description
<code>--use-pt-config</code>	<code>-c</code>	Use PowerToys settings instead of CLI flags
<code>--display-on</code>	<code>-d</code>	Keep display on (not just system)
<code>--time-limit</code>	<code>-t</code>	Duration in seconds
<code>--pid</code>	<code>-p</code>	Bind to a process ID; Awake ends when the process exits
<code>--expire-at</code>	<code>-e</code>	Expiration date/time (ISO 8601 format)
<code>--use-parent-pid</code>	<code>-u</code>	Bind to the calling process; Awake ends when the caller exits

## 4.4 Color Picker

**Shortcut:** `Ctrl+Break`

Pick any color from anywhere on screen and copy it to your clipboard in your preferred format.

### How to use:

1. Press `Ctrl+Break` to activate
2. Move your cursor over any pixel on screen -- a magnified preview follows your cursor
3. Click to pick the color
4. The Color Picker editor opens showing the color in multiple formats

**Supported formats:** HEX, RGB, HSL, HSV, CMYK, HSB, HSI, HWB, NCol, CIELAB, CIEXYZ, VEC4, Decimal, and more.

### Settings:

- Choose which format is copied to clipboard on pick
- Configure which formats appear in the editor
- Color history is preserved across sessions
- Pick behavior: editor (default) or copy and close

## 4.5 Command Not Found

**Activation:** Automatic in PowerShell 7+

When you type a command that does not exist, this module suggests WinGet packages that provide the command.

### Setup:

```
powershell  
  
# Install the module (PowerShell 7+ required)  
Install-Module -Name Microsoft.WinGet.CommandNotFound
```

```
powershell  
  
# Enable in your PowerShell profile  
Import-Module Microsoft.WinGet.CommandNotFound
```

### Example:

```
powershell  
  
PS> mongod  
mongod: The term 'mongod' is not recognized as a name of a cmdlet...  
  
Suggestion: winget install --id MongoDB.Server
```

#### Note

This module requires internet access to query WinGet package databases. It operates silently when offline. PowerShell 7+ is required -- Windows PowerShell 5.1 is not supported.

## 4.6 Command Palette (CmdPal)

**Shortcut:** [Win+Alt+Space](#)

Command Palette is the next-generation extensible launcher designed to succeed PowerToys Run. It provides a unified command interface with extension support.

### Features:

- Application launching
- File search
- System commands
- Calculator
- Web search
- Extensible via plugins/extensions
- Command history

### Usage:

1. Press [Win+Alt+Space](#)
2. Start typing to search applications, files, or commands
3. Use arrow keys to navigate results
4. Press Enter to execute

Command prefixes filter results:

Prefix	Function
>	System commands
=	Calculator
?	Web search
/	File search

## 4.7 Crop and Lock

### Shortcuts:

- `Win+Ctrl+Shift+T` -- Thumbnail mode (static crop)
- `Win+Ctrl+Shift+R` -- Reparent mode (interactive crop)

Crop and Lock creates a smaller, cropped view of an application window that floats on your desktop.

### Modes:

Mode	Shortcut	Behavior
Thumbnail	<code>Win+Ctrl+Shift+T</code>	Static screenshot crop -- does not update
Reparent	<code>Win+Ctrl+Shift+R</code>	Live interactive crop -- stays synced with source window

### How to use:

1. Focus the window you want to crop
2. Press the desired shortcut
3. Draw a rectangle over the region you want to keep
4. A new floating window appears with just that region

Reparent mode keeps the cropped region interactive -- you can click buttons and type in the cropped area. Thumbnail mode is a static snapshot.

## 4.8 Cursor Wrap

**Activation:** Configurable in Settings > Mouse Utilities

When your mouse cursor reaches the edge of one monitor, it wraps to the opposite edge of the adjacent monitor. Particularly useful for multi-monitor setups where monitors are not physically adjacent or when you want seamless cursor movement in all directions.

### Settings:

- Enable/disable per direction (horizontal, vertical)
- Works across all connected monitors

## 4.9 Environment Variables

**Activation:** Open from PowerToys Settings or search "Environment Variables" via PowerToys Run

A modern GUI editor for Windows environment variables, replacing the cramped built-in Windows editor.

### Features:

- View and edit User and System environment variables side by side
- **Profiles** -- save sets of environment variables and switch between them with one click
- Search and filter across all variables
- Duplicate detection warnings
- PATH editor with drag-and-drop reordering
- Add, edit, delete, and toggle variables without navigating nested System Properties dialogs

### Warning

Editing system environment variables requires PowerToys to run as administrator. If PowerToys is running without elevation, only user variables can be modified.

## 4.10 FancyZones

**Shortcut:** `Win+Shift+`` (backtick) to open the layout editor

FancyZones is a window layout manager that lets you define custom screen zones. Drag windows into predefined zones for consistent, productive layouts.

### How to use:

1. Press `Win+Shift+`` to open the FancyZones Editor
2. Choose a template layout (Focus, Columns, Rows, Grid, Priority Grid) or create a custom layout
3. Close the editor
4. Hold **Shift** while dragging a window to see zone overlays
5. Drop the window into a zone -- the window snaps to fill that zone

### Layout types:

Type	Description
Grid	Divide screen into rows and columns
Canvas	Free-form zone placement with overlap support
Templates	Pre-built layouts: Focus, Columns, Rows, Grid, Priority Grid

### Power features:

- Multiple layouts per monitor
- Layout hotkeys (assign number keys 0-9 to layouts for instant switching)
- Override Windows Snap (hold Shift while Snapping)
- Multi-monitor support with independent layouts per display

- Window auto-restore to last zone on app launch
- Zone highlight customization (color, opacity, border)

## FancyZones CLI

FancyZones includes a CLI tool ([PowerToys.FancyZonesCLI.exe](#)) for scripting and automation:

```
powershell
# List all layouts
PowerToys.FancyZonesCLI.exe get-layouts
```

Expected output:

```
Templates
=====
#1 Focus          (3 zones)
#2 Columns        (3 zones)
#3 Rows           (3 zones)
#4 Grid           (6 zones)
#5 Priority Grid  (3 zones)

Custom Layouts
=====
#1 My Dev Layout (4 zones) [hotkey: 1]
#2 Writing       (2 zones) [hotkey: 2]
```

```
powershell
# Show monitor info with active layouts
PowerToys.FancyZonesCLI.exe get-monitors
```

Expected output:

```
Monitor #1: 3840x2160 @ (0,0) [Primary]
  Active layout: My Dev Layout (custom)

Monitor #2: 2560x1440 @ (3840,0)
  Active layout: Columns (template)
```

```
powershell
# Show active layout per monitor
PowerToys.FancyZonesCLI.exe get-active-layout
```

```
powershell
# List layout hotkeys
PowerToys.FancyZonesCLI.exe get-hotkeys
```

```
powershell
# Assign hotkey 3 to a custom layout
PowerToys.FancyZonesCLI.exe set-hotkey --layout "My Dev Layout" --key 3
```

```
powershell
# Remove a hotkey
PowerToys.FancyZonesCLI.exe remove-hotkey --key 3
```

```
powershell
# Apply a layout to a specific monitor
PowerToys.FancyZonesCLI.exe set-layout --monitor 1 --layout "Priority Grid"
```

```
powershell
# Open the FancyZones Editor GUI
PowerToys.FancyZonesCLI.exe open-editor

powershell
# Open FancyZones settings in PowerToys
PowerToys.FancyZonesCLI.exe open-settings
```

### CLI Command Reference:

Command	Alias	Description
<code>get-active-layout</code>	<code>active</code>	Show active layout per monitor
<code>get-hotkeys</code>	<code>hk</code>	List layout hotkeys
<code>get-layouts</code>	<code>ls</code>	List all layouts with visual previews
<code>get-monitors</code>	<code>m</code>	Show monitor info with active layouts
<code>open-editor</code>	<code>e</code>	Open FancyZones Editor GUI
<code>open-settings</code>	<code>settings</code>	Open FancyZones settings in PowerToys
<code>remove-hotkey</code>	<code>rhk</code>	Remove a layout hotkey
<code>set-hotkey</code>	<code>shk</code>	Assign a hotkey (0-9) to a layout
<code>set-layout</code>	<code>--</code>	Apply a layout to a monitor

## 4.11 File Explorer Add-ons

**Activation:** Automatic (integrates with File Explorer preview pane)

Extends File Explorer's preview pane and thumbnail generation to support additional file types that Windows does not handle natively.

**Supported preview formats:**

Format	Type
<code>.svg</code>	Scalable Vector Graphics
<code>.md</code>	Markdown (rendered with formatting)
<code>.pdf</code>	PDF documents
<code>.gcode</code>	G-code (3D printer instructions)
<code>.qoi</code>	QOI images
<code>.stl</code>	3D model files
Developer files	<code>.json</code> , <code>.xml</code> , <code>.yaml</code> , source code with syntax highlighting

### How to enable previews:

1. Open File Explorer
2. Click **View** in the toolbar
3. Select **Preview pane** (or press `Alt+P`)
4. Click any supported file to see a preview in the right panel

Thumbnail generation also works in icon views for supported formats -- you will see actual file content in the thumbnail instead of a generic icon.

## 4.12 File Locksmith

**Activation:** Right-click a file or folder in File Explorer > **What's using this file?**

Identifies which processes have a file or folder locked, so you can close them and proceed with your operation (move, rename, delete).

### How to use:

1. Right-click a file or folder that appears locked
2. Select **What's using this file?** from the context menu
3. File Locksmith displays a list of processes holding locks, including the process name, PID, and user
4. Click **End task** next to a process to release the lock, or close the process manually

#### Tip

On Windows 11, this option may be under **Show more options** in the right-click context menu. You can also run PowerToys as administrator to see locks from elevated processes.

## 4.13 Hosts File Editor

**Activation:** Open from PowerToys Settings or search via PowerToys Run

A GUI editor for the Windows hosts file (`C:\Windows\System32\drivers\etc\hosts`), replacing manual text editing.

### Features:

- Add, edit, and delete host entries with a clean interface
- Toggle individual entries on/off without deleting them (commented/uncommented)
- Duplicate entry detection
- Filter and search entries
- Automatic backup before editing

#### Caution

The hosts file affects DNS resolution for your entire system. Incorrect entries can prevent access to websites and services. The Hosts File Editor requires administrator elevation to save changes.

## 4.14 Image Resizer

**Activation:** Right-click image files in File Explorer > **Resize pictures**

Batch resize images directly from File Explorer. Supports JPEG, PNG, BMP, GIF, TIFF, and other common formats.

**How to use:**

1. Select one or more image files in File Explorer
2. Right-click and select **Resize pictures**
3. Choose a preset size or enter custom dimensions
4. Click **Resize**

**Built-in presets:**

Preset	Dimensions
Small	854 x 480
Medium	1366 x 768
Large	1920 x 1080
Phone	320 x 568
Custom	User-defined

**Settings:**

- Fit modes: Fill, Fit, Stretch
- Shrink only (never enlarge small images)
- Replace original or create a new file
- Fallback encoder selection
- Custom filename template with tokens: %1 (original name), %2 (size name), %3 (width), %4 (height), %5 (original extension), %6 (date)

**Image Resizer CLI**

Image Resizer also works from the command line ([PowerToys.ImageResizer.exe](#)) for scripting and batch processing:

```
powershell
```

```
# Resize a single image to 800x600
PowerToys.ImageResizer.exe "C:\Photos\vacation.jpg" --width 800 --height 600
```

```
powershell
```

```
# Batch resize all PNGs in a folder, fit mode, output to specific directory
PowerToys.ImageResizer.exe C:\Screenshots\*.png --width 1920 --height 1080 --fit fit --destination C:\Resize
```

```
powershell
```

```
# Resize using a preset (index 0 = Small, 1 = Medium, 2 = Large, 3 = Phone)
PowerToys.ImageResizer.exe photo.jpg --size 2
```

```
powershell
```

```
# Replace originals, strip metadata, preserve date
```

```
PowerToys.ImageResizer.exe *.jpg --width 1200 --height 800 --replace --remove-metadata --keep-date-modified
powershell
# Set JPEG quality to 80%
PowerToys.ImageResizer.exe photo.jpg --width 1920 --height 1080 --quality 80
powershell
# Show current configuration and exit
PowerToys.ImageResizer.exe --show-config
```

### CLI Options Reference:

Flag	Description
<code>--width</code>	Target width in pixels
<code>--height</code>	Target height in pixels
<code>--unit</code>	Unit of measurement (pixel, percent, etc.)
<code>--fit</code>	Resize mode: fill, fit, stretch
<code>--size</code>	Use a preset by index (0-3)
<code>--shrink-only</code>	Only resize if image is larger than target
<code>--replace</code>	Overwrite original files
<code>--ignore-orientation</code>	Ignore EXIF orientation data
<code>--remove-metadata</code>	Strip EXIF/metadata from output
<code>--quality</code>	JPEG quality (0-100)
<code>--keep-date-modified</code>	Preserve original file timestamps
<code>--filename</code>	Custom output filename template
<code>--destination</code>	Output directory path
<code>--progress-lines</code>	Print progress to stdout (for scripting integration)
<code>--show-config</code>	Display current settings and exit

## 4.15 Keyboard Manager

**Activation:** Open from PowerToys Settings

Remap individual keys and keyboard shortcuts system-wide or per-application.

### Key remapping:

- Remap any key to any other key (e.g., Caps Lock to Escape)
- Remap shortcuts to other shortcuts (e.g., `Ctrl+Shift+P` to `Ctrl+P`)
- Per-application remapping -- different remaps for different programs
- Disable keys by remapping to "Disable"

### How to remap a key:

1. Open Settings > Keyboard Manager
2. Click **Remap a key**
3. Click the **+** button
4. Press the original key, then press the desired replacement key
5. Click **OK**

### How to remap a shortcut:

1. Open Settings > Keyboard Manager
2. Click **Remap a shortcut**
3. Click the **+** button
4. Record the original shortcut and the target shortcut
5. Optionally select a target application (by executable name, e.g., `code.exe`)
6. Click **OK**

### Common remappings:

Original	Remapped To	Use Case
Caps Lock	Escape	Vim users
Caps Lock	Ctrl	Ergonomic modifier key
Insert	Disable	Prevent accidental overwrite mode
<code>Alt+Tab</code>	<code>Ctrl+Tab</code>	Browser-style window switching
Print Screen	<code>Win+Shift+S</code>	Snipping Tool instead of legacy screenshot

## 4.16 Light Switch

**Activation:** Configurable shortcut in Settings

Toggles Windows between light and dark appearance modes with a single keyboard shortcut. Useful for quickly switching when ambient lighting changes or when moving between a bright office and a dim room.

## 4.17 Mouse Utilities

Mouse Utilities is a collection of sub-tools that enhance mouse functionality. Each sub-tool has its own shortcut and settings.

### Find My Mouse

**Shortcut:** Double-press `Ctrl` (left or right)

Dims the entire screen except for a spotlight around your mouse cursor. Invaluable on multi-monitor setups or high-resolution displays where the cursor gets lost.

**Settings:**

- Activation method: double-press Ctrl, or shake mouse
- Spotlight radius and opacity
- Animation speed
- Excluded apps

## Mouse Highlighter

**Shortcut:** `Win+Shift+H` (toggle)

Highlights mouse clicks with colored circles -- yellow for left-click, blue for right-click by default. Perfect for presentations and screen recordings where viewers need to follow cursor actions.

## Mouse Jump

**Shortcut:** `Win+Shift+D`

Displays a thumbnail view of all monitors, allowing you to click to instantly teleport your mouse cursor to that position. Essential for ultra-wide and multi-monitor setups where moving the mouse across the full span is slow.

## Mouse Pointer Crosshairs

**Shortcut:** `Win+Alt+P`

Draws crosshair lines centered on your mouse cursor that span the entire screen, making it easy to align elements or find the cursor. The crosshairs follow cursor movement in real-time.

## 4.18 Mouse Without Borders

**Activation:** Configure from PowerToys Settings

Control up to four computers with one mouse and keyboard over your local network. Share clipboard content and drag files between machines seamlessly.

### Setup:

1. Install PowerToys on all machines
2. On the first machine: open Settings > Mouse Without Borders and note the **Security key**
3. On additional machines: open the same settings page and enter the security key from the first machine
4. Machines discover each other on the local network automatically

### Features:

- Seamless cursor movement across machines (move the cursor off one screen edge and it appears on the next machine)
- Shared clipboard (text and files)
- File drag-and-drop between computers
- Customizable screen layout (arrange machines to match physical desk layout)
- Keyboard shortcut to lock all machines simultaneously
- Encryption for clipboard and file transfers

## 4.19 New+

**Activation:** Right-click in File Explorer > **New** > select a template

Create new files from customizable templates directly in File Explorer's "New" context menu.

**How to use:**

1. Create template files in the New+ templates folder (configurable in Settings > New+)
2. Right-click in any File Explorer folder
3. Select **New** from the context menu
4. Your custom templates appear alongside default options
5. Select a template to create a new file pre-populated with the template content

**Example templates you might create:**

- A `.py` Python script with standard boilerplate (`#!/usr/bin/env python3`, imports, main guard)
- An `.html` page with doctype and basic structure
- A `.md` Markdown file with front matter
- A `.gitignore` with common patterns
- A `Dockerfile` with a base image and common directives

## 4.20 Peek

**Shortcut:** `Ctrl+Space`

Quick-preview files without opening their associated application. Similar to macOS Quick Look.

**Supported file types:**

- Images (PNG, JPG, GIF, BMP, SVG, WebP, ICO)
- Documents (PDF, Markdown)
- Web files (HTML)
- Audio/Video (via system codecs)
- Source code (with syntax highlighting)
- Microsoft Office documents (when preview handlers are installed)

**How to use:**

1. Select a file in File Explorer
2. Press `Ctrl+Space`
3. A preview window appears instantly
4. Press `Ctrl+Space` again or `Escape` to close
5. Use arrow keys to navigate between files while preview is open

## 4.21 Power Display

**Activation:** Configurable in Settings

Provides monitor information and display profile management. View details about connected displays (resolution, refresh rate, color profile) and switch between saved display configurations.

## 4.22 Power Rename

**Activation:** Right-click files/folders in File Explorer > **Power Rename**

Bulk rename files and folders using search-and-replace with regular expression support.

### Features:

- Regex search and replace
- Case-sensitive matching
- Preview changes before applying (see old name and new name side by side)
- Include/exclude files, folders, or subfolders
- Enumeration support for sequential numbering
- Apply to filename only, extension only, or both

### Example -- add a date prefix to all JPEG files:

1. Select files in File Explorer
2. Right-click > **Power Rename**
3. Search for: `(.*)` (regex)
4. Replace with: `2026-05-07_`\$1
5. Enable **Use Regular Expressions**
6. Review the preview column, then click **Apply**

### Example -- replace spaces with underscores:

1. Select files
2. Right-click > **Power Rename**
3. Search for: (space)
4. Replace with: `_`
5. Click **Apply**

### Example -- sequential numbering:

1. Select files
2. Search for: `.*` (regex)
3. Replace with: `Photo_`\${}
4. The `${}` token inserts a sequential number

## 4.23 PowerToys Run

**Shortcut:** `Alt+Space`

A quick launcher for applications, files, folders, and system actions. Type to search, press Enter to launch.

### Built-in plugins:

Prefix	Plugin	Function
(none)	Programs	Search installed applications

?	Web Search	Search the web via default browser
=	Calculator	Evaluate math expressions
>	Shell	Run shell commands
//	Bookmarks	Search browser bookmarks
<	Running Processes	Find and manage running processes
~~	Registry	Search Windows Registry keys
:	Time/Date	Show current time, date, time zones
{	VSCode Workspaces	Open recent VS Code workspaces
!	Windows Settings	Search Windows Settings pages
\$	History	Search recent PowerToys Run results
%%	Unit Converter	Convert between units

**Tips:**

- Results update as you type -- no need to press Enter before seeing results
- Use `Ctrl+Shift+Enter` to run a result as administrator
- Right-click a result for additional actions (open file location, run as admin, copy path)
- Configure individual plugins on/off in Settings > PowerToys Run > Plugins

**Calculator examples:**

```
= 15% of 230
= sqrt(144)
= 2^10
= sin(45)
```

## 4.24 Quick Accent

**Activation:** Hold a letter key, then press the activation key (Space or left/right arrow by default)

Quickly type accented and special characters without memorizing alt codes or switching keyboard layouts.

**How to use:**

1. Hold down a letter key (e.g., `a`)
2. Press the activation key (default: Space bar)
3. An overlay appears with accent options: `à á â ã ä å æ`
4. Press Space repeatedly to cycle through options, or use arrow keys to select
5. Release all keys to insert the selected character

**Settings:**

- Activation key: Space, Left/Right arrow keys
- Character sets: selectable per language (French, Spanish, German, Portuguese, Polish, Czech, and more)
- Toolbar position on screen

## 4.25 Registry Preview

**Activation:** Open from PowerToys Settings, or double-click a `.reg` file with Registry Preview as the handler. Visualize and inspect Windows Registry `.reg` files before importing them.

### Features:

- Tree view of registry keys and values -- see the hierarchy at a glance
- Syntax highlighting for `.reg` file content
- Edit values before applying
- Side-by-side view: raw file content vs. parsed tree
- Write modified content back to the file

## 4.26 Screen Ruler

**Shortcut:** `Win+Shift+M`

Measure pixel distances and dimensions on screen. Essential for designers and developers working on UI layouts.

### Measurement modes:

Mode	Function
Bounds	Measure width and height of a rectangular region
Spacing	Measure distance between edges of windows/elements
Horizontal spacing	Measure horizontal distance only
Vertical spacing	Measure vertical distance only

### How to use:

1. Press `Win+Shift+M` to activate
2. Select a measurement mode from the toolbar at the top of the screen
3. Move your cursor -- measurements update in real-time, showing pixel distances
4. Press `Win+Shift+M` again or `Escape` to deactivate

## 4.27 Shortcut Guide

**Shortcut:** Hold `Win` key for approximately 1 second

Displays a full-screen overlay showing available Windows keyboard shortcuts for the current state of the desktop.

### What it shows:

- Windows key shortcuts (`Win+E` for Explorer, `Win+D` for desktop, `Win+L` for lock, etc.)
- Taskbar shortcuts (`Win+1` through `Win+9` for pinned taskbar apps)
- Snap layouts (`Win+Arrow keys`)

- Virtual desktop shortcuts ([Win+Ctrl+D](#) for new desktop, [Win+Ctrl+Left/Right](#) to switch)

The overlay disappears when you release the Win key or press Escape.

## 4.28 Text Extractor (OCR)

**Shortcut:** [Win+Shift+T](#)

Extract text from any region of the screen using Optical Character Recognition. Works with images, videos, non-selectable text in applications, screenshots -- anything visible on screen.

### How to use:

1. Press [Win+Shift+T](#)
2. Your screen dims and a crosshair cursor appears
3. Click and drag to draw a rectangle around the text you want to extract
4. The extracted text is copied to your clipboard instantly
5. Paste it anywhere with [Ctrl+V](#)

### Settings:

- Preferred OCR language (must have the Windows language pack installed)
- Table mode for structured text extraction (preserves column/row alignment)

## 4.29 Workspaces

**Shortcut:** [Win+Ctrl+W](#)

Save and restore complete window layouts -- application positions, sizes, and monitor placement -- for different projects or workflows.

### How to use:

1. Arrange your windows for a specific workflow
2. Press [Win+Ctrl+W](#) to open the Workspaces editor
3. Click **Create Workspace** to capture the current layout
4. Name the workspace (e.g., "Development", "Writing", "Meetings")
5. To restore: press [Win+Ctrl+W](#) and select a saved workspace -- all applications launch and position themselves automatically

### Features:

- Launch applications and position windows automatically
- Per-monitor workspace definitions
- Desktop shortcut creation for one-click workspace launch
- Edit workspaces to add/remove applications after creation
- Works with already-running applications (moves them to saved positions)

## 4.30 ZoomIt

**Activation:** Configurable shortcuts in Settings

A screen zoom, annotation, and break timer tool designed for presentations, demos, and screen recording.

### Modes:

Mode	Default Shortcut	Function
Zoom	<code>Ctrl+1</code>	Magnify screen area; scroll to adjust zoom level
Draw	<code>Ctrl+2</code>	Draw freehand annotations on screen
Type	<code>Ctrl+3</code>	Type text on screen at the cursor position
Break Timer	<code>Ctrl+4</code>	Full-screen countdown timer
Record	<code>Ctrl+5</code>	Screen recording
Snip	<code>Ctrl+6</code>	Screen capture to clipboard

### Drawing controls while in Draw or Zoom mode:

Control	Action
Left-click + drag	Freehand draw
<code>Shift</code> + drag	Straight line
<code>Ctrl</code> + drag	Rectangle
<code>Tab</code> + drag	Ellipse
<code>Shift+Ctrl</code> + drag	Arrow
<code>E</code>	Eraser tool
<code>K</code>	Clear screen (blackboard)
<code>W</code>	Clear screen (whiteboard)
<code>R, G, B, O, Y, P</code>	Change pen color (Red, Green, Blue, Orange, Yellow, Pink)

## 5. Common Tasks

### 5.1 Setting Up a Multi-Monitor Workspace

Combine several PowerToys modules for an optimal multi-monitor setup:

1. **FancyZones** -- Define custom layouts per monitor. Use a wide development layout on your primary display and a stacked communications layout on the secondary.

powershell

```
# Check your monitor setup via CLI
PowerToys.FancyZonesCLI.exe get-monitors
```

Expected output:

```
Monitor #1: 3840x2160 @ (0,0) [Primary]
Active layout: (none)
```

```
Monitor #2: 2560x1440 @ (3840,0)
Active layout: (none)
```

2. **Cursor Wrap** -- Enable cursor wrapping so the mouse seamlessly moves between monitors at screen edges.
3. **Mouse Jump** -- Press [Win+Shift+D](#) to teleport the cursor between distant monitors instantly when traversal is too slow.
4. **Workspaces** -- Save the arrangement as a Workspace ([Win+Ctrl+W](#)) so you can restore it after a reboot, docking change, or when switching between projects.

## 5.2 Batch Processing Files

### Rename hundreds of files with a pattern:

1. Select all target files in File Explorer
2. Right-click > **Power Rename**
3. Search: `IMG_(\d{4})\.jpg` (regex enabled)
4. Replace: `Vacation_2026_$.jpg`
5. Preview and apply

### Resize a folder of images for web:

```
powershell
```

```
# Resize all JPEGs to max 1200px wide, 80% quality, strip metadata
PowerToys.ImageResizer.exe C:\Photos\*.jpg --width 1200 --height 900 --fit fit --quality 80 --remove-metadata
```

### Rename + Resize workflow:

1. Use Power Rename to standardize filenames first
2. Then use Image Resizer to batch-resize the renamed files
3. Output goes to a separate directory so originals are preserved

## 5.3 Presentation Mode

Prepare for a presentation or demo using these modules:

1. **Awake** -- Right-click the tray icon > keep awake indefinitely (prevents embarrassing sleep mid-presentation)
2. **ZoomIt** -- Use [Ctrl+1](#) to zoom into code or UI details, [Ctrl+2](#) to draw annotations, [Ctrl+4](#) for break countdown timers
3. **Mouse Highlighter** -- Press [Win+Shift+H](#) to make every click visible to the audience
4. **Always on Top** -- Pin your demo window with [Win+Ctrl+T](#) so it stays visible when switching to notes or other apps

### ZoomIt drawing during a zoomed-in presentation:

1. Press [Ctrl+1](#) to zoom in
2. Scroll wheel to adjust zoom level

3. Press `Ctrl+2` to start drawing (or just start drawing while zoomed)
4. Use `Shift+drag` for straight lines, `Ctrl+drag` for rectangles
5. Press `Escape` to exit drawing, `Escape` again to exit zoom

## 5.4 Developer Productivity

### Quick file inspection without leaving Explorer:

- Press `Ctrl+Space` (Peek) on any source file for syntax-highlighted preview
- Enable File Explorer Add-ons for Markdown, JSON, SVG preview in the preview pane
- Use Text Extractor (`Win+Shift+T`) to grab text from error dialogs or console screenshots

### Environment variable management:

- Open Environment Variables from PowerToys Settings
- Create profiles: "Python Dev", "Node Dev", "Go Dev" with different PATH entries and SDK variables
- Switch profiles without rebooting -- changes take effect for new processes immediately

### Registry file review:

- Before importing `.reg` files, open them with Registry Preview to inspect the tree structure and values
- Edit values directly in the preview before applying to your registry

### Window management for coding:

- Set up a FancyZones layout: editor (60% left), terminal (40% right), browser on monitor 2
- Save as a Workspace called "Development"
- Press `Win+Ctrl+W` at the start of each day to launch your entire dev environment in one click

## 5.5 Extracting Text from Images and Screenshots

1. Take a screenshot or have any image displayed on screen
2. Press `Win+Shift+T` (Text Extractor)
3. Draw a rectangle around the text
4. Paste the extracted text -- it is now on your clipboard

This works with:

- Error dialog boxes with non-selectable text
- Text in images, memes, or infographics
- Text in video frames (pause the video first for best accuracy)
- Scanned documents displayed on screen
- Legacy applications that do not support text selection
- CAPTCHAs and watermarked text (accuracy varies)

For best OCR accuracy, ensure the text is clearly visible and not too small. Use ZoomIt (`Ctrl+1`) to magnify small text before extracting.

## 5.6 Taming Keyboard Shortcuts

## Remap uncomfortable shortcuts:

Open Keyboard Manager and create per-app remappings:

Application	Original	Remapped	Purpose
All apps	Caps Lock	Escape	Vim-friendly Caps Lock
<code>code.exe</code>	<code>Ctrl+Shift+P</code>	<code>Ctrl+P</code>	Swap palette shortcuts in VS Code
<code>terminal.exe</code>	<code>Ctrl+W</code>	<code>Ctrl+Shift+W</code>	Prevent accidental tab close in Terminal
All apps	Insert	(Disabled)	Prevent accidental overwrite mode

## 5.7 Working with Colors (Design Workflows)

1. Press `Ctrl+Break` to activate Color Picker
2. Hover over any pixel on any monitor -- the magnified preview shows the exact color
3. Click to capture the color
4. The editor shows the color in HEX, RGB, HSL, and all other configured formats
5. Click any format to copy it to clipboard

**Color history** is preserved across sessions, so you can refer back to previously picked colors.

## 5.8 Creating a Custom File Template Library

Set up New+ to create project files instantly:

1. Open Settings > New+
2. Note the templates directory path
3. Create template files in that directory:

```
powershell

# Navigate to the templates directory
cd "$env:LOCALAPPDATA\Microsoft\PowerToys\NewPlus\Templates"

# Create a Python template
Set-Content "script.py" @"
#!/usr/bin/env python3
"""Module docstring."""

def main():
    pass

if __name__ == "__main__":
    main()
"@
```

4. Right-click in any folder in File Explorer > **New** > your templates appear ready to use

## 5.9 Investigating Locked Files

When you encounter "This file is in use" errors:

1. Right-click the locked file > **What's using this file?** (File Locksmith)
2. Review the list of processes
3. Decide whether to:
  - **End task** from File Locksmith directly
  - Save work in the locking application and close it manually
  - Copy the file to a new location instead of moving/renaming

## 5.10 Keeping a Script Running Without Sleep

Use Awake CLI to prevent sleep only while a long-running script executes:

```
powershell

# Start Awake bound to the current PowerShell process
Start-Process PowerToys.Awake.exe -ArgumentList "--display-on --use-parent-pid" -NoNewWindow

# Run your long process
python train_model.py

# When the script finishes and PowerShell closes, Awake automatically deactivates
```

Alternatively, bind to a specific process:

```
powershell

$proc = Start-Process python -ArgumentList "train_model.py" -PassThru
PowerToys.Awake.exe --pid $proc.Id --display-on
```

# 6. Troubleshooting

## 6.1 PowerToys Won't Start

**Symptoms:** PowerToys icon does not appear in system tray after login.

**Solutions:**

1. Check the Task Scheduler for the PowerToys startup task:

```
powershell

Get-ScheduledTask -TaskPath "\PowerToys\" | Format-Table TaskName, State
```

Expected output:

TaskName	State
-----	-----
Autorun	Ready

2. If the task is missing or disabled:

```
powershell

Enable-ScheduledTask -TaskPath "\PowerToys\" -TaskName "Autorun"
```

3. Launch manually from the install directory:

```
powershell
```

```
&amp; "$env:LOCALAPPDATA\PowerToys\PowerToys.exe"
```

4. If PowerToys crashes immediately, check Event Viewer (Application log) for .NET runtime errors. Reinstalling typically resolves corrupted installations.

## 6.2 Module Not Working After Update

**Symptoms:** A module stops responding or behaves differently after a PowerToys update.

**Solutions:**

1. Reset the specific module's settings:

```
powershell
```

```
$module = "FancyZones" # Replace with the affected module name
$path = "$env:LOCALAPPDATA\Microsoft\PowerToys\$module\settings.json"
Rename-Item $path "$path.bak"
```

2. Restart PowerToys from the system tray (right-click > **Restart PowerToys**)
3. If the issue persists, perform a clean reinstall:

```
powershell
```

```
winget uninstall Microsoft.PowerToys
Remove-Item "$env:LOCALAPPDATA\Microsoft\PowerToys" -Recurse -Force
winget install Microsoft.PowerToys --source winget
```

## 6.3 Keyboard Manager Remapping Errors

**Related issue:** [#34798](#)

**Symptoms:** Keyboard Manager shows errors, remappings do not take effect, or keys become stuck after remapping.

**Solutions:**

1. Open Settings > Keyboard Manager and remove all remappings
2. Restart PowerToys
3. Re-add remappings one at a time to identify the conflicting remap
4. Avoid remapping keys that are part of other PowerToys activation shortcuts (circular dependency)
5. If keys appear stuck, press and release each modifier key individually: [Ctrl](#), [Alt](#), [Shift](#), [Win](#)

## 6.4 Command Not Found Not Working

**Related issues:** [#31350](#), [#33304](#), [#33669](#)

**Symptoms:** The module does not suggest packages, or PowerShell throws exceptions on startup.

**Solutions:**

1. Verify you are using PowerShell 7+ (not Windows PowerShell 5.1):

```
powershell
```

```
$PSVersionTable.PSVersion
```

Expected output (major version must be 7 or higher):

Major	Minor	Patch
7	5	0

## 2. Ensure WinGet is installed and functional:

```
powershell
```

```
winget --version
```

## 3. If you see an "untrusted repository" warning ([#31378](#)):

```
powershell
```

```
Set-PSRepository -Name PSGallery -InstallationPolicy Trusted  
Install-Module -Name Microsoft.WinGet.CommandNotFound -Force
```

## 4. If PowerShell throws exceptions on startup ([#33304](#)), update to the latest PowerShell 7.x:

```
powershell
```

```
winget upgrade Microsoft.PowerShell
```

## 5. The module requires internet access to query WinGet package databases. Without connectivity, it silently does nothing ([#33669](#)).

## 6.5 FancyZones Windows Not Snapping

**Related issue:** [#38084](#)

**Symptoms:** Windows do not snap to zones when dragging, or child windows snap over their parent windows.

**Solutions:**

1. Verify FancyZones is enabled and the zone overlay appears when you hold Shift while dragging
2. Check that the application is not in the excluded apps list (Settings > FancyZones > Excluded apps)
3. If the target application is running as administrator, PowerToys must also run as administrator to manage its windows
4. For child window issues ([#38084](#)), disable "Allow child windows to snap" or add the parent application to the excluded list
5. Reset FancyZones zone data:

```
powershell
```

```
Remove-Item "$env:LOCALAPPDATA\Microsoft\PowerToys\FancyZones\zones-settings.json"  
# Restart PowerToys afterward
```

## 6.6 Mouse Pointer Crosshairs Not Tracking

**Related issue:** [#24096](#)

**Symptoms:** Crosshairs do not follow cursor movement when using a trackpad.

**Solutions:**

1. This is a known issue with certain trackpad drivers that do not report continuous position updates
2. Try updating your trackpad/touchpad drivers from the manufacturer's website
3. As a workaround, use Find My Mouse (double-Ctrl) instead for cursor location

## 6.7 Find My Mouse Activating Unintentionally

**Related issue:** [#18529](#)

**Symptoms:** Find My Mouse triggers while clicking or during gaming.

**Solutions:**

1. Change the activation method from "Double-press Ctrl" to "Shake mouse" in Settings > Mouse Utilities > Find My Mouse
2. Add game executables to the excluded applications list
3. Increase the double-press speed threshold to require faster double-presses

## 6.8 Mouse Without Borders Connection Issues

**Related issue:** [#35341](#)

**Symptoms:** Machines do not discover each other, clipboard/file sharing fails.

**Solutions:**

1. Ensure all machines are on the same local network subnet
2. Check Windows Firewall -- allow PowerToys through the firewall on all machines:

powershell

```
New-NetFirewallRule -DisplayName "PowerToys Mouse Without Borders" `
  -Direction Inbound `
  -Program "$env:LOCALAPPDATA\PowerToys\PowerToys.MouseWithoutBorders.exe" `
  -Action Allow
```

3. Verify the security key matches exactly on all machines
4. File drag and clipboard sharing may fail when running as a service ([#35341](#)) -- run in user mode instead

## 6.9 Awake Not Keeping System Awake

**Related issue:** [#35572](#)

**Symptoms:** System sleeps despite Awake being enabled with "Keep awake indefinitely."

**Solutions:**

1. Verify the Awake tray icon shows the active state (filled coffee cup, not empty outline)
2. Check that "Keep display on" is enabled if your monitor is turning off
3. Group Policy or enterprise power management can override Awake -- check with your IT administrator
4. Test with the CLI to rule out settings issues:

powershell

```
PowerToys.Awake.exe --display-on --time-limit 3600
```

5. Some Windows updates can reset power management APIs -- update PowerToys to the latest version

## 6.10 Image Resizer DLL Load Failure

**Related issue:** [#34745](#)

**Symptoms:** Error "Failed to load PowerToys.ImageResizerExt.dll" in File Explorer context menu.

**Solutions:**

1. Restart File Explorer:

```
powershell
```

```
Stop-Process -Name explorer -Force  
Start-Process explorer
```

2. Re-register the shell extension:

```
powershell
```

```
regsvr32 "$env:LOCALAPPDATA\PowerToys\PowerToys.ImageResizerExt.dll"
```

3. If the DLL is missing, repair the installation:

```
powershell
```

```
winget install Microsoft.PowerToys --source winget --force
```

## 6.11 Cannot Disable Mouse Utilities

**Related issue:** [#34524](#)

**Symptoms:** Toggling Mouse Utilities off in settings does not stop the module from running.

**Solutions:**

1. Disable each sub-tool individually (Find My Mouse, Highlighter, Jump, Crosshairs) rather than the parent toggle
2. Restart PowerToys after disabling
3. If the module persists, update PowerToys to the latest version -- this was addressed in a subsequent release

## 6.12 PowerToys Run Not Finding Applications

**Symptoms:** Installed applications do not appear in PowerToys Run results.

**Solutions:**

1. Open Settings > PowerToys Run > Plugins > Programs and verify the plugin is enabled
2. Verify the program sources include Start menu, Desktop, PATH, and Registry
3. Clear the PowerToys Run cache:

```
powershell
```

```
Remove-Item "$env:LOCALAPPDATA\Microsoft\PowerToys\PowerToys Run\Cache\*" -Force
```

4. Restart PowerToys and wait a moment for the index to rebuild

## 6.13 Color Picker Not Activating

**Symptoms:** Pressing `Ctrl+Break` does nothing.

**Solutions:**

1. Some keyboards do not have a [Break](#) key (especially compact layouts). Remap the activation shortcut in Settings > Color Picker
2. Common alternatives: [Win+Shift+C](#) or [Win+C](#)
3. Check that no other application has claimed the same shortcut
4. Verify Color Picker is enabled in Settings

## 6.14 Advanced Paste AI Features Not Working

**Symptoms:** Custom AI actions return errors or produce no output.

**Solutions:**

1. Verify your OpenAI API key is correctly entered in Settings > Advanced Paste
2. Check your API key has available credits/quota at [platform.openai.com](https://platform.openai.com)
3. Non-AI features (paste as plain text, Markdown, JSON) work without an API key -- test these first to isolate the issue
4. Ensure your network allows HTTPS connections to [api.openai.com](https://api.openai.com)

## 6.15 Settings File Corruption

**Symptoms:** PowerToys behaves unexpectedly, modules have wrong settings, or the Settings app crashes on launch.

**Solutions:**

powershell

```
# Back up current settings before resetting
Copy-Item "$env:LOCALAPPDATA\Microsoft\PowerToys" `
  "$env:USERPROFILE\Desktop\PowerToys-Settings-Backup" -Recurse

# Reset all settings to defaults
Remove-Item "$env:LOCALAPPDATA\Microsoft\PowerToys\*" -Recurse -Force

# Restart PowerToys
Stop-Process -Name PowerToys -Force -ErrorAction SilentlyContinue
&& "$env:LOCALAPPDATA\PowerToys\PowerToys.exe"
```

## 6.16 Shell Extensions Not Appearing in Context Menu

**Symptoms:** Right-click context menu items (Image Resizer, Power Rename, File Locksmith) are missing.

**Solutions:**

1. On Windows 11, PowerToys entries appear under **Show more options** in the context menu
2. Restart File Explorer:

powershell

```
Stop-Process -Name explorer -Force
Start-Process explorer
```

3. Verify the modules are enabled in PowerToys Settings
4. If using Windows 11, check each module's Settings for "Show in extended context menu" toggle

## 6.17 Text Extractor Returns Incorrect Text

**Symptoms:** OCR results are garbled, wrong language, or missing characters.

### Solutions:

1. Install the correct language pack for the text you are extracting:

powershell

```
# List installed OCR language packs
Get-WindowsCapability -Online | Where-Object {
    $_.Name -like "Language.OCR*" -and $_.State -eq "Installed"
}
```

2. Install additional language support:

powershell

```
# Example: install Japanese OCR support
Add-WindowsCapability -Online -Name "Language.OCR~~ja-JP~0.0.1.0"
```

3. Set the preferred language in Settings > Text Extractor
4. For better accuracy, zoom in on the text before extracting (use ZoomIt [Ctrl+1](#))
5. Ensure the text is clearly visible -- low contrast or small font sizes reduce accuracy

## 6.18 Elevated Application Compatibility

**Symptoms:** PowerToys cannot interact with applications running as administrator (FancyZones won't snap, Always on Top won't pin, Keyboard Manager remaps do not work in elevated apps).

### Solutions:

1. Run PowerToys as administrator: Settings > General > **Always run as administrator**
2. Restart PowerToys -- it will request UAC elevation
3. Note: running as administrator means the startup task is updated automatically to launch elevated

## 6.19 High Memory or CPU Usage

**Symptoms:** PowerToys processes consume excessive system resources.

### Solutions:

1. Disable modules you do not use -- each enabled module runs its own process and consumes resources
2. Check which module is the culprit:

powershell

```
Get-Process *PowerToys* | Sort-Object WorkingSet64 -Descending |
    Format-Table Name, @{N='Memory(MB)';E={ [math]::Round($_.WorkingSet64/1MB,1) }}, CPU
```

3. FancyZones and Mouse Without Borders tend to be the heaviest -- disable if not needed
4. Restart PowerToys to clear any accumulated state

## 6.20 Workspaces Not Restoring Correctly

**Symptoms:** Applications launch but windows are not positioned in the saved locations.

### Solutions:

1. Ensure the monitor configuration matches what was saved (same number of monitors, same resolution, same arrangement)
2. Some UWP/Store applications do not support programmatic window positioning
3. Edit the workspace to remove applications that cannot be positioned reliably
4. Give applications time to launch -- some apps need a few seconds to fully initialize before they accept window position commands

## 6.21 Quick Accent Not Showing Characters

**Symptoms:** Holding a letter key and pressing Space does not show accent options.

**Solutions:**

1. Verify Quick Accent is enabled in Settings
2. Check the activation key setting -- it must match your chosen key (Space, Left arrow, or Right arrow)
3. Some keyboard layouts or input methods interfere -- test with a standard US English layout
4. Ensure the character set for your desired language is selected in Settings > Quick Accent

## 6.22 FancyZones Layout Editor Won't Open

**Symptoms:** Pressing `Win+Shift+`` does nothing, or the editor window opens off-screen.

**Solutions:**

1. Check for conflicting shortcuts from other applications
2. Try opening via CLI:

```
powershell
```

```
PowerToys.FancyZonesCLI.exe open-editor
```

3. Reset FancyZones editor position if it is stuck off-screen:

```
powershell
```

```
Remove-Item "$env:LOCALAPPDATA\Microsoft\PowerToys\FancyZones\editor-parameters.json" -Force
```

## 6.23 Peek Preview Not Rendering

**Symptoms:** Peek opens but shows a blank window or an error message.

**Solutions:**

1. Ensure WebView2 Runtime is installed (required for Peek to render previews)
2. Check that the file type is supported by Peek
3. For large files, Peek may take a moment to render -- wait a few seconds
4. Update PowerToys to the latest version for expanded file type support

## 6.24 New+ Templates Not Appearing

**Symptoms:** Custom templates do not show in the right-click > New menu.

**Solutions:**

1. Verify template files are in the correct templates directory (check Settings > New+)
2. Template files must have the correct file extension for the type you want
3. Restart File Explorer after adding new templates:

```
powershell
```

```
Stop-Process -Name explorer -Force  
Start-Process explorer
```

4. File Explorer caches the New menu -- a sign-out/sign-in may be needed for new templates

## 6.25 Registry Preview Not Opening .reg Files

**Symptoms:** Double-clicking `.reg` files opens the built-in Registry Editor instead of Registry Preview.

**Solutions:**

1. Set Registry Preview as the default handler:
  - Right-click a `.reg` file > **Open with** > **Choose another app**
  - Select Registry Preview and check **Always use this app**
2. Alternatively, open Registry Preview from PowerToys Settings and drag/drop `.reg` files into it

## 7. FAQ

### General

**Q: Is PowerToys free?**

A: Yes. PowerToys is completely free and open-source under the MIT license. There are no paid tiers, subscriptions, or premium features.

**Q: Does PowerToys work on Windows 10?**

A: Yes. PowerToys supports Windows 10 version 2004 (build 19041) and later, as well as all versions of Windows 11.

**Q: Does PowerToys run in the background?**

A: Yes. PowerToys Runner is a lightweight background process that loads enabled modules. It starts automatically at login via a Task Scheduler task. You can exit it from the system tray icon at any time.

**Q: Can I use only some modules?**

A: Absolutely. Each module can be independently enabled or disabled from Settings. Disabled modules are not loaded and consume no resources.

**Q: Does PowerToys work with ARM64 Windows devices?**

A: Yes. PowerToys provides native ARM64 builds.

**Q: How do I run PowerToys as administrator?**

A: Open Settings > General and enable **Always run as administrator**. This is necessary for PowerToys to interact with elevated (admin) applications.

**Q: Where are my settings stored?**

A: All settings are stored as JSON files in `%LOCALAPPDATA%\Microsoft\PowerToys\`. Each module has its own subfolder.

**Q: Can I sync settings across computers?**

A: There is no built-in settings sync. You can manually copy the `%LOCALAPPDATA%\Microsoft\PowerToys\` folder between machines, or use a cloud sync tool to keep them in sync.

**Q: Does PowerToys support Group Policy?**

A: Yes. Enterprise administrators can manage PowerToys modules and settings via Group Policy. See Admin Guide for full configuration reference.

**Q: How do I completely reset PowerToys to defaults?**

A: Delete the settings folder at `%LOCALAPPDATA%\Microsoft\PowerToys\` and restart PowerToys. All modules will revert to their default settings.

**Q: How often does PowerToys check for updates?**

A: Every 24 hours, via GitHub releases. It will not auto-download on metered connections unless you configure it to.

**Q: Does PowerToys conflict with other utility suites?**

A: PowerToys generally coexists well with other utilities. The most common conflict is when multiple tools register the same keyboard shortcuts. If you experience conflicts, change the PowerToys shortcut from its module settings page.

## Module-Specific

**Q: What is the difference between PowerToys Run and Command Palette?**

A: PowerToys Run ([Alt+Space](#)) is the established quick launcher. Command Palette ([Win+Alt+Space](#)) is its next-generation successor with improved extensibility and a richer plugin architecture. Both coexist -- you can use either or both simultaneously.

**Q: Can FancyZones replace Windows Snap?**

A: FancyZones works alongside Windows Snap. When you hold Shift while dragging, FancyZones zones appear instead of the default Snap layouts. You can also configure FancyZones to override Snap behavior entirely in its settings.

**Q: Does Advanced Paste require an internet connection?**

A: The standard paste formats (plain text, Markdown, JSON) work entirely offline. Only the AI-powered custom actions require an internet connection and a configured OpenAI API key.

**Q: Can Mouse Without Borders work across different networks?**

A: Mouse Without Borders is designed for local network use. All machines must be on the same network subnet. It does not support WAN/internet connections or VPN-bridged setups reliably.

**Q: What OCR languages does Text Extractor support?**

A: Text Extractor uses the Windows OCR engine and supports any language for which you have installed the Windows language pack. Common languages include English, Chinese (Simplified and Traditional), Japanese, Korean, French, German, Spanish, Portuguese, and many more.

**Q: Can Keyboard Manager remap mouse buttons?**

A: No. Keyboard Manager handles keyboard keys and shortcuts only. It cannot remap mouse buttons or scroll wheel actions.

**Q: Does Always on Top work with fullscreen games?**

A: Fullscreen exclusive applications bypass the Windows window manager, so Always on Top cannot pin them. Games running in borderless windowed mode work correctly with Always on Top.

**Q: Can I use FancyZones with virtual desktops?**

A: Yes. FancyZones layouts apply per-monitor and work across virtual desktops. Each virtual desktop can use the same zone layouts independently.

**Q: How does Workspaces handle apps that are already running?**

A: Workspaces will move already-running application windows to their saved positions. If an application is not running, Workspaces will launch it first and then position the window.

**Q: Can Image Resizer handle RAW photo formats?**

A: Image Resizer supports formats that .NET/WIC can decode. Standard RAW formats (CR2, NEF, ARW) may work if the appropriate Windows codec is installed via the Microsoft Store. JPEG, PNG, BMP, GIF, and TIFF work without additional codecs.

**Q: Does ZoomIt work in remote desktop sessions?**

A: Yes. ZoomIt works in RDP sessions, though some drawing features may have reduced performance depending on the connection quality.

**Q: Can I add my own file types to File Explorer Add-ons?**

A: The supported file types are built into the PowerToys codebase. Custom format support would require extending the source code. For general file preview, use Peek ([Ctrl+Space](#)), which supports a wider range of file types.

**Q: Does Color Picker work on multi-monitor setups?**

A: Yes. Color Picker works across all connected monitors. The magnifier follows your cursor to any screen.

**Q: Can Power Rename undo a batch rename?**

A: Power Rename does not have a built-in undo. Always preview your changes before clicking Apply. For safety, copy files to a backup folder before bulk renaming.

**Q: How do I disable the Shortcut Guide overlay?**

A: Go to Settings > Shortcut Guide and toggle the module off. You can also increase the press duration so it does not trigger accidentally when you press the Win key briefly.

**Q: Can Quick Accent work with non-Latin scripts?**

A: Quick Accent is designed for Latin-based scripts and their diacritical marks. It supports characters for major European languages. For CJK or other script input, use the Windows Input Method Editor (IME) instead.

**Q: Does Crop and Lock's reparent mode work with all applications?**

A: Reparent mode works with most standard Win32 applications. Applications that use custom rendering (DirectX, OpenGL) or non-standard window management may not work correctly. Use thumbnail mode as a fallback for those cases.

**Q: How many machines can Mouse Without Borders connect?**

A: Up to four machines simultaneously. All machines must run PowerToys with the same security key.

**Q: Can I trigger Workspaces from the command line?**

A: Workspaces can create desktop shortcuts for saved workspaces. You can invoke these `.lnk` shortcuts from scripts or the command line.

**Q: Is there a portable version of PowerToys?**

A: No. PowerToys requires installation because it registers shell extensions, system services, and scheduled tasks that cannot function from a portable deployment.

**Q: Can File Locksmith show network locks?**

A: File Locksmith shows processes on the local machine that hold file locks. It does not detect locks held by remote processes accessing shared files over the network.

**Q: Does Peek support Office documents (Word, Excel, PowerPoint)?**

A: Yes, when the Office desktop applications or the appropriate Office preview handlers are installed on your system.

**Q: Can I use the FancyZones CLI without the PowerToys GUI running?**

A: The CLI requires the FancyZones module to be running (as part of PowerToys) to interact with layouts and monitors. It is a control interface, not a standalone tool.

*DocCompiler.ai v3.1 | Upgraded source coverage | Generated 2026-05-07 | Source: [@microsoft/PowerToys](https://github.com/microsoft/PowerToys) @ [fde1599](https://github.com/fde1599) A Cyber Panda Solutions LLC product. This document is intentionally comprehensive -- built to be the complete, source-verified reference an AI or human needs without consulting other sources. **Do not print:** the always-current version lives at [doccompiler.ai/microsoft/PowerToys](https://doccompiler.ai/microsoft/PowerToys).*