Table of Contents

Hazel Overview 4
What's New in Hazel 6 6
Installation & Setup 9
Install Hazel 10
Understand Evaluation and Demo Modes 12
Purchase a Hazel License 13
Register Hazel 15
Update Hazel 16
Uninstall Hazel 18
Hazel Basics 19
The Main Hazel Window 20
About Folders & Rules 22
The Hazel Status Menu 26
Stopping & Restarting Hazel 27
Notifications 28
Hazel Settings 29
Manage Your Trash 33
Use Automatic Deletion 34
Use App Sweep 36
Work with Folders & Rules 38
Manage Folders 39
Create & Edit Rules 44
Understand the Logic of Rules 45
Create a Rule 48
Preview a Rule 51
Copy or Move a Rule 53
Edit or Remove a Rule 54
Use Sample Rules 55
Search Rules 56
Manage Rules 57
Enable, Disable, or Pause Rules 58
Run Rules Manually 59

Show Rule Status 60
Revert a File 61
Import Rules 62
Export Rules 63
Sync Rules 64
Troubleshoot Rules 66
Attributes & Actions 67
Attribute Reference 69
Using Match Patterns in Conditions 73
Using Custom Attributes 77
Custom Text Attributes 78
Custom Date Attributes 82
Custom List Attributes 85
Custom List Item Attributes 94
Custom Table Attributes 97
Reusing Custom Attributes in Condition Patterns 101
Action Reference 104
Copying Folder Structure 110
Specifying Upload Options 114
Replacing Folders 116
Using Patterns in Actions 117
Using the Counter Attribute 121
Editing Date Patterns 124
Using Custom Attributes in Actions 126
Using Shortcuts 135
Using AppleScript or JavaScript 137
Using Automator 146
Using Shell Scripts 148
Advanced Topics 151
Processing Subfolders 152
Using Nested Conditions 157
Syncing Folders 162
Viewing Logs 165
Learn More 166

Credits	167

Hazel Overview

Meet Hazel, your personal housekeeper. Hazel helps you reduce clutter and save time by automatically moving, sorting, renaming, and performing various other actions on the folders that tend to accumulate lots of files (such as your Desktop and Downloads folders). Hazel's extensive capabilities fall into two main categories: Folders & Rules, and Trash Management.

Get Started

If you're new to Hazel, start by reading Installation & Setup, which includes information on licensing and registration, updates, and (if it should ever become necessary) uninstalling Hazel.

Then move on to Hazel Basics to become acquainted with Hazel's interface and its main features.

Folders & Rules

Hazel can monitor whatever folders you choose, and then automate the management and organization of the items within them. For each folder, you create one or more rules (much like the rules in Apple Mail) that identify items in the folder by their characteristics (name, size, date added, and so on) and then take actions when an item matches (move, copy, rename, add tags, or whatever you need). Hazel can even monitor Smart Folders (with minor limitations), which means it can act on any file or folder on your Mac that follows a well-defined pattern.

To learn what rules can do and how all the pieces of a basic rule work together, see About Folders & Rules. You can then explore the details in Work with Folders & Rules, including how to Manage Folders, Create & Edit Rules, and Manage Rules. Once you have Hazel up and running with a selection of simple rules, you can delve into more advanced features in Attributes & Actions and Advanced Topics.

Trash Management

In addition to letting you manage any number of user-defined folders with rules of your own creation, Hazel provides two automated ways to handle a very special folder: Trash. Hazel can do two special things with your Trash:

- **Automatic deletion:** When you drag a file or folder to the Trash, it isn't deleted, just moved aside. Ordinarily, it's only when you empty the Trash (by choosing Finder > "Empty Trash") that your Mac frees up the space that was occupied by

those items. But Hazel can selectively and automatically delete items from your Trash (thus recovering the space they used) based on how long the items have been in the Trash or the total size of the Trash folder—meaning you never have to empty the Trash manually again. See Use Automatic Deletion.

- **App Sweep:** You can drag an application to the Trash just like any other file, but because apps often store settings, support files, and other resources in various other places, simply deleting the app itself can leave lots of clutter behind (possibly including programs that continue running in the background). Hazel's optional App Sweep feature watches the Trash for applications and other software. When you throw away a piece of software, App Sweep checks for other support files and offers to throw them away as well. See Use App Sweep for more details.

What's New in Hazel 6

Hazel version 6 has the same core features as version 5, but adds new capabilities.

On-the-Fly Text Recognition

A rule can examine the text within graphics and PDFs if they have not been OCRed. See the "Contents" topic in Attribute Reference.

Read Password-Protected PDFs

A rule can open password-protected PDFs as long as you supply the password (one per rule) in advance. See the "Contents" topic in Attribute Reference.

Revert Changes to Files

If a rule has made certain modifications to a file or folder (such as renaming it or changing tags) and you want to undo those changes, you can now do so via Finder's contextual menu. See Revert a File.

Custom List Attributes

A new Custom List attribute enables a rule to create a list based on matching attributes (such as tags or contents) and then reuse that list in an action. See Custom List Attributes.

Lock and Unlock Files

An action can now lock or unlock a file. See the "Toggle lock" topic in Action Reference.

Pause Action

You can now insert a pause for an arbitrary period of time in a sequence of rule actions. See the “Pause” topic in Action Reference.

Improved Number Matching

The “number” attribute in match patterns can now include thousands separators, decimal points, or both (see Using Match Patterns in Conditions). On output, thousands separators can be configured on a case by case basis while decimal values are passed to the output (see the “Attribute Formatting” topic in Using Pattens in Actions),

Maximum Length

You can now set the maximum length of an attribute (such as a filename) in an action. See the “Attribute Formatting” topic in Using Patterns in Actions.

Easier Debugging

If you try to save a rule that has one or more errors, Hazel now highlights the problematic conditions or actions. See the “Create a Rule” topic in Create a Rule.

Reset Counters

If you use a counter when renaming items and you later want to reset that counter to zero for a given folder, you can do so with a contextual menu command. See the topic “Resetting a Counter” on the Using the Counter Attribute page.

Rule Preview Additions

Rule preview now shows the file being previewed up top. When previewing contents, contents are scrollable and searchable. Autocomplete will use text from your previewed

document when matching against contents.

Cloud-Only File Options

Starting in Hazel 6.1, you can determine how any folder in a cloud-synced location (such as iCloud Drive or Dropbox) handles cloud-only files with no local copy. See “Folder Options” in Manage Folders.

Installation & Setup

You can install (or uninstall), register, and update Hazel in just a few quick steps. They're mostly self-explanatory, but complete details are provided in the following pages:

- Install Hazel: Download and install Hazel.
- Understand Evaluation and Demo Modes: Learn the limitations of running Hazel without a full license.
- Purchase a Hazel License: Buy a license to permanently unlock Hazel's full feature set (if you haven't already).
- Register Hazel: Add your license file to Hazel.
- Update Hazel: Make sure you're running the latest version, and configure automatic updates.
- Uninstall Hazel: Remove Hazel and all its components if the need arises.

Install Hazel

Hazel 6 requires a Mac running macOS 12 (Monterey) or later.

To install Hazel:

1. If you haven't already done so, download the latest version of Hazel from the Noodlesoft website.
2. Double-click the disk image file you downloaded.
3. Read the license terms and click "Agree" to accept them.
4. Double-click the "Hazel" icon.
5. An "Install Hazel" alert asks where you want to install Hazel. Click Install in Applications to install Hazel in the default location (Macintosh HD Applications), or click Select Location to choose a different location.
6. An "Install Cleanup" alert asks whether you want to eject the installer disk image and move it to the Trash. Click Yes or No, as you prefer.
7. The Hazel app opens. If you previously had an older version of Hazel installed, you may see one or more alerts stating that Hazel would like to access files in certain folders. Click OK for each one to grant permission.
8. If Trash functions are enabled, a "Trash Access" alert appears, informing you that you must grant Hazel full disk access to use its Trash-related features. To do this:
    - Click "Open System Settings." Make sure you're in System Settings (System Preferences in Monterey) > "Privacy" > "Full Disk Access."
    - In Monterey, click the lock icon in the lower-left corner of the window and authenticate to unlock it.
    - Drag the Hazel icon into the list. (If it's already there, but its checkbox is deselected, select the checkbox.)
    - Click "Quit & Reopen" to quit the Hazel app and reopen it so that it has full disk access.
9. A notification appears, asking whether Hazel can display notifications as it runs. Click the Options pop-up menu and choose Allow to permit these notifications.
10. If you are installing Hazel for the first time, an alert informs you about the 14-day free trial and asks if you want to install some sample rules. Click "Yes" or "No," as you prefer.

SEE ALSO

Understand Evaluation and Demo Modes

Purchase a Hazel License

Register Hazel

Update Hazel

Uninstall Hazel

Understand Evaluation and Demo Modes

When you first download and install Hazel, if you do not register, Hazel runs in Evaluation mode. In Evaluation mode, all features except rule export are enabled, allowing you to give Hazel a thorough workout. This mode lasts for 14 days, after which Hazel switches into Demo mode. In Demo mode, Hazel monitors only one folder and only two rules can be active. Import and export functions are also disabled.

To unlock Hazel's full feature set when running in Evaluation or Demo mode, Purchase a Hazel License. After you do so and register Hazel (see Register Hazel), all features are permanently enabled..

SEE ALSO

Purchase a Hazel License

Install Hazel

Register Hazel

Uninstall Hazel

Purchase a Hazel License

You must buy a license to be able to use Hazel's full feature set indefinitely.

To purchase a license from within Hazel:

1. Click the "Purchase Hazel" button in the banner at the top of the main window, or choose Hazel > "License" and click "Purchase."
2. Select which type of license you want to buy (individual or family pack). Click "Next."
3. Fill in your payment information. Make sure your email address is valid and that your spam filter allows email from noodlesoft.com.
4. After confirmation of your payment details, your license will be downloaded automatically and Hazel will register itself.
5. Your receipt and a copy of your license file will also be emailed to the address you provided.

If you want to use PayPal or a coupon code, you must instead purchase Hazel via our website.

To purchase a license on the web:

1. Click the "Purchase Hazel" button in the banner at the top of the main window, or choose Hazel > "License" and click "Purchase."
2. Click "Go to Web Store."
3. Follow the instructions in your browser for specifying the product and entering your payment information.
4. After confirmation of your payment details, a license file will be emailed to you. The license will come as an attachment, so make sure your spam filter allows it. If your spam filter has such a feature, you should add noodlesoft.com to your whitelist.
5. To register Hazel, you can double-click the license file, drag it into Hazel's License window, or save the license file to disk and add it via the "Add License" button.

SEE ALSO

Register Hazel

Install Hazel

Understand Evaluation and Demo Modes

Register Hazel

If you have received a license, you can unlock Hazel's full feature set. (For instructions on purchasing a full license, see Purchase a Hazel License.)

To register Hazel, choose Hazel > "License" and do one of the following:

- Double-click the license file, or
- Drag the license file into the License window, or
- Click Add License, navigate to the license file, and click Open.

After you have successfully registered, Hazel displays your name in the "Licensed To" field. In Evaluation Mode (see Understand Evaluation and Demo Modes), Hazel displays the expiration date as well.

SEE ALSO

Update Hazel

Install Hazel

Understand Evaluation and Demo Modes

Purchase a Hazel License

Update Hazel

Update Hazel

The quickest best way to install a newer version of Hazel is to use the built-in updating system.

To update Hazel to the latest version:

1. Do either of the following:
    - Choose Hazel > “Check for Updates.” When Hazel checks for an update, it connects to the Noodlesoft site. The only information it transmits is the version you are currently running. This is used to tailor the release notes appropriately.
    - Choose Hazel > “Settings” (“Preferences” in Monterey), click “Advanced,” and click Check Now.
2. If an update is found, Hazel displays release notes and asks if you would like to download and install the new version. You can then do any of the following:
    - To upgrade immediately, click “Install Update.” Hazel downloads and installs the upgrade for you.
    - If you don’t want to upgrade right now, click “Remind Me Later.” Hazel will remind you based on your automatic update settings.
    - If you don’t want to upgrade to this version at all, click “Skip This Version.” Hazel won’t remind you of an update until a newer version has been released.

To configure Hazel to update itself automatically:

1. Choose Hazel > “Settings” (“Preferences” in Monterey) and click “Advanced.”
2. Choose “Daily,” “Weekly,” or “Monthly” from the “Check for updates” pop-up menu. (To turn off automatic checking, choose “Never.”)
3. Optionally check the “Include anonymous system profile” option to send your system profile whenever you check for updates; this provides information that helps us decide which operating system versions and features to support in the future. Click “What’s this?” to learn more about this option.

SEE ALSO

Uninstall Hazel

Install Hazel

Register Hazel

Uninstall Hazel

Hazel does not require or include an uninstaller. To remove it from your Mac, drag the Hazel app from the Macintosh HD Applications folder to the Trash.

SEE ALSO

Hazel Basics

Install Hazel

Purchase a Hazel License

Register Hazel

Update Hazel

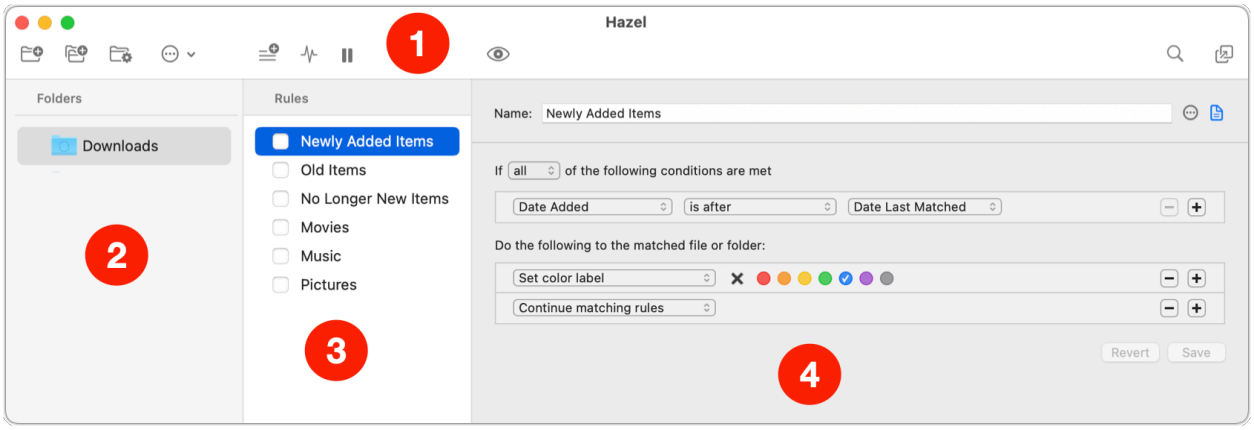
Hazel Basics

Hazel is a bit like chess in that it's easy to learn the basics, yet you can spend years mastering every last subtle nuance. To start on the right foot, read the following topics to acquaint yourself with Hazel's main features and user interface:

- The Main Hazel Window: Discover the major features in Hazel's main window.
- About Folders & Rules: Learn what Hazel's rules can do, and see the anatomy of a simple rule.
- The Hazel Status Menu: Discover how to enable, disable, and use Hazel's optional system-wide status menu.
- Stopping & Restarting Hazel: Prevent Hazel from running any rules (if needed for testing or troubleshooting) and then restart the rules when you're ready.
- Notifications: Learn how Hazel can alert you to matching rules using Notification Center.

The Main Hazel Window

The primary place you'll interact with Hazel is in its main window.



Here are the major components of the Hazel window:

- **Toolbar**①: The icons here control major Hazel functions. There are ten icons in three groups.
  - Tools for working with folders:
    - **Add Folder:** Click this icon to add a new folder for Hazel to monitor. See Manage Folders.
    - **New Group:** Click this icon to create a new folder group (to organize monitored folders). See Manage Folders.
    - **Folder Options:** Access the two "Throw away" options, just like choosing Folder > "Folder Options." See the "Set Folder Options" topic in
    - **Action:** Use this pop-up menu to import, export, sync, pause, and resume rules, and to adjust folder options. See Manage Rules.
  - Tools for working with rules:
    - **New Rule:** Click this icon to add a new rule to the selected folder. See Create a Rule.
    - **Rule Status:** Click this icon to display a window showing which rules have applied and when. See Show Rule Status.
    - / **Pause/Resume Rules:** Click this icon to pause (or resume)

rule processing for the selected folder. See Enable, Disable, or Pause Rules.

- **Preview Rule**: Enable preview mode to see whether a rule will match a selected item, just as if you had chosen Rule > “Preview Rule.” See Preview a Rule.

- Other tools:
  - **Search:** Click this icon and type a search term to filter your list of rules. See Search Rules.
  - **Detach Rule Editor:** Click this icon to display the rule editor in an independent floating window. See Detach the Rule Editor.
- You can also optionally add two more tools that don’t appear by default. Choose View > “Customize Toolbar” and drag either of these to the desired location:
  - **Rule Sync:** Click this to set up syncing for the currently selected folder; equivalent to choosing Folder > “Rule Sync Settings.” See Sync Rules.
  - **Run Rules Now:** Run the rules for the selected folder now, which works the same as choosing Rule > “Run Rules Now.” See Run Rules Manually.

- **“Folders” list**②: This list contains all the folders you’ve asked Hazel to watch. By default, it contains only one folder—Downloads. (Note: You can add a folder to Hazel without assigning any rules to it, but every rule you create must be attached to a folder.)
- **“Rules” list**③: This list contains all the rules you’ve created (or imported) for the currently selected folder. Select the checkbox next to a rule’s name to enable it; deselect it to disable that rule. (In this example, all rules are disabled.)
- **Rule editor④:** This pane is where you define the logic of your rules—what action(s) occur in response to what condition(s). See Create a Rule.

SEE ALSO

About Folders & Rules

The Hazel Status Menu

Hazel Settings

About Folders & Rules

Hazel lets you automate the management and organization of your files and folders. You can ask Hazel to watch any number of folders, and for each one, you create one or more rules that detail what should happen to the folder's contents and under what circumstances.

Hazel can be used to organize almost any folder, but the best candidates are ones that tend to collect files, such as where your browser downloads files, where your email app puts attachments, or a shared Dropbox folder. Hazel can also use Smart Folders (with minor limitations).

What Rules Can Do

Rules in Hazel, much like rules in Apple Mail, specify two things:

- What to look for in the selected folder (*conditions*)
- What to do when the conditions are met (*actions*)

You can have Hazel watch any number of folders; and for each folder, you can create any number of rules. Each rule, in turn, can have one or more *conditions* that, when met, trigger one or more *actions*.

Conditions

A Hazel rule can match a vast range of conditions for both files and folders—practically anything you can think of. To give just a few examples, a condition can examine attributes such as:

- The name or size of a file or folder—for example, "Filename contains *cherry*" or "Size is greater than 100 MB"
- The creation or modification date of a file or folder—for example, "Date Created is in the last week" or "Date Last Modified occurs after 3:00 PM on a weekend in January or March"
- The text contents of a file—for example, "Contents do not contain *virtual machine*" or "Contents contain match *any word followed by a three-digit number*"
- Metadata such as Finder tags and comments—for example, "Tags contain *Work*" or "Comment is not blank"
- The number of items (files or folders) in a subfolder—for example, "Sub-file/

folder Count is less than 5”

- Normally hidden Spotlight metadata for photos, music, and other media files—for example, “Resolution width is less than 300 (DPI)” or “Composer contains *Bach*”

And, multiple conditions can be combined in any way you like—a rule can be set to match any, all, or none of the conditions you specify; and you can even nest conditions to create complex logical tests.

Actions

When your rule matches the condition(s) you specify, it then takes one or more actions. Again, the range of options is immense, but here are a few examples:

- Move or copy the matched item to another location, or upload it to a server
- Sort the matched item into a subfolder
- Rename the matched item
- Zip or unzip the matched item
- Change metadata such as Finder tags and comments
- Import the item into Photos or Music
- Display a notification
- Run an AppleScript, JavaScript, Automator workflow, Shortcut, or shell script

Example Rules

Putting conditions and actions together, here are a few examples of complete rules you can create in Hazel:

- When a family member puts a photo in a shared Dropbox folder, import it into Photos, delete the original, and display a notification.
- When a PDF file appears on my Desktop, move it into my Documents folder and then open it in Preview.
- When my Downloads folder contains a `.dmg` file that’s larger than 100 MB and older than one week, move it to the Trash.
- When I put a Word file in my Research folder that contains the text “earnings report,” apply the Finder tag “Finances.”
- When a scanned document appears in my Images folder, run an AppleScript that opens it in Nitro PDF Pro and performs OCR (optical character recognition)

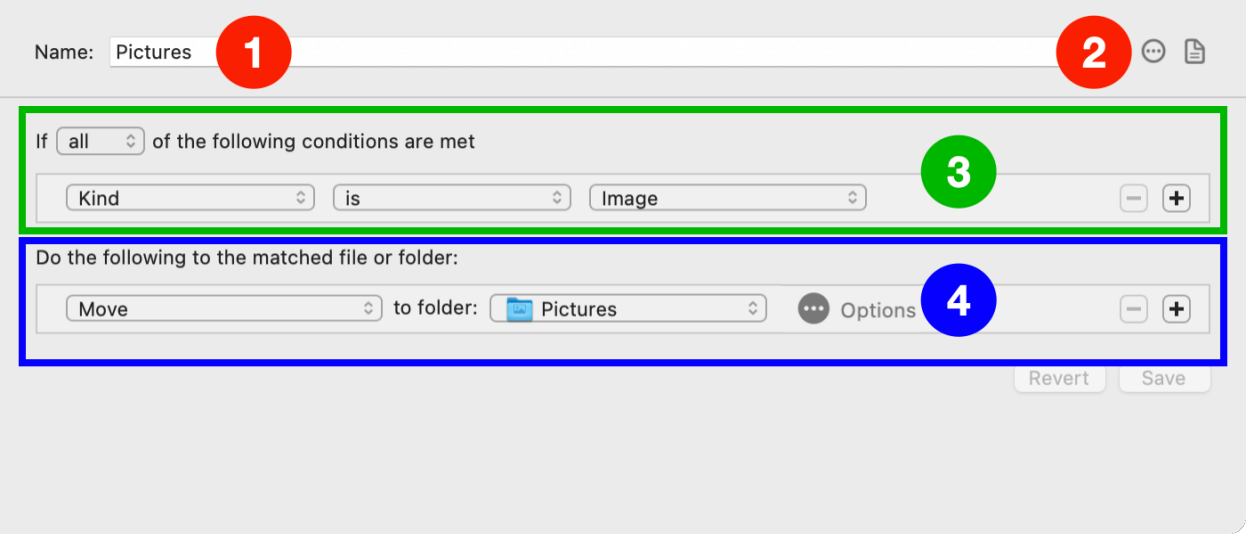
on it.

In cases where a folder has numerous rules, or the conditions and actions are complex, it can take some planning and experimentation to achieve exactly your desired results. For a detailed explanation of Hazel's logic when processing rules, read Understand the Logic of Rules.

Anatomy of a Simple Rule

To illustrate how Hazel rules are constructed, let's look at a simple example. This happens to be one of the sample rules you can install by choosing Help > "Load Sample Rules."

This example shows the "Pictures" rule (which is meant to be attached to the Downloads folder, not shown):



This rule looks for files in the Downloads folder with a kind of "Image" and, for each one it finds, moves it to the Pictures folder (that is, Macintosh HD ▸ Users ▸ your-username ▸ Pictures).

Here's what each element does:

- **Name①:** The rule's name can be anything you like. This one is simply called "Pictures."
- **Rule Options and Note②**:
    - Click the rule options ⋯ icon to display settings for text recognition in images or PDFs, and passwords for PDFs (see the "Contents" heading in

Attribute Reference).

- If you want to make notes or comments about the rule—perhaps explaining what it does or why you constructed it a certain way—click the Note icon, enter the note, and then click the X icon (or just click anywhere outside the note bubble). When a rule includes a note, the Note icon turns blue.

- **Conditions③:** The conditions area, outlined in green in the image above, specifies what has to be true in order for the rule to be triggered (and thus for the action(s) to run). This rule has only one condition, created by choosing options from a series of pop-up menus: “Kind is Image.” (We could add more conditions by clicking the plus button in the Conditions section.) That is, if the rule encounters a file in the Downloads folder with a kind of “Image,” the condition is true and the action applies to that file.

- **Actions④:** The actions area, outlined in blue above, specifies what happens to each file or folder for which the condition(s) above have been met. This rule has only one action (again, constructed by choosing options from a series of pop-up menus): “Move to folder Pictures.” (We could add more actions by clicking the plus button in the Actions section.)

  Notice the Options icon next to the action. Some actions offer additional parameters; click this icon to adjust them. (For the “Move” action, options let you specify what happens if a file with the same name already exists at the destination, whether to throw the file away if it’s a duplicate, and whether or how to copy the folder structure when moving a folder.)

Some conditions and actions have more or fewer pop-up menus, require you to fill in blanks, or let you build patterns of various kinds. But they all follow this basic structure, and every condition and action starts with a choice from the leftmost pop-up menu in its row.

SEE ALSO

Work with Folders & Rules

Attributes & Actions

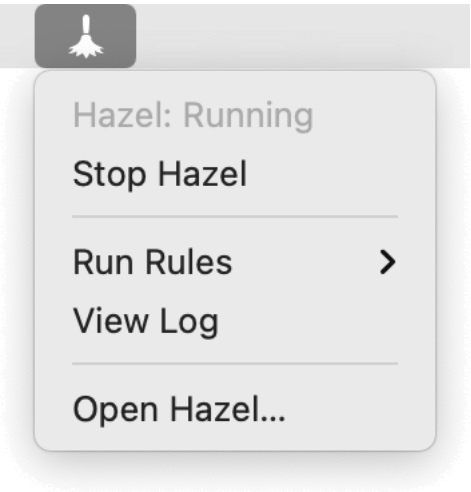
The Hazel Status Menu

Hazel's menu bar icon gives you immediate indication of its status, plus quick access to some of its functions, such as starting/stopping and running rules manually.

The icon has two states to indicate Hazel's status:

- A dark icon means Hazel is running.
- A light icon means Hazel is stopped.

Click the icon to display a menu that lets you quickly access other functions:



The status menu is enabled by default. To enable or disable it, go to Hazel > "Settings" > "General" and select or deselect "Show Hazel in the menu bar."

Note: The Hazel status menu does *not* have to be running for Hazel rules to function.

SEE ALSO

Stopping & Restarting Hazel

Run Rules Manually

Stopping & Restarting Hazel

Normally, Hazel is always watching your folders and running your rules as needed. Sometimes, though, you may need to stop Hazel temporarily—for instance, if you want to create some experimental rules and preview them before they run for real, or if you're trying to troubleshoot a problem that could be caused by a misconfigured rule.

To stop Hazel, do either of the following:

- Choose Hazel > "Stop Hazel."
- If the Hazel status menu is showing, choose "Stop Hazel" from that menu.

The icon should dim to reflect Hazel's stopped state. While Hazel is stopped, rules do not run, and no Trash processing occurs.

To start Hazel again, do either of the following:

- Choose Hazel > "Start Hazel."
- If the Hazel status menu is showing, choose "Start Hazel" from that menu.

The icon should darken to reflect Hazel's running state.

You can also enable or disable any individual rule, or pause all the rules for a given folder; see Enable, Disable, or Pause Rules.

SEE ALSO

Notifications

The Hazel Status Menu

Notifications

When Hazel needs to get your attention, it can use Notification Center to do so.

The types of notifications Hazel may send are as follows:

- **Errors:** Alerts when problems occur while processing rules.
- **File Events:** Alerts when a rule's actions change a file significantly (for example, moving, copying, archiving, or unarchiving). File events apply only to built-in actions, not to script actions.
- **Trash Events:** Alerts when Hazel deletes something permanently from the Trash.
- **Custom Messages:** User-defined alerts that appear when a rule uses the "Display notification" action.

You can disable any sort of notification you prefer not to receive.

To configure Hazel for notifications:

1. In Hazel, go to "Hazel" > "Settings" > "Advanced." (Note that in Monterey, this menu is called "Preferences.")
2. Select the checkbox(es) for the type(s) of notifications you want Notification Center to display—"Errors," "File Events," or "Trash Events."
3. If you have not already enabled Hazel's notifications in System Settings (System Preferences in Monterey), go to System Settings > "Notifications," select Hazel Notifier in the list, and turn the "Allow notifications" switch on. You can also set other notification parameters for Hazel here, as you can for any app.

To have a Hazel rule send a custom message using Notification Center, select the "Display notification" action (see Action Reference for further details).

SEE ALSO

Work with Folders & Rules

Hazel Settings

When you choose Hazel > “Settings” (“Preferences” in Monterey), a window appears with three categories of settings. Click “General,” “Trash,” or “Advanced” to display a category.

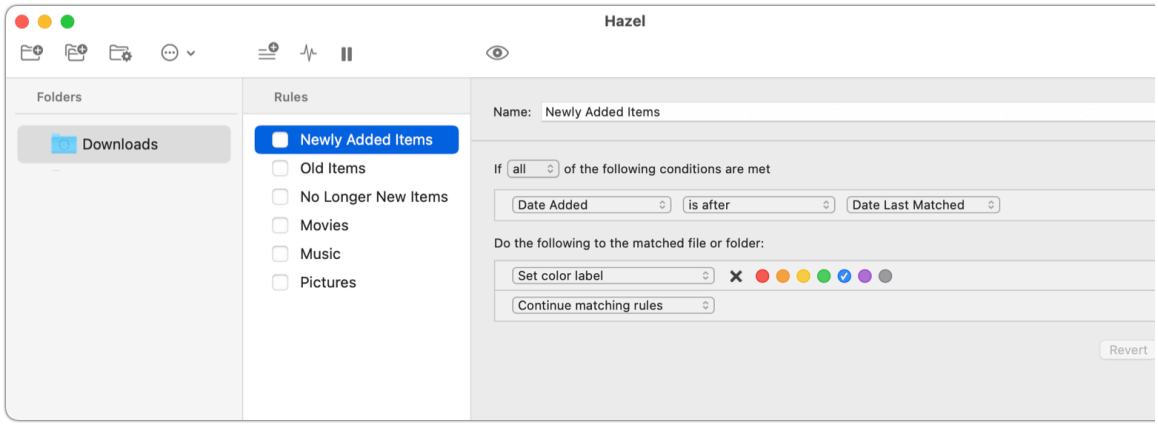
General Pane

The General pane contains controls for layout and displaying or hiding the Hazel status menu.

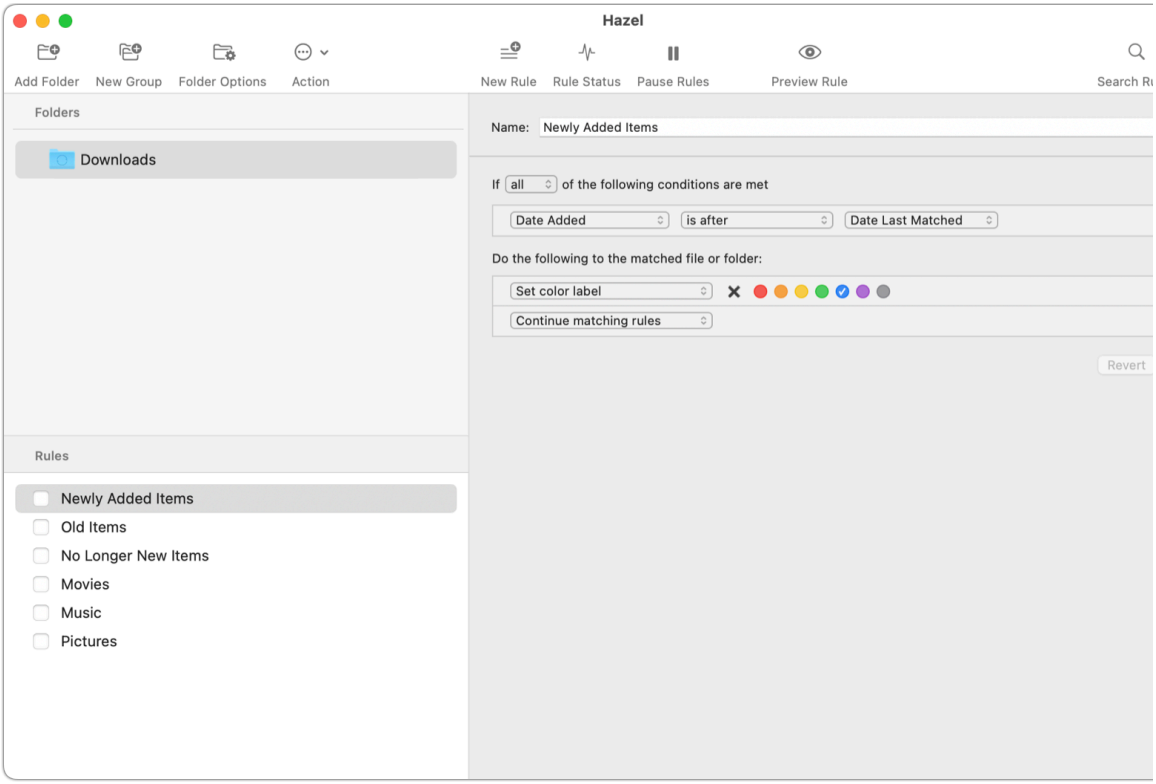
Layout

Click an icon to switch Hazel’s display to use any of three layout options:

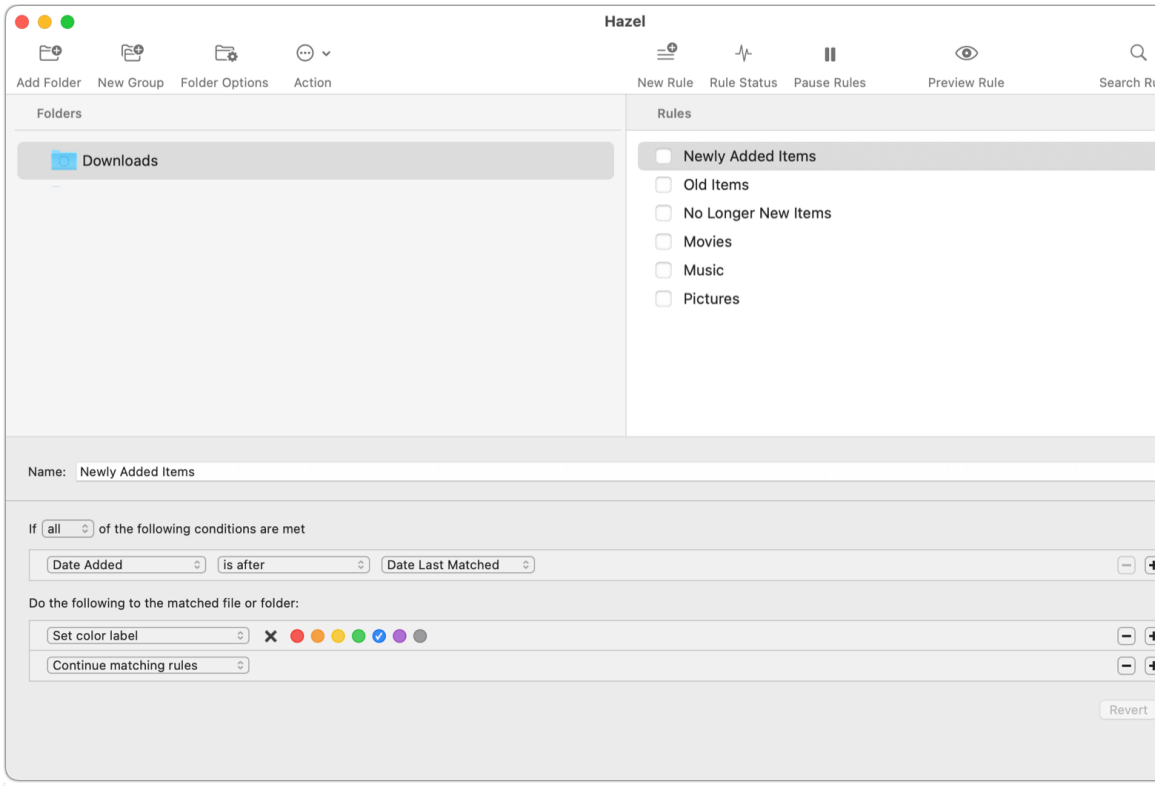
- **Three columns (default):** Show the Folders list on the left, then the Rules list, with the Rule Editor on the right.



- **Two columns:** Show the Folders list above the Rules list, with the Rule Editor on the right.



- **Two rows:** Show the Folders list on the left, the Rules list on the right, and the Rule Editor at the bottom.



Show Hazel in the Menu Bar

Show or hide the Hazel status menu. See The Hazel Status Menu.

Trash Pane

The Trash pane contains controls related to Hazel's Trash features, including automatically removing items from the Trash when certain criteria are met and deleting support files belonging to apps that you drag to the Trash. See Manage Your Trash for details.

Advanced Pane

The Advanced pane contains controls related to notifications (see Notifications) and updates (see

SEE ALSO

Update Hazel

Notifications

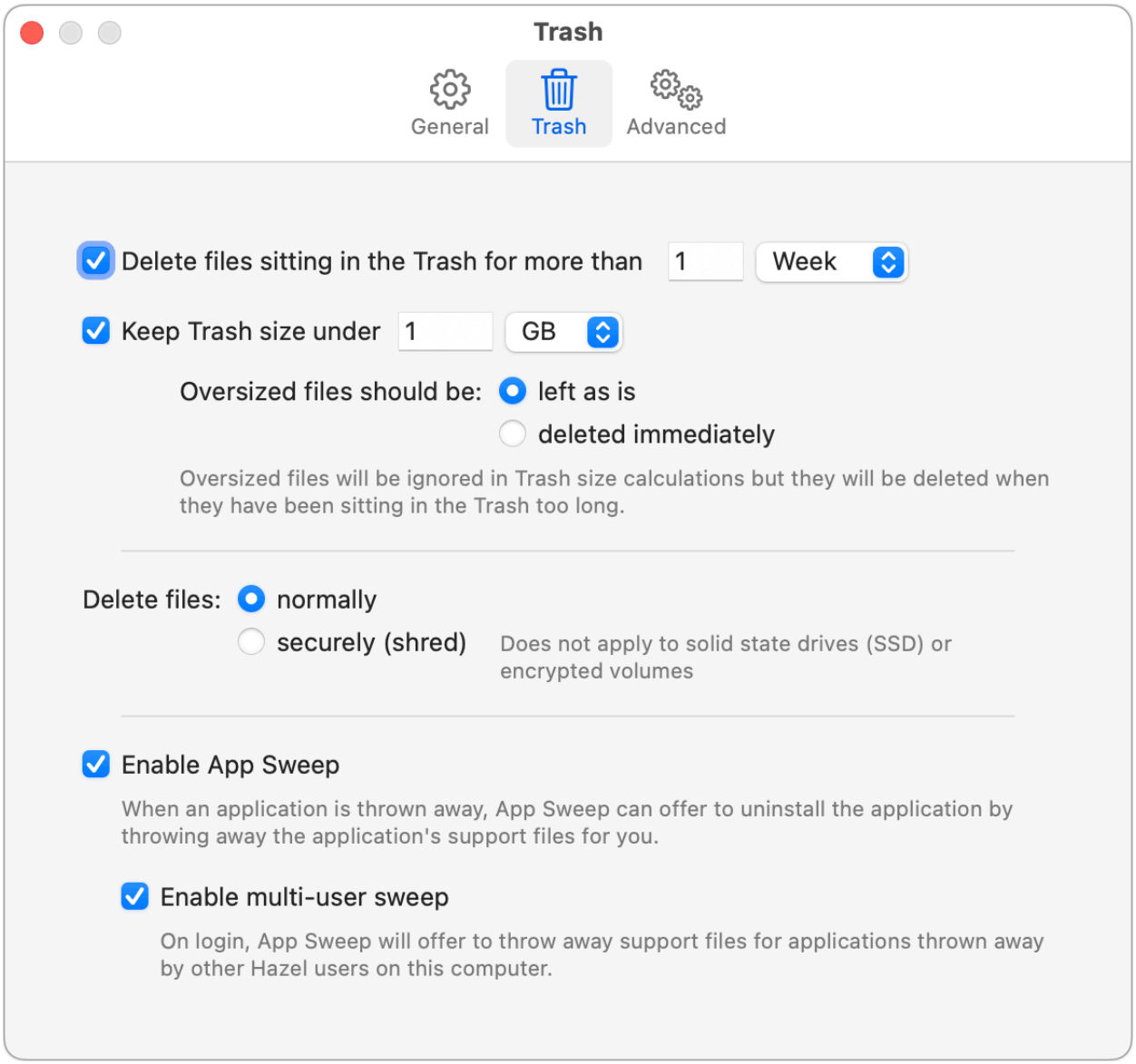
The Hazel Status Menu

Manage Your Trash

Manage Your Trash

Hazel's Trash setting pane offers two tools to help you manage your Trash. They're described on the following pages:

- Use Automatic Deletion: Let Hazel delete files from your Trash based on age or size.
- Use App Sweep: Hazel can uninstall support files for apps you drag to the Trash.

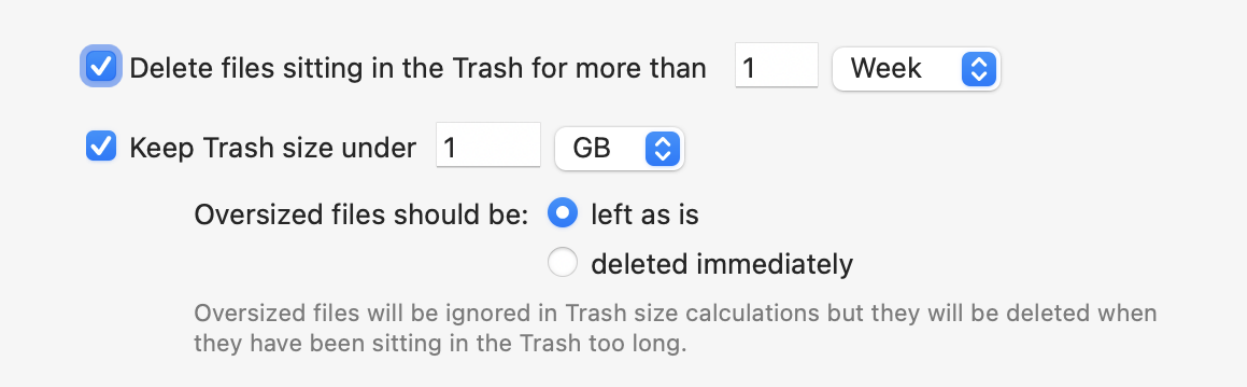


Use Automatic Deletion

When you drag a file or folder to the Trash, it isn't deleted, just moved aside. Ordinarily, it's only when you empty the Trash (by choosing Finder > "Empty Trash") that your Mac frees up the space that was occupied by those items. But Hazel can selectively and automatically delete items from your Trash (thus recovering the space they used)—meaning you never have to empty the Trash manually again. Hazel can use either or both of two criteria to determine which items should be deleted from the Trash:

- **How long they've been in the Trash:** Hazel can record how long each file has been in the Trash and then delete it after whatever time delay you set.
- **Total Trash size:** Hazel can monitor the total size of all files in the Trash and delete files as necessary (in the order in which they were moved to the Trash) to keep that total under whatever limit you set.

  Important: Note that deletion here means *permanent* deletion, so it is wise to set the limits longer and higher than you think you need when you first try it out.



To enable automatic deletion:

1. Go to Hazel > "Settings" > "Trash."
2. To have Hazel automatically delete items from the Trash based on how long they've been there, select the "Delete files sitting in the Trash for more than" checkbox, fill in a number, and choose a time period ("Hours," "Days," "Weeks," "Months," or "Years") from the pop-up menu.
3. To have Hazel automatically keep the Trash under a certain size, select the "Keep Trash size under" checkbox, fill in a number, and choose a unit of measurement ("MB," "GB," or "% of disk capacity") from the pop-up menu. With "Keep Trash size under" selected, you can also choose how Hazel handles oversized files—that is, any file dragged to the Trash that, all by itself, exceeds

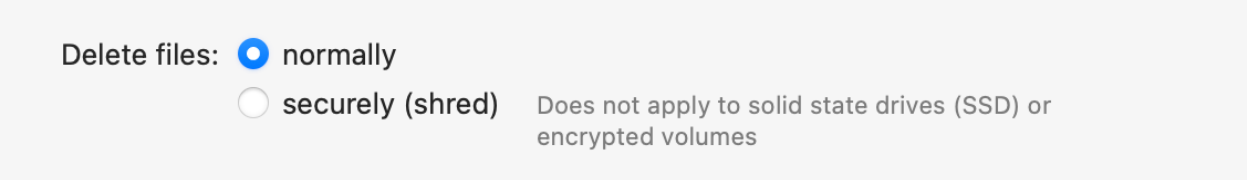
your specified size limit:

- **left as is:** The file will be left in the Trash and Hazel will ignore it when calculating the total Trash size (otherwise, the Trash would always be over the limit). If you also have the “Delete files sitting in Trash for more than” option selected, the file will be deleted when its time limit is reached. Otherwise, it will be removed only if you manually empty the Trash.
- **deleted immediately:** The file will be deleted immediately and permanently.

Choose normal or secure deletion:

Normally, when files are deleted, their data is still intact on your disk, and macOS merely marks the space they occupy as being available for reuse. Until another file overwrites that space, the file could potentially be retrieved by disk recover or undelete tools. So Hazel offers a 7-pass secure deletion option, which overwrites deleted files to make them unrecoverable.

To use normal deletion, select “normally” in the “Delete files” area. To delete files securely, select “securely (shred).”



However, please note the following:

- Secure deletion does not take place on SSDs and other solid-state storage. That’s because SSDs do not correctly process secure erase commands—from the user’s perspective, they appear to work, but the data is not in fact securely overwritten (or not *entirely* overwritten). Therefore, even if you select “securely (shred),” it’s possible that your deleted files will still be recoverable.
- Secure deletion does not take place for encrypted drives. As the files are already encrypted, they should not be recoverable.
- Secure deletion takes much longer than normal deletion, and can slow down your Mac while it’s in progress.

SEE ALSO

Use App Sweep

Use App Sweep

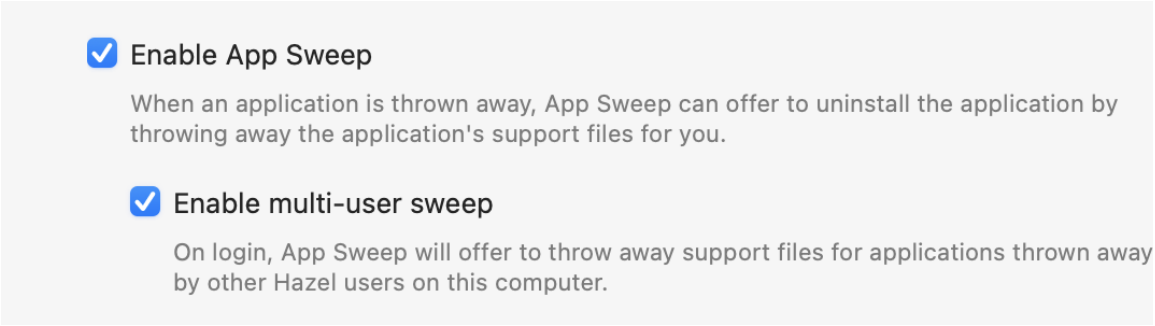
You can drag an application to the Trash just like any other file, but because apps often store settings, support files, and other resources in various other places, simply deleting the app itself can leave lots of clutter behind (possibly including programs that continue running in the background). Hazel's optional App Sweep feature watches the Trash for applications and other software. When you throw away a piece of software, App Sweep checks for other support files and offers to throw them away as well.

Note that if the app in question has its own uninstaller, you should use that uninstaller instead, because removing certain apps may require special steps that only the uninstaller knows about. In general, if you installed the app by dragging it to your disk (as opposed to running an installer program), it's a good candidate for App Sweep.

About Multi-user App Sweep: App Sweep can also uninstall support files from within other users' home folders on your Mac. If you select the "Enable multi-user sweep" option, Hazel keeps track of apps you have thrown away. When other users on your Mac log in, Hazel gives them the option to throw away support files for those apps, as if they had thrown away the apps themselves. This feature works only for user accounts that have mutually enabled Hazel's multi-user sweep option. (That is, if either of two accounts has this feature disabled, the other account will never be prompted to delete support files from apps the other user has uninstalled.)

To enable App Sweep:

1. Go to Hazel > "Settings" > "Trash."
2. Select the "Enable App Sweep" checkbox.

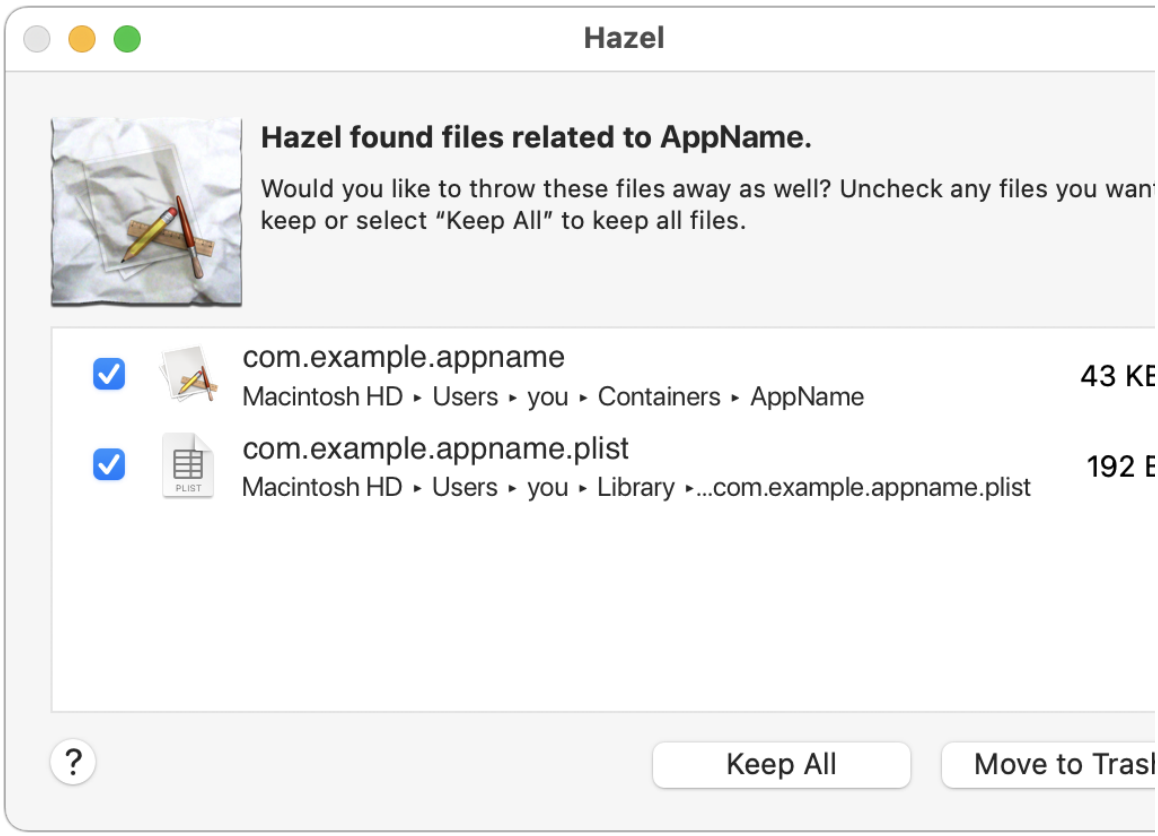


3. To enable App Sweep for all Hazel users on your Mac (as described just above), also select the "Enable multi-user sweep" checkbox.

Once App Sweep is enabled, it watches the Trash and activates itself if it notices you've thrown away an application.

To uninstall an application:

1. Drag the application to the Trash.
2. If there are any support files for the app you've dragged to the Trash, Hazel displays them in a dialog:



3. If you want to save any of the individual files, uncheck them; or, if you want to save all of them, click "Keep All."
4. Click "Move to Trash" to throw away the selected support files.

If you want to reinstall the app (before the Trash is emptied or automatic deletion removes the files), open the Trash, select the app, and choose File > "Put Back" (⌘-Delete). Hazel will offer to reinstall any support files it threw away previously. (If for any reason you need to salvage just the support files manually, you can find them in the Trash listed under the application's name.)

SEE ALSO

Use Automatic Deletion

Work with Folders & Rules

If you've been reading the help sequentially, you already know the basics of Hazel's capabilities and how simple rules work. (If not, refer back to About Folders & Rules.) This next set of topics goes into much greater detail about folders and the rules that operate on them:

- Manage Folders: Learn how to add App Folders (such as those used for browser downloads), conventional folders, and Smart Folders to Hazel; how to remove folders; and how to set folder options.
- Create & Edit Rules: Get to know the logic Hazel employs when processing rules. Create your first rules; preview a rule before applying it to a folder; copy, move, edit, or delete rules; use Hazel's predefined sample rules; and search or filter your rules.
- Manage Rules: Enable or disable individual rules, pause rules for a whole folder, run rules manually, see which rules have matched items in a given folder, import and export rules, and sync rules to a file (so they can be shared with other users or Macs).

Manage Folders

By default, Hazel shows just one folder—Downloads (Macintosh HD ▸ Users ▸ your-username ▸ Downloads)—but you can add as many folders as you like. For each folder, you can then create one or more rules to monitor the folder and perform the actions of your choice.

Adding a folder, by itself, causes nothing to happen, so feel free to add any folders you think you'd like to monitor and then add rules at your leisure.

Add a Folder

Hazel can be used to organize almost any folder, but the best candidates are ones that tend to collect files, such as where your browser downloads files, where Mail puts attachments, or a shared Dropbox folder. Hazel can also use Smart Folders (with minor limitations).

Because folders associated with downloads are among the most common ones to monitor, Hazel offers a shortcut to some of the most common ones, known as app folders. When you begin configuring Hazel, we suggest starting with folders in this category, and then moving on to conventional folders and Smart Folders.

Add an App Folder

To add an app folder, go to Folder > "Add App Folder" and choose an application's download folder. (Note that apps won't appear in this list unless they're installed on your Mac, and even then, some apps appear only if you've changed their download folder from its default location.) Your choices may include:

- **Safari Download Folder:** The folder where Safari stores downloads. By default, this is Macintosh HD ▸ Users ▸ your-username ▸ Downloads, but you can change it in Safari > "Settings" > "General" > "File Download Location."
- **Mail Download Folder:** The folder in which Mail stores a copy of any attachment you open by double-clicking it in an email message. Like Safari, Mail uses Macintosh HD ▸ Users ▸ your-username ▸ Downloads by default, but you can change it in Mail > "Settings" > "General" > "Downloads folder."
- **Firefox Download Folder:** The folder where Firefox stores downloads (Macintosh HD ▸ Users ▸ your-username ▸ Downloads by default). You can change it in Firefox > "Settings" > "General" > "Downloads".

- **Transmission Download Folder:** The folder where Transmission stores *completed* downloads. You must manually configure this to be a different location from where Transmission stores *incomplete* downloads, so that Hazel doesn't operate on files before they're fully downloaded. To do this, go to Transmission > "Settings" > "Transfers" > "Adding." Check "Keep incomplete files in" and choose (or create) a folder for in-progress files. Then choose a *different* folder from the "Default location" pop-up menu just above it. (Don't use "Same as torrent file," because that doesn't refer to a fixed folder that Hazel can track.) The app folder link will then point to Transmission's "Default location," which is where completed downloads are kept.
- **Chrome Download Folder:** The folder where Chrome stores downloads. Like other browsers, Chrome uses Macintosh HD ▸ Users ▸ your-username ▸ Downloads by default. You can change this in Chrome > "Settings" > ""Downloads."
- **Brave Download Folder:** The folder where Brave stores downloads. Like other browsers, Brave uses Macintosh HD ▸ Users ▸ your-username ▸ Downloads by default. You can change this in Brave > "Settings" > ""Downloads."
- **Vivaldi Download Folder:** The folder where **Vivaldi** stores downloads. Like other browsers, **Vivaldi** uses Macintosh HD ▸ Users ▸ your-username ▸ Downloads by default. You can change this in **Vivaldi** > "Settings" > ""Downloads."
- **Opera Download Folder:** The folder where Opera stores downloads. Opera uses Macintosh HD ▸ Users ▸ your-username ▸ Downloads by default, but you can change it in Opera > "Settings" > "Features" > "Downloads."

If you use another app to download files regularly, you can add that location as a conventional folder (discussed just ahead). But note that you should not have Hazel watch a folder in which files download over an extended period of time (as is often the case with P2P and BitTorrent clients). Some such apps have an option to move completed downloads to another location—in such cases, you should have Hazel watch that secondary location.

Add a Conventional Folder

If a folder you want Hazel to watch isn't in the App Folder menu, you can add it manually. Go to Hazel's "Folders" pane and then do either of the following:

- Drag a folder from the Finder into the "Folders" list.
- Click the Add Folder icon on the toolbar. In the dialog that appears, navigate to the folder you want, select it, and click "Open."

Add a Smart Folder

Hazel can also monitor Smart Folders, which are saved Finder searches. (For example, you might search for files with a certain tag that are also over a given size, and save that search so you can find matching files whenever you like. To learn more, see Apple's support article Create or change a Smart Folder on Mac.) This means that Hazel can, in theory, monitor any file or folder, anywhere on your Mac—as long as it matches well-defined search criteria.

Hazel automatically locates all the Smart Folders on your Mac. To add one of these, choose Folder > "Add Smart Folder" > *Smart Folder Name*.

Important: Because Smart Folders represent files that may be located anywhere on your Mac, Hazel rules for Smart Folders cannot descend into subfolders or match against subfiles.

Tip: See a Folder's Full Path

The folder list shows only folder names, not their locations. To see a folder's full path, hover your pointer over the folder (without clicking). The path appears as a tool tip.

To see a folder in the Finder, right-click (or Control-click) the folder name and choose "Show in Finder" from the contextual menu.

Remove a Folder

To remove a folder from Hazel's Folders list, select it and press Delete or choose Edit > "Delete." Read the warning that appears—***removing a folder deletes all its rules and cannot be undone***—and click "Remove" to confirm.

Important: Removing a folder also removes all its rules. If you want to be able to use a rule on another folder, drag it to that folder first. (See Copy or Move a Rule.)

Set Folder Options

Hazel offers a handful of options that affect how items in folders are handled beyond what the rules themselves say. To access these, select a folder and choose "Folder" > "Folder Options."

First, there are two "Throw away" options that apply to the currently selected folder and enable you to delete files you're unlikely to need:

- **“Duplicate files”:** Duplicate files are exact copies of files that were already downloaded to this folder, or that were duplicated in the Finder. These files usually include a number at the end, such as `filename-1` or `filename(1)`. When you select the “Duplicate files” checkbox, Hazel automatically removes duplicate files from this folder if they are identical to the original (the version of the file without the added number).
- **“Incomplete downloads”:** Incomplete downloads occur when a download is aborted or interrupted. Hazel can automatically move these files to the Trash after the time period you set here. (Be sure to set a time period after which you’d be unlikely to resume a download.)

Next, for folders within iCloud Drive, Dropbox, or other cloud-synced locations, there’s an additional control that determines what Hazel should do if a matched file is stored only in the cloud, with no local copy:

- **Ignore:** Don’t apply rules to cloud-only files at all.
- **Process normally:** Rules apply as with local files, but rules that require access to the file contents may fail. This is the default option.
- **Force file to be downloaded:** Download the cloud-only files and then apply the rules. (Local copies of these downloaded files may later disappear again, based on the cloud provider’s settings.)

Note: Folder option controls apply to conventional folders and app folders, but not to Smart Folders.

Use Folder Groups

Folder groups enable you to organize and reduce clutter in your Folders list by grouping folders together in whatever ways make sense for you. Moving a folder into or out of a group does not change its rules or other settings, nor can you assign rules to folder groups—each rule must still be assigned individually to a folder.

To create a folder group:

1. Choose Folder > “Add Folder Group” (Control-Shift-A) or click the Add Folder Group icon on the toolbar.
2. Type a name for your folder group and press Return.

To use a folder group:

- Drag any folder underneath a group name to put it in that group, or drag it out to

remove it. A horizontal line shows where the folder will go.

- Use the disclosure > icon to expand or collapse the contents of a folder group.

  Note: Groups can be nested within other groups.

To rename a folder group:

Click a group name, wait a moment until it becomes editable, type a new name, and press Return.

To delete a folder group:

Hover over the group's name and click the Remove ⊖ icon.

Important: Deleting a folder group does *not* remove any of the folders within it —only the group itself. The contents of the group move up to the level of the now-deleted group.

Note: Hazel now displays each folder group as its own submenu on the Run Rules submenu of the status menu. See Run Rules Manually.

SEE ALSO

About Folders & Rules

Create & Edit Rules

Create and Edit Rules

Rules are Hazel's central and most important feature. These topics describe how to work with rules in various ways:

- Understand the Logic of Rules: Learn about the internal logic Hazel employs when evaluating rules (and how to alter its default behavior when needed).
- Create a Rule: Set up a simple rule consisting of one or more conditions and one or more actions.
- Copy or Move a Rule: Copy or move a rule to a different folder, where you can then customize it further if needed.
- Edit or Remove a Rule: Alter an existing rule, or delete a rule you no longer need.
- Use Sample Rules: Add Hazel's built-in sample rules to any folder, learn what they do, and enable or edit them.
- Search Rules: Look for rules matching any text string, and filter the display to show only those rules.

Understand the Logic of Rules

In the simplest case, where a folder has only one rule and that rule has only one condition and one action, Hazel's logic is pretty self-explanatory: either the rule matches a file or folder and runs its action, or it doesn't. However, as you create more (and more-complex) rules, you may encounter ambiguous situations in which Hazel behaves in an unexpected way. It can help to understand the logic by which Hazel applies rules.

The basic process is this:

- Hazel tests the conditions of its first rule against the first file or folder in the monitored folder.
    - If the conditions match, Hazel runs the action(s) of that rule, in order, *and then moves on to the next file or folder*.
    - If the conditions do not match, Hazel tests the conditions of its next rule against the first file or folder, and so on until it has tried all the rules.
- Hazel continues like this until it finds a match or runs out of items to check in the monitored folder.

You can alter this default behavior in certain ways if you need to achieve a particular effect.

Here are further details about Hazel's logic, and examples of each one, to help guide your efforts:

- **Hazel rules operate equally on both files and folders.**
  In general, files and folders are treated the same way in both conditions and actions. There are, however, a few exceptions—for example, it's possible to specify a rule condition in such a way that it applies only to files or only to folders, and the "Run rules on folder contents" action applies only to folders.
- **Only the first matching rule runs on any given file or folder.**
  Suppose you have two rules for a folder. The first looks for file names starting with "A" and tags them Red. The second looks for files with a .png extension and opens them in Preview. Now you put a file named "Apple.png" in the monitored folder. The first rule matches (because it starts with "A") and applies the Red tag. But the second rule does *not* run, because a rule has already matched the file. On the other hand, if you put a file named "Banana.png" in the monitored folder, only the second rule will run—it will open the file in Preview.

  If you need *both* rules to apply, you can add the "Continue matching rules"

action to the first rule. That action tells Hazel to override its normal behavior and process the file through the next rule, even though a match has already occurred. If the next rule matches, however, Hazel won't try any subsequent rules unless those rules also include the "Continue matching rules" action.

Because Hazel's rules operate this way, you should pay careful attention to the order of your rules. If you need multiple actions to apply to complex combinations of attributes, consider using nested conditions.

- **As long as a file is not moved from the monitored folder, Hazel will continue to run that file through its rule list in *subsequent* passes.**
  Let's say you have a rule that looks for files with a creation date more than 3 days ago and, if one is found, it's moved to your Desktop. A newly created file appears in the folder, and the rule does nothing, because its creation date is too recent. But as long as the file is still in that folder 3 days from now, the rule will move it to your Desktop at that point.

- **Hazel checks and acts on only the files and folders at the top level of the monitored folder by default.**
  Imagine a Hazel rule is attached to your Downloads folder, and that rule displays a notification for every newly added file or folder. But inside your Downloads folder is a Research subfolder, and you drag a PDF into that subfolder. No notification will appear, because the file wasn't added to the top level of the folder Hazel is monitoring. As far as the rule is concerned, the contents of that subfolder are invisible.

  If you need Hazel to match against files and folders in subfolders of the main folder, there's a way to do so, but be aware that this is an advanced feature. See Processing Subfolders for instructions.

- **If an action fails, subsequent actions do not run for that file.**
  If a rule has five actions and the third one fails (for example, it encounters a permissions error when copying a file, or a destination folder is missing), actions four and five will not run on that file. However, Hazel will continue trying to run the actions on subsequent passes until it succeeds (as long as the file still matches the rule).

- **If a rule matches a file or folder, that rule will not reapply its actions on a subsequent pass.**
  Hazel keeps track of which files or folders have matched which rules, and once a match has occurred and the rule's action(s) have run successfully, the rule ignores that item the next time through. This approach prevents Hazel from running in endless loops.

For example, suppose you have a rule that looks for apps in the monitored folder and opens them if found. An app appears in the folder, and the rule opens it. Then you quit the app. Hazel won't keep opening the app every time it examines that folder, even though the app still matches your rule. Likewise, if a rule modifies a file (for example, adding a tag) and you undo or change that modification manually, the rule won't reapply it. However, if you change the rule's conditions, Hazel will check them against the file and run the actions if they match; similarly, if an item stops matching a rule, and then later matches it again, Hazel does run the rule again. In addition, if the file or folder matches another rule, later in your list, that rule's actions will run.

- **You can exclude files from subsequent rules by using the "Ignore" action.** You can use the fact that Hazel runs only the first matching rule on any file to effectively exempt certain files from being acted on by other rules in your list. For instance, let's say you have a rule that monitors your Downloads folder and copies all PDFs to your Desktop. But you want to be sure that rule doesn't apply to any PDFs that have "NSFW" in their filenames. You can add a rule—*at the beginning of your rules list*—with a condition "Name matches *NSFW*" and set that rule's action to "Ignore." Whenever an item matches that condition, the action runs, but since the action is to do nothing, no further rules apply to that item.

- **You can create a catch-all rule for files and folders that don't match any other rules.**
  If you want to have a default action run on files and folders that don't match any of your other rules, give your last rule the single condition "Any file" and then add whatever actions you like. Since "Any file" matches, as it says, any file (as well as any folder), it will match anything that has gotten through all the rules preceding it.

SEE ALSO

Create & Edit Rules

Using Nested Conditions

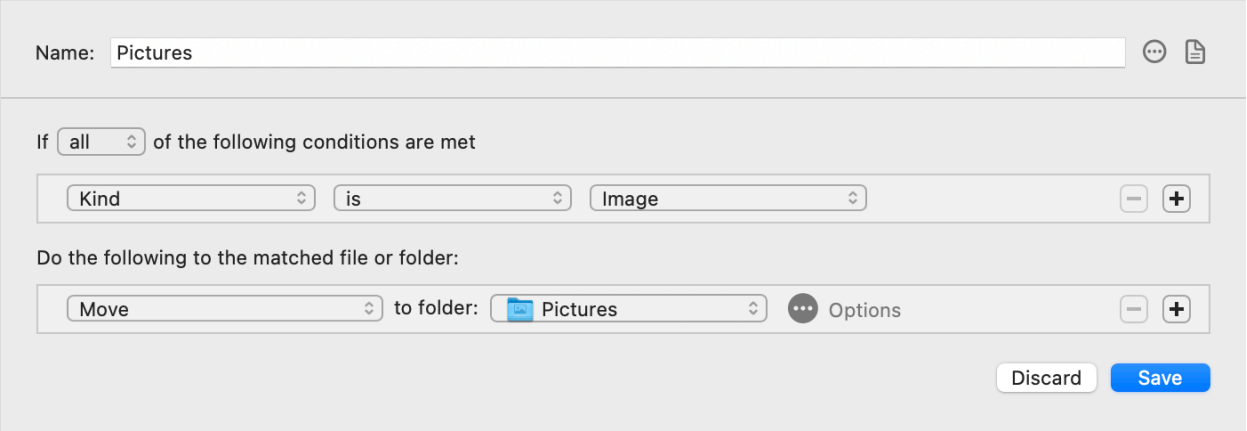
Processing Subfolders

Create a Rule

Hazel's rules, which determine which decisions it makes as it organizes your files, can be as simple or complex as you need them to be. (See About Folders & Rules for an introduction to what rules can do.) Fundamentally, all rules consist of two parts:

- **Conditions:** What Hazel should look for when examining files and folders in the monitored folder
- **Actions:** What Hazel should do when the conditions are met

Each folder can have any number of rules. Each rule, in turn, can have one or more conditions that, when met, trigger one or more actions.



Decide What Your Rule Will Do

The About Folders & Rules page gives numerous examples of what a rule can do. When deciding what rules to create for yourself, keep in mind the following tips:

- Work on one folder at a time. (Your Downloads folder is probably a good place to start.)
- Within each folder, work on one rule at a time. (You'll find Hazel's actions easier to understand if you can watch them working individually at first.)
- Unsure what rules to create? Ask yourself these questions about the items currently in that folder (or items you expect that folder to contain regularly):
    - What will you want to **move**? (And remember, the destination for items you move doesn't have to be within the monitored folder!)

- What will you want to **delete**? (Some folders accumulate files that turn out to be useless, and Hazel can delete them automatically—perhaps after a specified period of time—by moving them to the Trash.)
- What will you want to **change**? (If you regularly rename files according to a certain pattern, change extensions, add tags, or make other modifications, Hazel may be able to do those tedious tasks for you.)
- What will you want to **know about**? (Hazel can display a notification when a file matches a rule. It can even open a file for you.)

Create a Rule

Once you know what you want your rule to do, it takes just a few clicks to create it.

To create a rule:

1. Select (or add) the folder to which you want the rule to apply.
2. Click the Create Rule button on the toolbar. A panel (similar to the one pictured above) appears in which you can specify the rule's details.
3. In the "Name" field, enter a description. This can be anything you like; it's just to help you identify the rule later. Optionally, click the Note icon to enter a descriptive note (for yourself or anyone with whom you may later share this rule.)
4. Choose "any," "all," or "none" from the pop-up menu to indicate whether you want the rule to match if at least one condition (any), all conditions (all), or no conditions (none) are true for a given item in the folder being monitored. (If your rule has only one condition, this choice has no effect.)
5. Create conditions. Fill in the first condition by choosing an attribute from the leftmost pop-up menu (the one labeled Kind in the example above). Depending on which attribute you choose, Hazel may then display one or more additional pop-up menus or fields, which you can use to specify the condition you're looking for. (For details on the available attributes, see Attribute Reference. For examples of conditions you might create, see About Folders & Rules.)
6. If a single condition is insufficient to describe what you want to match, you can add more:
    - Create an additional condition by clicking the plus button at the right side of the condition.
    - Remove a condition by clicking the minus button at the right end of

the condition.

- Reorder conditions by dragging them up or down in the list.
- If you need multiple levels of conditions (to specify, for example, “any of (A and B) or (C and D)”), read about Using Nested Conditions.
- When adding conditions, be sure to check whether “any” or “all” is selected, and whether that’s still the choice you want.

7. Create actions. As with conditions, you begin by choosing an action from the leftmost pop-up menu (under “Do the following to the matched file or folder”) and then fill in the additional pop-up menus or fields as they appear. If an Options icon appears (as in the example above), click it for additional settings. Use the plus and minus buttons to create additional actions, and drag them up or down in the list to rearrange the order in which they apply.
8. When you’re done, click “Save” to save your rule. If your rule contains any errors (such as a mandatory field that’s not filled in), Hazel displays an error message and highlights whichever portion(s) of the rule need to be corrected.

Newly created rules become active immediately, and you may see their effects after just a few seconds. Once you’re satisfied that the rule is behaving as you intended, you can repeat these steps to create additional rules.

If you want to temporarily disable a rule, deselect its checkbox in the “Rules” list. To learn about previewing what a rule will do as you create it, see Preview a Rule.

SEE ALSO

Preview a Rule

Use Sample Rules

About Folders & Rules

Attributes & Actions

Using Nested Conditions

Preview a Rule

As you create rules, it can be helpful to check them against the files and folders already in the target folder so you can be sure your conditions match all and only the right items. Hazel offers two ways to do this: a live preview (which works on just one item at a time) and a “Rule Status” view (which shows which rules match each of the files in the target folder, and can be used either as a preview or after the fact; see Show Rule Status).

To preview a rule with a single file:

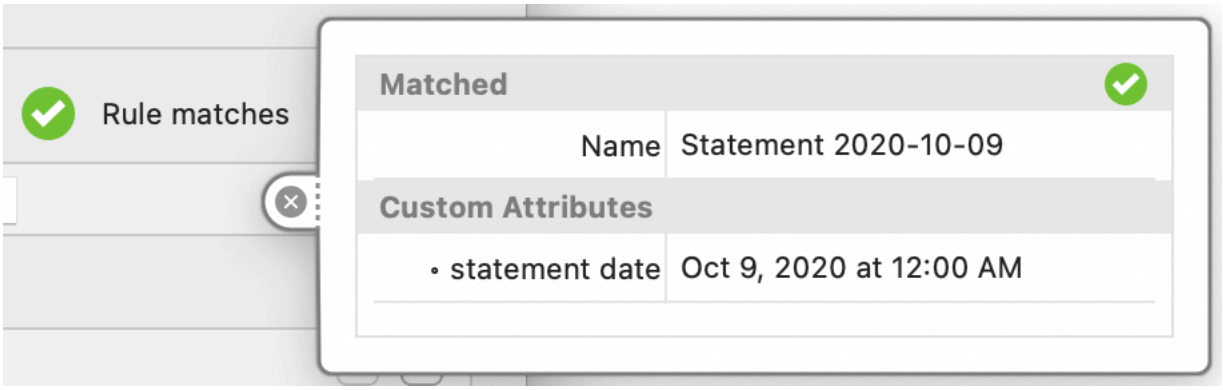
1. Choose Rule > “Preview Rule” or press ⌘-Option-P.
2. Select a file or folder within the monitored folder to preview against the rule and click Open.

If the rule as a whole (taking into account all conditions) matches, “Rule matches” appears above the conditions. If not, “Rule does not match” appears.

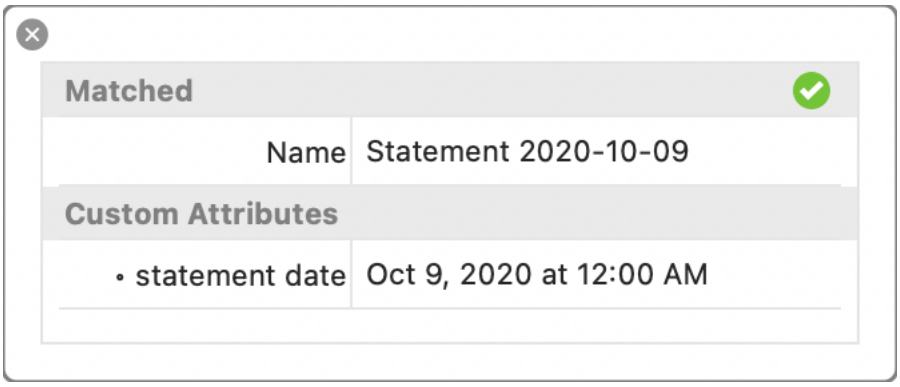
Note the following:

- Turning on Preview Rule only tells you whether the conditions match; it does not run any of your actions if they do.
- Preview Rule works even when a rule is disabled.
- With Preview Rule enabled, you can select other rules *in the same folder* to see if they match the item you selected.

In addition to showing whether the rule as a whole matches, each *condition* that matches the selected file displays a green check mark; each condition that does not match displays a red X. You can click either of these items to see a popover with the current value of the attribute in question as well as any custom attributes you’ve defined.



You can also drag this popover to detach it from the window:



When it is in this mode, the popover will stay open while you edit your rule. You can close it by clicking the close button in the top-left corner.

As you modify your conditions, the icons change in real time to reflect the current match state for the selected file or folder.

To turn off the live preview, choose Rule > “Turn Off Rule Preview” or press ⌘-Option-P.

Although Hazel’s live preview gives you quick feedback for a selected file or folder, you may instead want an overview of which rules match all the items in a folder. Hazel’s rule status interface enables you to do so—see Show Rule Status for details.

SEE ALSO

Show Rule Status

Create a Rule

Enable, Disable, or Pause Rules

Copy or Move a Rule

If you have a rule you want to apply to another folder, you can copy it (and then, if necessary, make any desired modifications to the copy). You can also move a rule to a new folder (removing it from the first one in the process).

To copy a rule to another folder:

1. Drag the rule from the “Rules” list to the desired folder in the “Folders” list.
2. Drop the rule on the folder to add the rule to the end of that folder’s rule list. Alternatively, you can drag the rule to a specific position in the rule list for the destination folder. A horizontal line appears where it will be placed.

To move a rule to another folder:

Follow the steps above, but hold down the ⌘ key during the drag and drop. The rule disappears from the original folder.

Note: Rules that are copied or moved between folders are automatically enabled in the new location. If you prefer for a moved rule to be disabled in the new location, you must deselect it manually.

SEE ALSO

Enable, Disable, or Pause Rules

Import Rules

Export Rules

Sync Rules

Edit or Remove a Rule

To edit an existing rule, select it in the “Rules” list, make any desired changes, and click Save.

You can also detach the rule editor so it appears as a floating window. This gives you a way to compare two or more rules side by side—handy if you’re creating multiple similar rules with lots of complex conditions and actions. To detach the rule editor, click the Detach Editor icon on the toolbar. The rule editor for the current rule splits off into a separate window, which you can move, close, zoom, or minimize in the usual ways. You can then select another rule to view and edit in the main window.

To remove a rule:

1. Select the rule and press the Delete key.
2. Read the warning message and click “Remove Rule” to confirm deletion.

Note: If you delete a rule and immediately change your mind, you can put it back by choosing Edit > “Undo” or pressing ⌘-Z.

SEE ALSO

Create a Rule

Manage Rules

Use Sample Rules

Hazel includes five sample rules you can add to any folder as a starting point for creating your own rules.

To add the sample rules:

1. Select a folder in Hazel's "Folders" list.
2. Choose Help > "Load Sample Rules."
3. To enable a rule, select its checkbox.

The sample rules (all disabled by default) appear in that folder's "Rules" list. You can edit any of these rules to meet your needs.

Some sample rules include descriptions, as indicated by a blue Note icon. Click the icon for more information on what the rule does.

SEE ALSO

Create a Rule

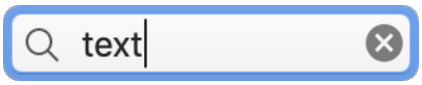
Copy or Move a Rule

About Folders & Rules

Search Rules

If you need to find one rule among many, or filter the list of rules for a certain folder to show only those matching some criteria, you can use Hazel's search feature to do so.

To search or filter, click the search icon on the toolbar and type any text in the search field that appears.



As you type, Hazel filters the list of rules to show only items containing the search term. (The search term can appear in the rule's name, in its note, or in *any* of the rule's fields —including embedded AppleScripts, JavaScripts, and shell scripts.)

SEE ALSO

Create a Rule

Preview a Rule

Manage Rules

As you accumulate rules, you may need to manage them in any of the following ways:

- Enable, Disable, or Pause Rules: Enable or disable individual rules, or pause or resume all the rules for a folder.
- Run Rules Manually: Force Hazel to run its rules for a folder immediately.
- Show Rule Status: See the items in the monitored folder along with which rules (if any) have matched each one, and when.
- Revert a File: Undo certain modifications a rule has made to a file.
- Import Rules: Add rules other people have shared with you, or restore rules you've backed up.
- Export Rules: Share rules with other people, or save backup copies for yourself.
- Sync Rules: Sync the rules for any folder to a file (stored, for example, in Dropbox or iCloud Drive), where they can be accessed by other copies of Hazel on the same Mac (in other user accounts) or in copies of Hazel running on other Macs.
- Troubleshoot Rules: If something isn't working as expected, try these tips to figure out what's going wrong.

Enable, Disable, or Pause Rules

You can enable or disable any individual rule, or pause all the rules for a given folder.

To enable or disable a rule:

- To enable or disable a single rule, click its checkbox.
- To enable or disable all rules at once, hold down the ⌘ key while clicking a checkbox. (If you ⌘-click on an enabled rule, all rules are disabled; if you ⌘-click on a disabled rule, all rules are enabled.)

To pause all enabled rules for a folder, do any of the following:

- Select the folder and choose “Pause Rules” from the Action menu.
- Select the folder and click the Pause Rules icon on the toolbar.
- Right-click (or Control-click) the folder and choose “Pause Rules” from the contextual menu.

To resume all enabled rules for a folder, do any of the following:

- Select the folder and choose “Resume Rules” from the Action menu.
- Select the folder and click the Resume Rules icon on the toolbar.
- Right-click (or Control-click) the folder and choose “Resume Rules” from the contextual menu.

SEE ALSO

Stopping and Restarting Hazel

Run Rules Manually

Run Rules Manually

Normally, Hazel runs rules automatically as needed. Occasionally, though, you may need to force rules to run. For example:

- After you edit a rule, Hazel introduces a brief delay before running it again, but you may want it to run immediately.
- You may want to run rules once on a folder for which all rules are paused (see Enable, Disable, or Pause Rules), rather than resuming the rules for that folder.
- If rules should be running but aren't, you may want to try running them manually as a troubleshooting step.
- You may want to override Hazel's default behavior of running an action only once on a matching file. (See Understand the Logic of Rules.) Running rules manually causes actions to apply to matching files even if they previously matched a given rule.

To manually run the rules for a given folder, do any of the following:

- Select the folder whose rules you want to run and choose Rule > "Run Rules Now," or choose "Run Rules Now" from the Action menu.
- Right-click (or Control-click) a folder in Hazel and choose "Run Rules Now" from the contextual menu.
- With the Hazel status menu enabled, choose "Run Rules" > *Folder Name* from that menu. For folders in groups, choose "Run Rules" > *Group Name* > *Folder Name*.

SEE ALSO

Enable, Disable, or Pause Rules

Stopping & Restarting Hazel

Understand the Logic of Rules

Show Rule Status

Hazel's rule status display shows all the items in the monitored folder along with which rules (if any) have matched each one, and when.

To display a folder's rule status:

1. Select the folder for which you want to view the rule status.
2. Click the Rule Status icon or choose Rule > "Rule Status" (⌘**-Option-R). The Rule Status window shows you all the items in the folder as well as which rules they match:**
    - For files that have already matched a rule, you can see how long ago they matched, which is when Hazel last ran and processed the file. If the "Date Matched" column is empty for a given item, Hazel has not processed that item yet.
    - If there was an error processing a file, instead of the date matched, Hazel displays the error that occurred.
3. You can view a file's attributes by selecting it and then clicking its Info icon. In the Info popover, you can also choose a rule from the popup menu at the top. If the rule matches, the attributes that match are highlighted in blue. If it does not match, the attributes that caused it to fail are highlighted in red. Click the X icon to close the popover.
4. When you're finished viewing rule matches, close the window.

Please note the following:

- You can press the Space bar to bring up a Quick Look preview of the selected item.
- With the Rule Status window open, you can select a different folder to show the status of its rules.
- If you change a rule while the Rule Status window is open, you must click Refresh to update the display.

SEE ALSO

Preview a Rule

Enable, Disable, or Pause Rules

Revert a File

When a Hazel rule makes certain changes to a file, you can undo them by right-clicking (or Control-clicking) the file and choosing Revert from the Finder's contextual menu or clicking on the corresponding menu items in Hazel's menubar icon's menu. This can be useful for restoring files to their previous state when a rule has done something unintended, but it's limited.

Revert does apply to the following actions:

- Move
- Rename
- Sort into subfolder
- Add tags
- Remove tags
- Set color label
- Add comment
- Toggle extension
- Toggle lock
- Archive
- Unarchive

However, it does not apply to any other action, such as Copy, Sync, Upload, Make alias, Import into Music/Photos/TV, or running automations.

SEE ALSO

Understand the Logic of Rules

Enable, Disable, or Pause Rules

Run Rules Manually

Import Rules

You can add rules to the current folder by importing them from a file—either one that you previously exported, or one shared by another Hazel user.

To import rules:

1. Select the folder you want to import the rules to.
2. Select the rule in the rule list after which you want the rules to be added.
3. From the Action menu, choose “Import Rules.”
4. Select the file containing the rules you want to import and click “Open.”
5. Hazel adds the newly imported rules, but initially disables them. To enable a rule, select its checkbox.

SEE ALSO

Export Rules

Enable, Disable, or Pause Rules

Sync Rules

Export Rules

You can save all the rules for a given folder to a separate file, which you can then use as a backup or share with other Hazel users.

To export all rules for a single folder:

1. Select the folder whose rules you want to export. (Hazel doesn't provide a way to export individual rules, but you can work around this limitation by copying a rule to a new folder and then exporting that folder's rules.)
2. From the Action menu, choose "Export Rules."
3. Select the folder where you want to save the rules, and optionally edit the name of the file.
4. Click "Save."

Alternatively, you can export rules for all your folders at once.

To export all rules for all folders:

1. From the Action menu, choose "Export All."
2. Select the folder where you want to save the rules.
3. Click "Save."

Hazel creates separate files at your selected destination for each folder's rules.

SEE ALSO

Import Rules

Sync Rules

Sync Rules

In addition to exporting and importing rules (each of which is a one-time procedure), you can sync all the rules for a given folder to an external file, such that whenever your rules change in Hazel, that file changes—and whenever that file changes, Hazel updates its rules. Syncing your rules to an external file enables you to share them with other user accounts on your Mac (as long as the file is stored in a folder all users can access) or with other Macs (if the file is stored in a location all the Macs can access, such as Dropbox, iCloud Drive, or a network server). You can even use this mechanism to sync the same set of rules between two different folders in a single copy of Hazel.

When syncing rules, the following considerations apply:

- Hazel can sync only the complete set of rules for any given folder, not individual rules.
- Hazel does not sync rules' enabled/disabled state. Thus, you could have the same set of rules on two Macs but have a different combination of them enabled on each machine.
- Newly added rules are initially disabled in all other instances of Hazel (other user accounts or other Macs) using the synced file—just as if they were moved or copied. You must manually enable these rules to use them in the other locations.
- Syncing cannot *merge* two sets of rules. So, if you sync rules from Mac A to File A, and then tell Mac B to use File A for its rules, Hazel will *overwrite* the rules Mac B was already using for that folder. Therefore, if you expect to sync the rules for any folder between Macs or users, it's best to do so before you set up any rules in the folder in question in the second instance (user account or Mac) of Hazel.
- Hazel also keeps a local cache of your rules, enabling you to continue running Hazel even if the synced rules file is temporarily inaccessible.

Configuring rule sync is a two-step process (for each folder you want to sync). First you set up a sync file. Then, for each additional instance of Hazel you want to sync, you set up a folder to use that file for its rules.

To set up a sync file:

1. Select the folder whose rules you want to sync.
2. From the Action ⋯ menu, choose "Rule Sync Settings."
3. In the dialog that appears, click "Set up new sync file."
4. Select the location (such as your Dropbox folder or iCloud Drive) where you

want to save the synced rules, and edit the name if you like. Click “Save.”

5. Click “Done.”

A sync icon appears next to the folder name to indicate that its rules are being synced.

To sync rules with another folder:

1. In another instance of Hazel (another user account on the same Mac, or a different Mac), make sure you have access to the location where the sync file you created above was stored.
2. From the Action menu, choose “Rule Sync Settings.”
3. In the dialog that appears, click “Use existing sync file.” Read the warning that the current rules for that folder will be overwritten, and if you’re sure that’s what you want to do, click Continue.
4. Select the sync file you set up earlier and click “Open.”
5. Click “Done.”

A sync icon appears next to the folder name to indicate that its rules are being synced, and the rules from the sync file appear in the “Rules” list for that folder.

Any changes to the synced rules that you make in one instance of Hazel will now apply to the other as well.

Note: If an error occurs in syncing, reading, or writing the rules, an error icon appears. Hover over this icon with your pointer to see details about the error.

To stop syncing rules with another folder:

1. Select the folder whose rules you want to stop syncing.
2. From the Action menu, choose “Rule Sync Settings.”
3. In the dialog that appears, click the X icon next to the rule file.
4. Click “Done.”

The rules for this folder are now detached and will no longer sync.

SEE ALSO

Import Rules

Export Rules

Enable, Disable, or Pause Rules

Troubleshoot Rules

Rules don’t work? Here are a few ways to get to bottom of things.

Rule Preview

While editing your rule, you can preview it. See Preview a Rule for details.

View the Rule Status Window

You can view the files in your folder and what rules they match. See Show Rule Status.

Check the Logs

If everything checks out in the preview/status, the next step is to check the logs (see Viewing Logs). You can do that in either of two ways:

- Choose Help > “View Logs.”
- If you enabled the Hazel status menu, choose “View Log” from that menu.

You can use the search field to find log messages specific to the file you are having problems with. You can also enable debug mode which will result in much more descriptive logging. Instructions on how to do that here: Debug Mode.

If you are still unable to track down the problem, contact support by choosing Help > “Contact Support.” Be sure to attach the information you gained from using the tools above.

SEE ALSO

Enable, Disable, or Pause Rules

Run Rules Manually

Learn More

Attributes & Actions

This group of topics provides extensive coverage of all the attributes used in rule conditions and all the actions Hazel can apply when conditions match:

- **Attributes**
    - Attribute Reference: A list of all the attributes Hazel supports for rule conditions, with explanations of each one. Subtopics:
    - Using Match Patterns in Conditions: Some conditions let you create patterns of text, numbers, dates, and so on. This page describes how to edit them.
    - Using Custom Attributes: If Hazel's built-in attributes don't meet your needs, you can create your own custom text, date, list, or table attributes.
- **Actions**
    - Action Reference: A list of all the actions Hazel supports, with explanations of each one. Subtopics:
    - Copying Folder Structure: Some actions can copy or move an entire folder; this page describes your options for maintaining the folder structure at the destination.
    - Specifying Upload Options: This page describes options for uploading files via FTP, SFTP, or WebDAV.
    - Using Patterns in Actions: Just as a condition can include a user-defined pattern, so can an action (such as "Rename" or "Add comment"). Subtopics:
        - Using the Counter Attribute: Rule actions can number files or folders sequentially with Hazel's flexible counter attribute.
        - Editing Date Patterns: Certain actions can use date attributes, and Hazel lets you format these dates in any way you like.
- Using AppleScript or JavaScript: You can use AppleScript or JavaScript to create conditions or actions used in Hazel rules. This page provides an overview.
- Using Automator: Hazel actions can trigger Automator workflows, as described on this page.
- Using Shell Scripts: Shell scripts can provide both conditions and actions for

Hazel rules.

Attribute Reference

Each rule starts with one or more conditions you specify. Each condition, in turn, starts with an attribute—a quality of a file or folder in the monitored folder that Hazel can examine. Conditions can use a wide selection of attributes. (And, if none of these attributes provide you with the information you need, you can create your own custom attributes using AppleScript, JavaScript, or match patterns. See Using Custom Attributes for details.)

Hazel recognizes the following attributes:

Name

The name of the file (or folder), without any extension. So, if you have a file called `example.jpg`, this attribute matches against `example`.

Extension

The file's extension, which is the part of the file name after the last dot (.). The file `example.jpg` has the extension `jpg`. (By default, extensions are hidden in the Finder, but Hazel matches them regardless of whether they're visible. To toggle the visibility of extensions in the Finder, go to Finder > "Settings" > "Advanced" and select or deselect "Show all filename extensions.")

Full Name

The full name of the file (or folder)—that is, the name plus the extension.

Date Added

The date (including time) when the file was added to the current folder. This attribute is useful for determining how long a file has been in the monitored folder. (See Note below.)

Date Created

The date (including time) the file was created. (See Note below.)

Date Last Modified

The date (including time) the file was last changed. (See Note below.)

Date Last Opened

The date (including time) the file was last opened. (See Note below.)

Date Last Matched

The date (including time) when the file or folder last matched a rule and the rule's actions ran successfully. (See Note below.)

Current Time

The current time at the moment the rule runs. (See Note below.)

Note: When using any of the above "Date" attributes with "occurs before" or "occurs after," or the "Current Time" attribute, keep in mind that Hazel treats each day as starting or ending at midnight. So, if you set up a condition like "current time is after 10 PM on weekdays," the condition matches only from 10 PM until midnight each weekday—Hazel does not consider 12:01 AM the following morning to be "after" 10 PM.

Kind

The type of file, such as a document, movie, or folder. With second pop-up menu set to "is" or "is not," choose "Other" from the third pop-up menu to display a list of all available file types on your Mac.

Tags

Any tag that can be assigned in the Finder. When the second pop-up menu is set to "contains tags" or "does not contain tags," clicking in the field that appears displays a list of all currently defined Finder tags (click Show All to display all tags if the list is long). If the second pop-up menu is set to "contain" or "do not contain," you can enter any freeform text in the field.

If the second pop-up menu is set to "match" or "do not match," you can enter a match pattern in the field, and optionally grab text so that it can be used in your actions.

Color Label

The color label, if any, applied to the file. This will match any of the tags which are currently associated with that color.

Comment

Freeform notes associated with a file or folder. You can see and edit these comments in the Finder by selecting the file or folder and choosing File > "Get Info."

Size

The size of the file. Because of the way macOS calculates sizes, this figure (the actual size) may be less than the amount of space the file occupies on disk.

Locked

Whether the file is locked or not.

Contents

The text content of the file. If the second pop-up menu is set to “contain” or “do not contain,” you can match specific words/text..

If the second pop-up menu is set to “contain match” or “do not contain match,” you can enter a match pattern in the field, and optionally grab text so that it can be used in your actions.

There are several options for the Contents attribute:

- **Text recognition:** Hazel can perform on-the-fly text recognition on images (and on PDFs that do not already contain a text layer) for the purpose of matching content. By default, Hazel will use text recognition if the file does not already have searchable text. Click the rule options icon next to the rule’s name to show other options, such as using text recognition even if the file already has searchable text or turning it off. Note that this feature does not save the recognized text to the file, as a conventional OCR app does.
- **PDF passwords:** If you use Hazel to process password-protected PDFs, you can supply the password when creating a rule so that Hazel can look inside the files. To do this, click the rule options icon next to the rule’s name and enter the password in the Document Password field. The password is stored securely in your system keychain. Note that each Hazel rule can contain only one document password; if more than one password might be used with the PDFs you process, you must create a separate rule for each password.
- **Is blank/is not blank:** When Contents is the attribute in a condition, you can now choose “is blank” or “is not blank” from the second pop-up menu. A file is considered blank if it contains no text. This test is potentially useful in conjunction with text recognition as described just above, as it can enable your rule to take different actions depending on whether an image or PDF was found to contain any text. Alternatively, if you disable text recognition, you can use this condition to determine whether a file intrinsically has text—for example, whether a PDF has already been processed by OCR.

Source URL/Address

The URL or email address the file came from. Note that only certain apps, such as Safari and Mail, store this information.

Subfolder Depth

The number of subfolders in the folder, starting from the folder being monitored. This is useful only if you have a rule set up to descend into subfolders using the “Run rules on folder contents” action (see Processing Subfolders). A subfolder depth of 0 indicates the folder being monitored.

Sub-file/folder Count

How many files and folders are contained by the item being matched by the rule (preferably a folder). This counts the number of items at the top level of the monitored folder, not those contained in subfolders. If the current item being processed is a file, this number is 0. (This is similar to Spotlight’s Number of Items attribute, except that this does not include hidden files in the count.)

Any File

This attribute matches all files (and folders). This is useful when the final rule for a given folder is a “catch-all” rule that applies to any files or folders that don’t match any of the previous rules—but it makes sense only when it is the sole condition in a rule.

Passes AppleScript

Allows you to provide an AppleScript to evaluate the file. For details on writing AppleScript for Hazel, see Using AppleScript or JavaScript.

Passes JavaScript

Allows you to provide a JavaScript to evaluate the file. For details on writing JavaScript for Hazel, see Using AppleScript or JavaScript.

Passes shell script

Allows you to provide a shell script to evaluate the file. For details on writing shell scripts for Hazel, see Using Shell Scripts.

Other

This attribute allows you to select any Spotlight attribute, of which macOS may have dozens or hundreds (depending on which apps you have installed).

SEE ALSO

Using Match Patterns in Conditions

Using Custom Attributes

Using the Counter Attribute

Action Reference

Using Match Patterns in Conditions

Sometimes you may need a rule to check for a condition that matches a *pattern* as opposed to one that simply matches a fixed attribute (for example, “match any file with a three-digit number in its name,” as opposed to “match any file with the number 372 in its name”). Hazel offers an extensive pattern-building interface for just such situations. (You can also use patterns in actions, but the process is a bit different; see Using Patterns in Actions.)

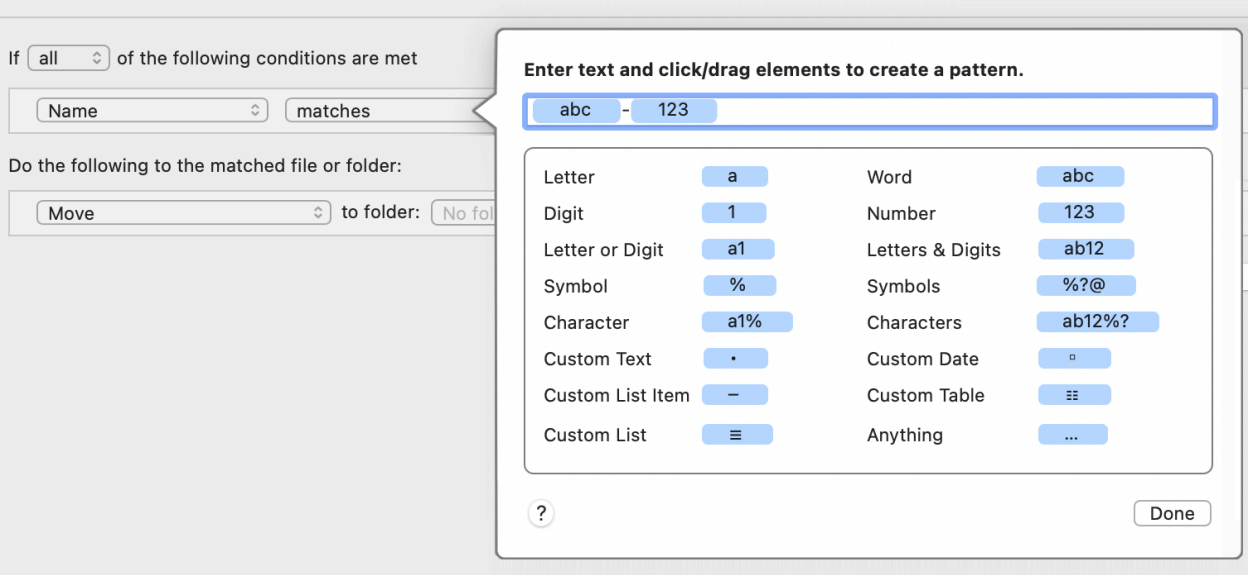
Hazel provides 11 built-in variables that you can use in constructing your match pattern, but if those are insufficient for your needs, you can create your own custom text, date, list, and table attributes to use in match patterns (see Use Custom Attributes). Custom attributes, in turn, are available to be used in later conditions and in the rule’s actions, enabling you to do things like rename files with only part of the original filename, or sort files into any of several subfolders based on their extension. (Again, refer to Using Patterns in Actions for details.)

For the following attributes, you can create patterns by choosing “matches” or “does not match” (or “match”/“do not match,” or “contain match”/“do not contain match,” depending on the context) from the second pop-up menu of the condition:

- Name
- Extension
- Full Name
- Tags
- Comment
- Contents
- Source URL/Address
- Other

For example, you might create a pattern that means “Name matches any word, followed by a hyphen, followed by any number.”

When you click in the field to edit a “match” or “does not match” pattern, a popover like this one appears:



Fill in the field at the top of this popover with the pattern you want. You can use a combination of plain text you type yourself (to include static elements as part of your pattern) and *tokens*—blue bubbles representing any of numerous variables. To add a token to your pattern from the list below, either click it (once) or drag it to the desired position in the pattern.

For instance, in the image above, the pattern specifies a word (any sequence of letters) followed by a hyphen, followed by a number (one or more digits). Using that pattern, the name `blah-123` matches, but `123-blah` does not (since it doesn't start with letters), nor does `blah123` (no hyphen after the word) or `blah-blah` (no number at the end).

Here's what each token matches:

- **"Letter (a)," "Digit (1)," "Letter or Digit (a1)," "Symbol (%)," or "Character (a1%)":** These tokens each match a single character of the type(s) specified. The "Symbol" token matches any punctuation character while "Character" matches any non-blank character.
- **"Word (abc)," "Number (123)," "Letters & Digits (ab12)," "Symbols (%?@)," or "Characters (ab12%?)":** These tokens each match a sequence of characters of the type(s) specified. That is, "Word" matches any sequence of letters, Number matches any sequence of digits, and so on.
- **Note:** The Number token in match patterns matches numbers that include thousands separators, decimal points, or both; and decimal values are passed on to the output.
- **"Anything (…)":** This token matches ***any single character or group of***

characters or spaces, including nothing. (It's thus much like the * wildcard in most command-line environments.) If you have a pattern **(123)(…)**, then the text `456` matches. The **"Number" (123)** token matches the `456`, and the **"Anything" (…)** token matches nothing.

The five custom attributes are even more powerful, in that they let you create named patterns of your own that are then available to other conditions and actions in this rule:

- **"Custom Text (•)":** This token lets you create your own custom text-based attribute. See Custom Text Attributes.
- **"Custom Date (▫)":** This token lets you create your own custom date attribute (that is, a particular date format). See Custom Date Attributes.
- **"Custom List (≡)":** This token can create a list based on a rule condition and reuse it in an action. See Custom List Attributes.
- **"Custom List Item (—)":** This token lets you create a list, such that your condition can match any of the items on the list. See Custom List Item Attributes.
- **"Custom Table (☷)":** This token lets you create a table, such that your condition can match any of the items in a particular column, after which you can apply actions based on the contents of another column in the same row. See Custom Table Attributes.

When you're finished building your pattern, click Done to close the popover and continue creating or editing your rule.

Match Pattern Rules and Tips

Although most patterns are self-explanatory, you should be aware of a few special considerations:

- Patterns are not case-sensitive (even with text you type manually). ***A*** is the same as ***a***.
- When a condition uses the "match"/"matches" or "does not match"/"do not match" instruction in the second pop-up menu, the pattern must match the *full text* of the attribute. If you want to be sure your condition also matches cases with additional text before and/or after your pattern, use the "Anything" (…) pattern token before and/or after your pattern. Using the example above, the text `45blah-123` does not match, because even though the last part after the `45` does match, the pattern has to start from the beginning and encompass

everything until the end. Likewise, `blah-123ab` does not match because of the `ab` at the end. To match all these variants, the pattern could be changed to **(…)(abc)-(123)(…)**

- In contrast to the preceding rule, when a condition uses the "contain match" instruction in the second pop-up menu (available with the "Contents" attribute and some of the "Other" attributes), the pattern only needs to match *part* of the text. (So, if the condition is "Contents contain match **(abc)-(123)**," that would match a file that contains the text `A#3foo-75zxy`, because the string `foo-75`, found within that text, matches the pattern. Also, a space in a "contain match" pattern matches any number of spaces (or other invisible characters).
- With certain attributes ("Contents" and some of the "Other" attributes), an additional option appears at the bottom of the popover: "Match the ___ occurrence from the beginning/end." This enables you to specify which occurrence you want to match, if there could be several (for example, "Match the 3rd occurrence from the beginning," or "Match the 2nd occurrence from the end"). One reason to do this is to match only files that have a certain number of occurrences of a certain pattern—if you say "Match the 3rd occurrence from the beginning" but the file has only two occurrences, it won't match. Another reason to do this is to create a custom text token (see Custom Text Attributes) that matches a particular occurrence of a string so that you can refer to it in a later condition or action in your rule. (For example, match the 3rd instance of a custom token representing a phone number, and display that phone number in a notification when the rule runs.)

- When creating match patterns, it can be especially helpful to use Hazel's live preview feature (see Preview a Rule) to test your pattern as you go.

SEE ALSO

Editing Date Patterns

Using Custom Attributes

Using Patterns in Actions

Using Custom Attributes

In Hazel, you can create your own custom text and date attributes while building patterns. These behave much like the built-in attributes, such as “Name” or “Date Added,” except that you can give them whatever name you want and fill them in with values from different places. You can also create list item attributes (to match any item from a list) and table attributes (to match items from one column and use information from another column in a subsequent condition or action).

To learn about the various types of custom attributes and how to use them, see:

- Using Match Patterns in Conditions
- Custom Text Attributes
- Custom Date Attributes
- Custom List Attributes
- Custom List Item Attributes
- Custom Table Attributes
- Reusing Custom Attributes in Condition Patterns
- Using Custom Attributes in Actions

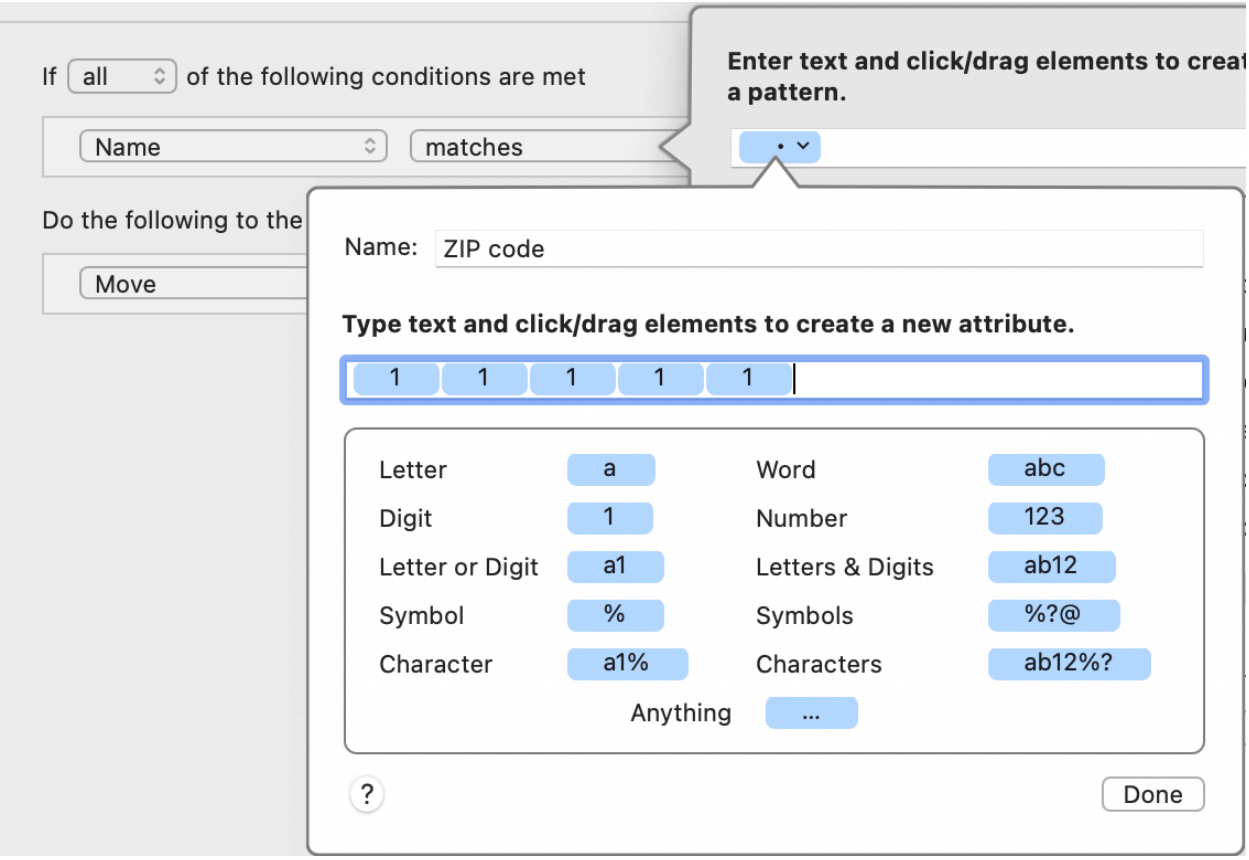
Custom Text Attributes

You can use add “Custom Text •” token to a pattern to create your own custom text attribute, which then becomes a token available for any patterns in any subsequent conditions or actions in this rule. In addition, it’s available as an attribute (that is, an entry in the leftmost pop-up menu) in any conditions (in this rule only) after the one in which you define it.

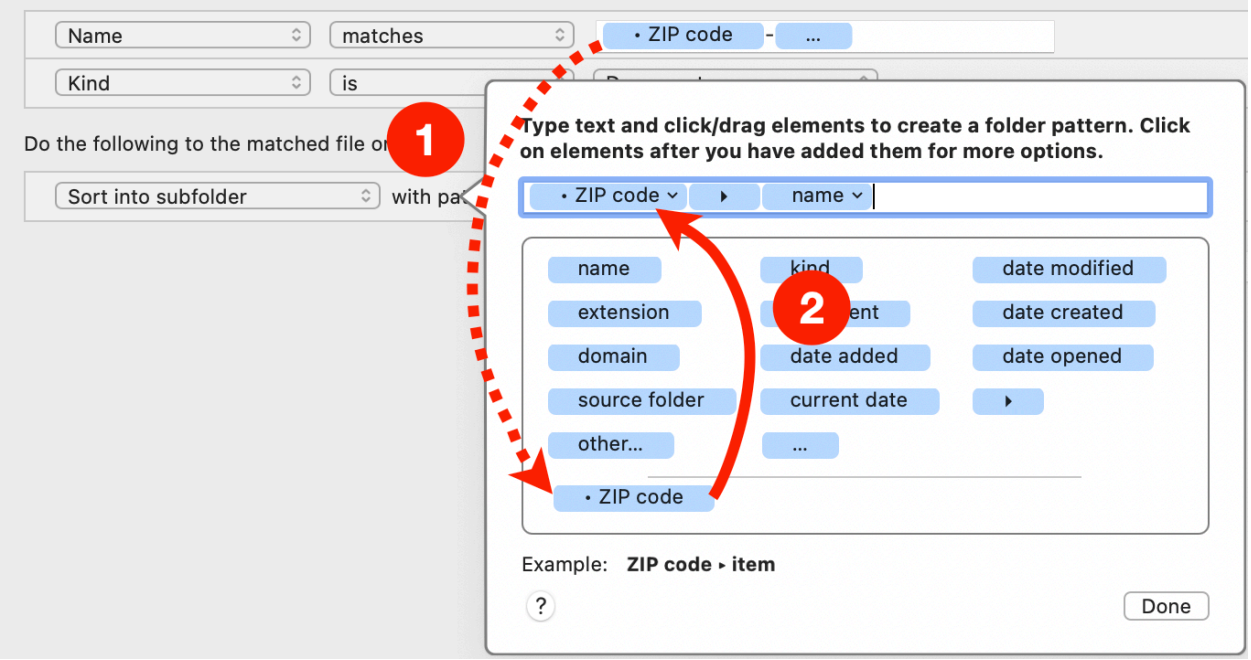
Note: You can also create custom text attributes by exporting attributes from an AppleScript or JavaScript.

Custom text tokens are useful in cases where you want to save some matched text and insert it somewhere else. For example, let’s say you have a bunch of files whose name start with a 5-digit ZIP code. You can define a token that matches that portion of the name and call your custom token “ZIP code.” This new token can then be dropped into a “Sort into Subfolders” pattern to sort the files into folders based on ZIP code.

To create a custom text token, click the “Custom Text •” token or drag it into the field at the top of the popover. When you do so, a secondary popover appears, in which you can edit your custom token:



As in the main match pattern editor, you can specify a pattern here. You must also enter a name to identify your custom token; that name then appears in the first pop-up menu for later conditions in this rule, as well as in the list of tokens you can use to create patterns for the rule's conditions or actions. Here, the token is named "ZIP code" and it matches five consecutive digits. (To help you distinguish custom text attributes, Hazel prefixes their names with • when they appear in tokens or menus.) Click "Done" when you're finished creating your custom text token.



When you're done editing, your custom text token appears, with the name you specified (prefixed with •) in the other pattern popovers for the current rule (1), which you can then drag into your patterns (2). (It also appears as an attribute at the bottom of the leftmost pop-up menu for later conditions in this rule.) When Hazel runs the rule, it keeps track of what text matched it, and when your custom token is used in the pattern for an action (see Using Patterns in Actions), Hazel inserts the text that was matched.

Using the "ZIP code" example above, if a file has the name `10101-Survey`, Hazel assigns the "• ZIP code" attribute the value `10101` (the "• ZIP code" token matches five digits in a row). When sorting into subfolders, the file is moved to a folder called `10101`.

SEE ALSO

Using Match Patterns in Conditions

Custom Date Attributes

Custom List Attributes

Custom List Item Attributes

Custom Table Attributes

Reusing Custom Attributes in Condition Patterns

Using Patterns in Actions

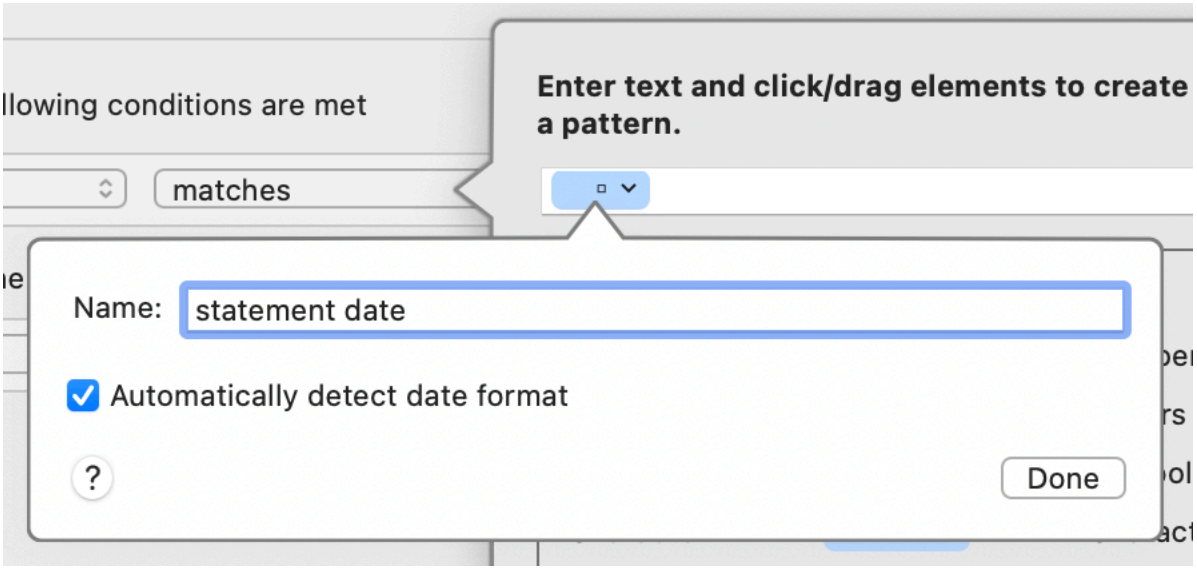
Custom Date Attributes

in addition to custom text tokens, you can create custom date tokens, which let you match dates in various formats. When a custom date attribute is then used in a pattern as part of an action, Hazel treats it as a date rather than text, which means you can transform it into a different date format if needed.

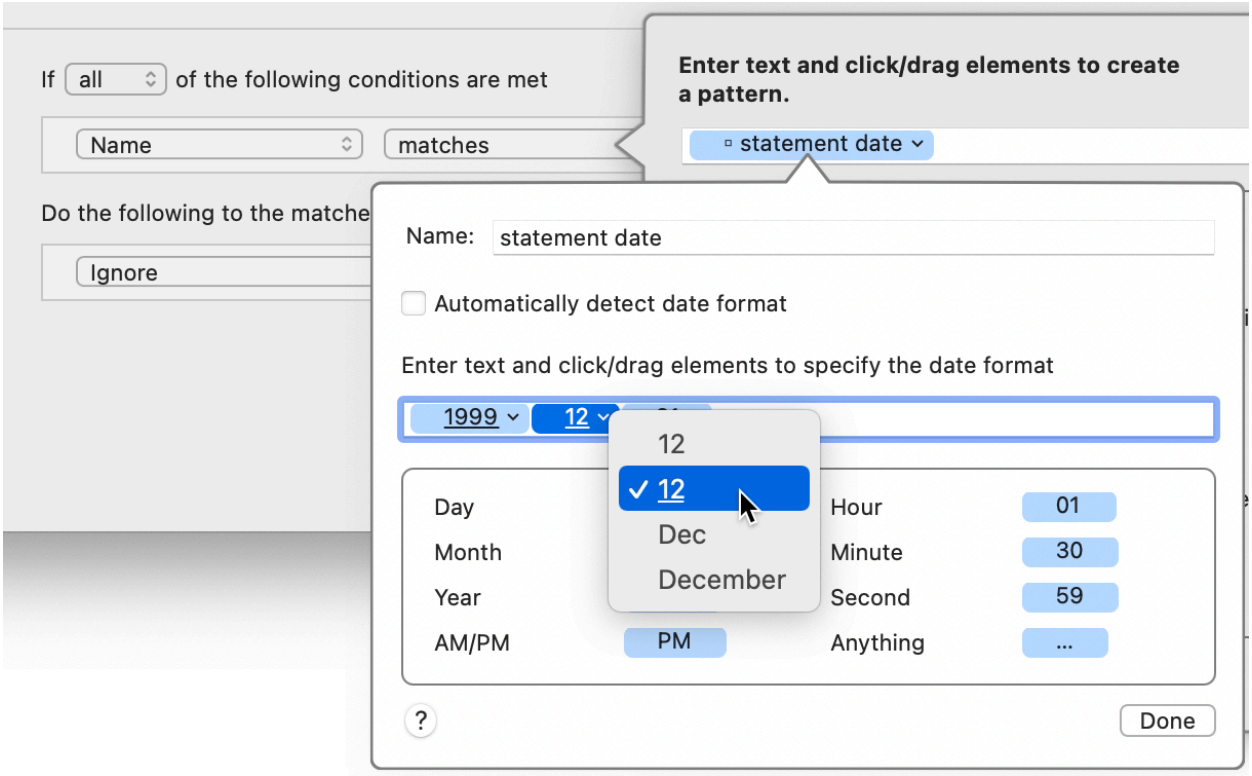
Note: You can also create custom date attributes by exporting attributes from an AppleScript or JavaScript.

For example, we can match any date in the form “2016-09-21” by using a date token. You can do this by dragging the “Custom Date ▫” token into the pattern.

By default, Hazel will try to detect the date format for you. This is indicated by the “Automatically detect date format” checkbox. When checked, Hazel will try to determine dates in various formats as defined by your system.



If automatic detection fails to detect a date or if you need more control, you can uncheck the option and specify a pattern yourself. From there, customize the pattern by clicking/dragging in the date-related tokens below, combining them with any desired text (such as hyphens or slashes).



Once you've added a date token, you can click the tiny arrow on its right side to display a pop-up menu with options that let you customize it. For example, the pop-up menu for "Year" ("99") tokens lets you specify two-digit ("99") or four-digit ("1999") years; "Month" tokens can be formatted as "12" (no leading zeroes for single-digit numbers), "12" (single-digit numbers padded with a leading zero), "Dec" (abbreviated names), or "December" (full names); and "Day" tokens can be formatted as "31" (no leading zeroes for single-digit numbers) or "31" (single-digit numbers padded with a leading zero). Like custom text attributes, custom date attributes must have a name. (To help you distinguish custom date attributes, Hazel prefixes their names with ○ when they appear in tokens or menus.) Click "Done" when you're finished creating your custom date token.

You can reuse custom date attributes in other conditions in the same rule. For details, see "Reusing Custom Attributes in Condition Patterns," just below.

In addition, you can reuse custom date attributes in actions within the same rule. When a custom date token is used as part of a pattern in an action, it acts like any other date attribute, in that you can reformat the date or adjust it as needed. One case where this is useful is when the original month is given as a number, such as "01." When you reformat it, you can have it output in textual form, such as "Jan" or "January." (And, of course, you can do the reverse, too). To learn how to formatting dates, see Editing Date Patterns.

SEE ALSO

Using Match Patterns in Conditions

Custom Text Attributes

Custom List Attributes

Custom List Item Attributes

Custom Table Attributes

Reusing Custom Attributes in Condition Patterns

Using Patterns in Actions

Using Custom Attributes in Actions

Custom List Attributes

This token can create a list based on a rule condition and then reuse that list in an action. (If you want to match a specific item in a list, you should instead use Custom List Item Attributes.)

For example, suppose you want to create a list of tags from an enclosing folder, and then apply those matching tags to each item within the folder. You can create a custom list in your condition that captures the tags, then refer to that list in the “Add tags” action.

To create a custom list attribute:

1. Create a condition with an attribute (such as “Tag” or one of numerous “Other” options) that may be used more than once per item, and that uses a match-related verb in the second pop-up menu (“matches”/“does not match,” “match”/“do not match,” or “contain match”/“do not contain match”).
2. **Note:** Custom lists can also work with lists in text (such as `apple,orange,banana`) if your condition uses an attribute such as Name, Comment, or Contents. This process, which is slightly different, is described later on this page.
3. Drag the “Custom List (≡)” token to the pattern field.
4. Type a name for your custom list attribute.
5. Type, and/or drag one or more tokens, into the field below the “Name” field to create the pattern to match against. Hazel will apply the pattern to each item in the list of attributes it examines, and create a new list comprising all those that match. (To get a copy of the entire list, drag the “Anything (…)” token to the pattern field.)
6. Click Done, then click Done again.

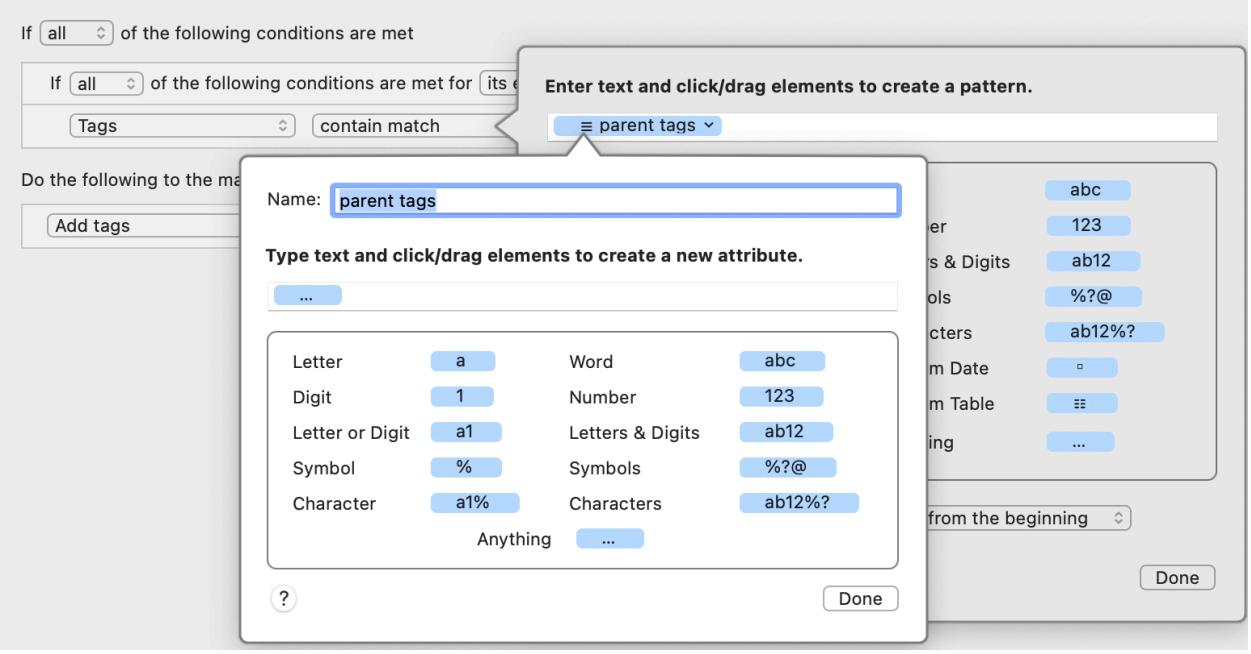
To use a custom list attribute in an action:

1. Choose an appropriate action. The “Add tags” action is the only one in Hazel 6 that can apply multiple list items individually. However, you can convert the list to regular text by choosing an action such as “Rename,” “Sort into subfolder,” “Add comment,” or “Display notification”; then, click the token for the custom list, choose Edit List Separator from the pop-up menu, and type the character(s) you want to use to separate list items. (For example, you could rename a file using the names of the tags in its parent folder, separated by bullet characters: `Tag1•Tag2•Tag3`, or put those tag names in a comment.)

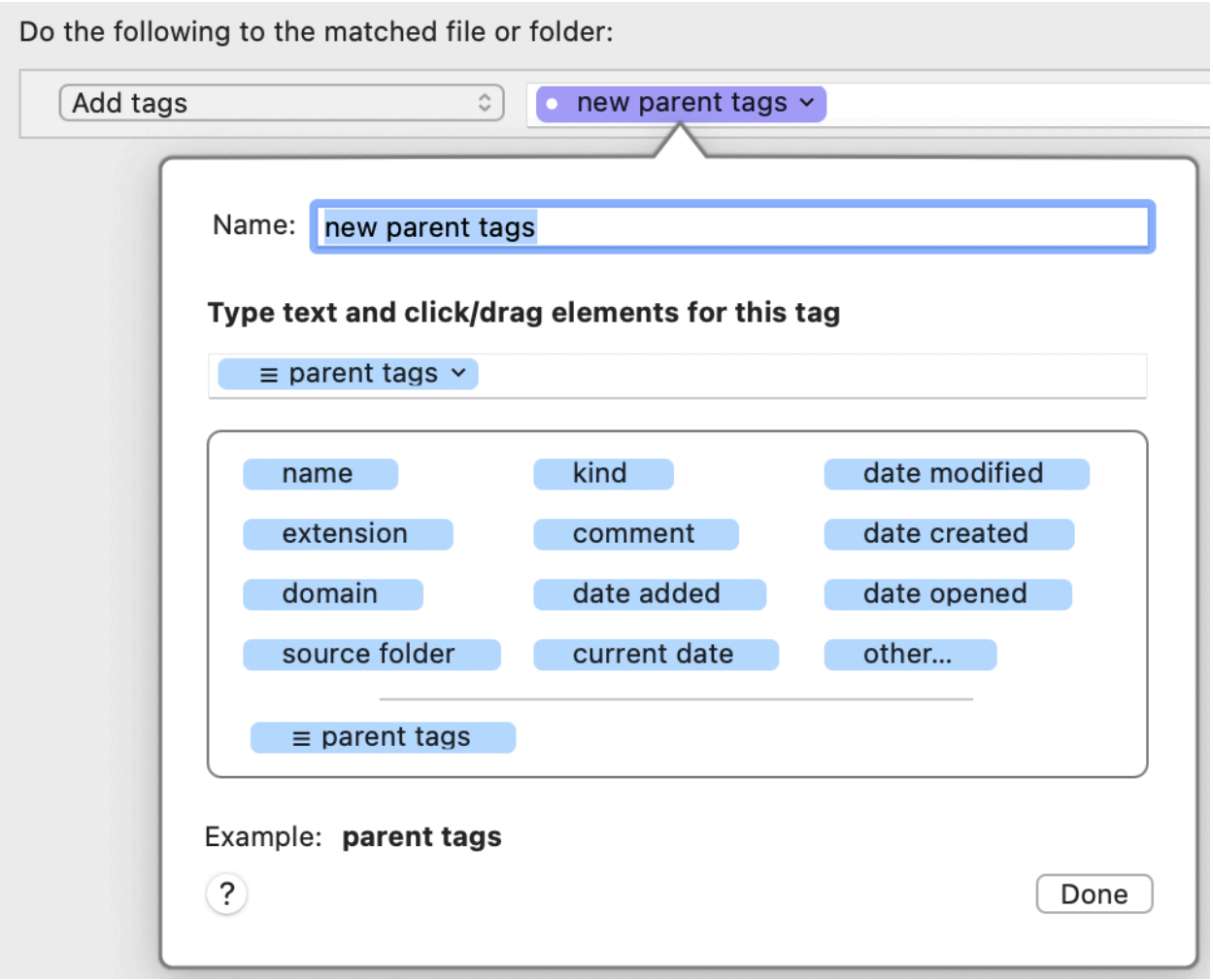
2. Drag the custom list you created in the condition into the pattern field, along with any other desired text or pattern. (In the case of the “Add tags” action, first choose “Dynamic Tag…” then drag in the custom list. See
for more information on Dynamic Tags.)

To illustrate how this works, let’s use the example mentioned above: using matching tags from an enclosing folder to apply tags to items within it.

First, we specify “If **all** of the following conditions are met for **its enclosing folder**” and add the condition “Tags contain match.” We drag the “Custom List (≡)” token to the pattern field, name it “parent tags,” and drag the “Anything (…)” element into the pattern, as shown below. This means that, no matter how many tags the enclosing folder has, Hazel will create a list, called “parent tags,” containing all and only those tags.

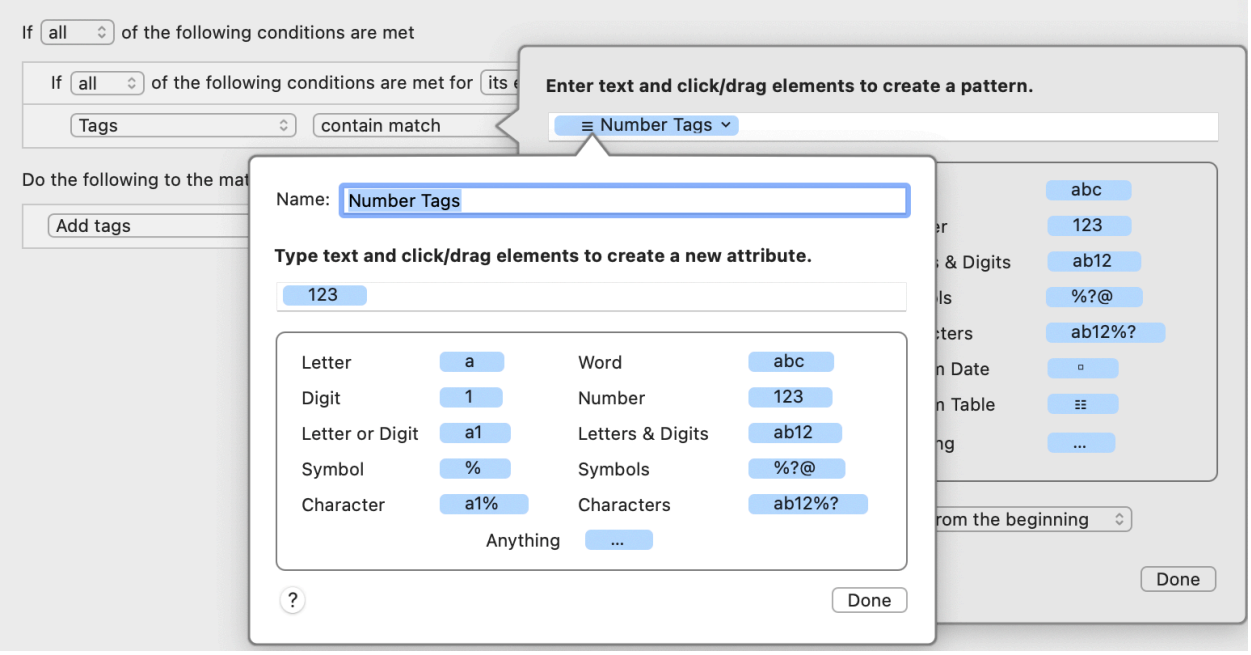


Next, we add the “Add tags” action to make use of our new list of tags containing only numbers. We first click in the “Add tags” field and choose “Dynamic Tag…” from the pop-up menu. We name the dynamic tag “new parent tags.” Then, we drag the “ parent tags” token we created in the condition to the pattern field.



Now, suppose Hazel is monitoring a folder that has one or more tags applied to it. The condition matches all those tags and makes a list, “parent tags,” with all those tags on it. The action applies that list of tags to items newly added to the monitored folder.

You can also filter the list such that the new list is a subset of the original one. For example, if you wanted to apply to the enclosed items only those tags from the parent folder consisting of digits, when creating the list attribute in the condition, instead of using the “Anything (…)” token, you could drag in the “Number (123)” token, as shown below, to make it more specific.



As mentioned above, custom lists can also work with lists in text (such as `apple,orange,banana`), such as may be found in a file's contents, comment, or name—regardless of what character separates the list items. When matching text, a condition must always capture the entire list; it can't filter the list as in the example just above. However, if the source attribute contains multiple matching lists, you can specify which one of those lists to capture.

To create a custom list attribute from text:

1. Create a condition with an attribute (such as "Name," "Comment," or "Contents") that consists of arbitrary text, and that uses a match-related verb in the second pop-up menu ("matches"/"does not match," "match"/"do not match," or "contain match"/"do not contain match").
2. Drag the "Custom List (≡)" token to the pattern field.
3. Type a name for your custom list attribute.
4. Type, and/or drag one or more tokens, into the field below the "Name" field to create the pattern to match against. Hazel will search the attributes it examines for a list whose individual elements match your pattern, and assign the name you entered to that list. (To create a custom list in which each element can be any combination of characters, drag the "Anything (…)" token to the pattern field.)
5. Using the List Separator field at the bottom of the popover that appears, enter one or more characters (such as a comma) or choose Newline or Tab from the pop-up menu to tell Hazel how to identify individual items in the list. For

example, if a text file contains a list like `dog, cat, bird` that field should contain a comma followed by a space, since that's what separates the list items.

6. Click Done.
7. Optionally, to choose a particular list from a source that contains more than one, change the field at the bottom: "Match the __ occurrence (from the beginning/ from the end)."
8. Click Done again.

Your custom list now contains every item in the original text list, and you can use that custom list in an action.

To use a custom list attribute from text in an action:

1. Choose an appropriate action. The "Add tags" action is the only one in Hazel 6 that can apply multiple list items individually. However, you can use the list as regular text by choosing an action such as "Rename," "Add comment," or "Display notification"; then, click the token for the custom list, choose Edit List Separator from the pop-up menu, and type the character(s) you want to use to separate list items.
2. Drag the custom list you created in the condition into the pattern field, along with any other desired text or pattern. (In the case of the "Add tags" action, first choose "Dynamic Tag…" then drag in the custom list. See
   for more information on Dynamic Tags.)

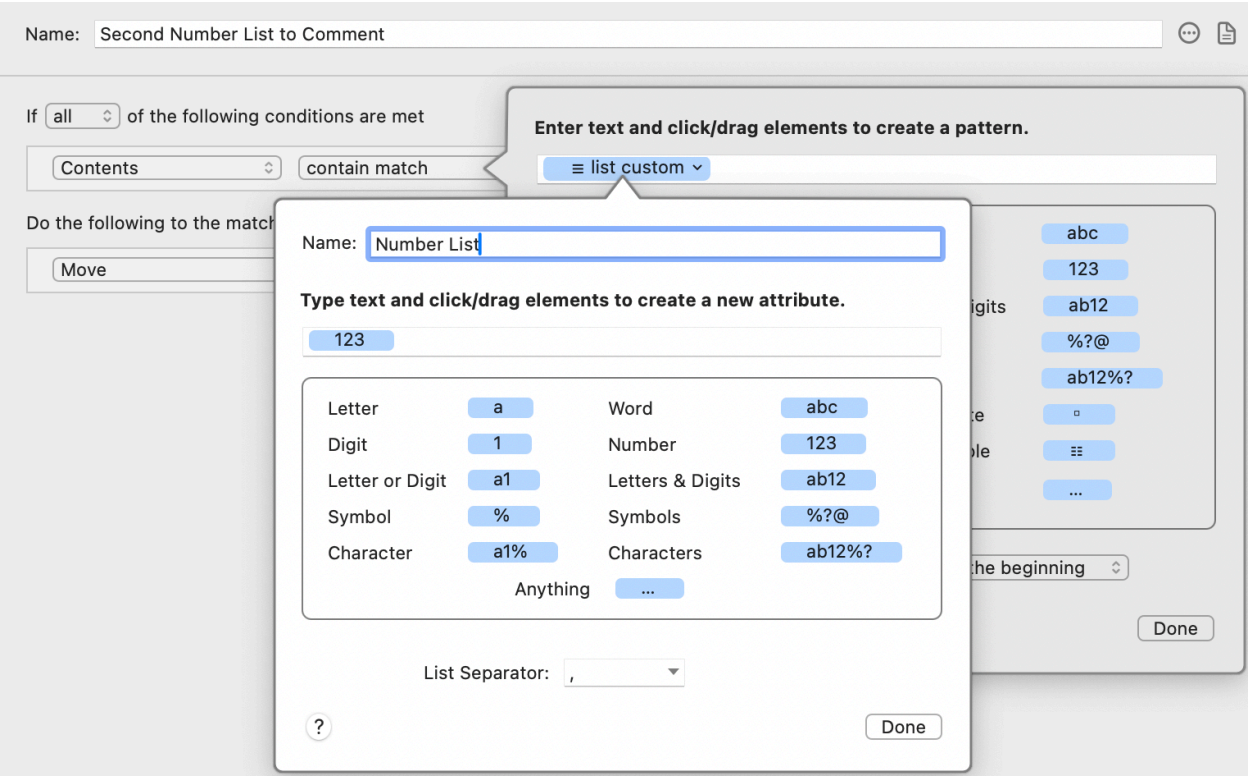
Let's put this "custom list attribute from text" idea all together by creating a rule that looks in the contents of text files for lists containing sequences of digits, separated by commas, such as:

```
10,20,30,40,50
```

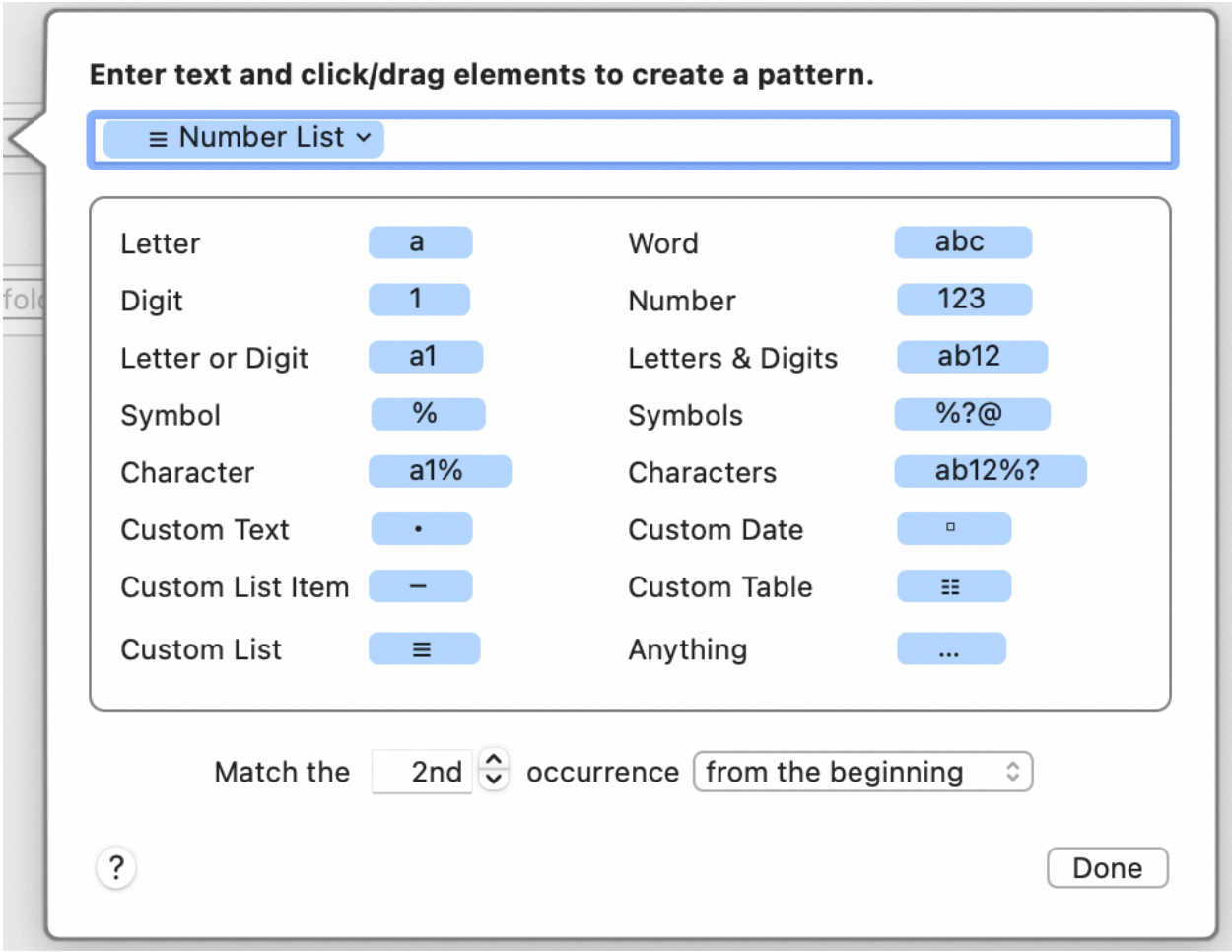
```
123,456,789
```

```
1.3,7800,3.41459
```

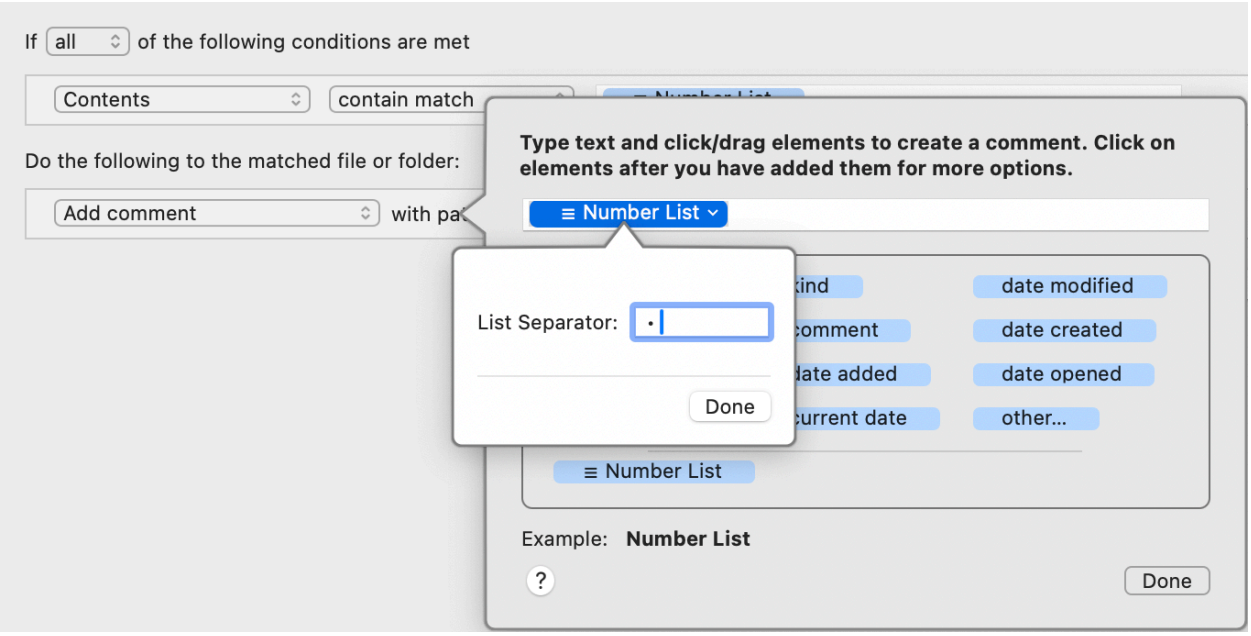
Suppose, for the sake of this example, that you want to drag many such text files into your monitored folder, but it's always the *second* list of numbers you're interested in. You want the rule to add all those numbers to a comment, separated by a bullet surrounded by spaces. Here's what the condition might look like:



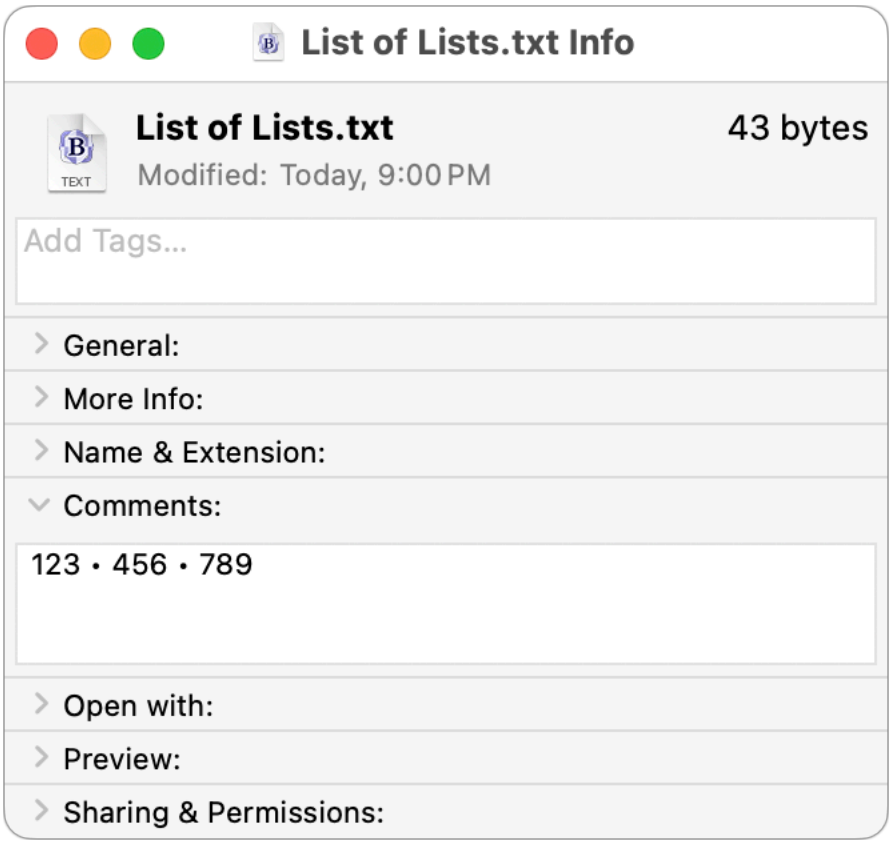
The condition is “Contents contain match” with a new custom list item as the pattern. That list, which we name “Number List,” in turn contains just the token 123, and the List Separator field at the bottom shows a comma. After we click Done, we make another change in the parent pattern popover: using the arrow buttons at the bottom, we set it to “Match the 2nd occurrence from the beginning.”



The action is “Add comment,” and its pattern consists of the Number List token we created in the condition. We click the little arrow on the right side of the token and, in the List Separator field, type a space, a bullet, and another space.



Now, if we add to this monitored folder a text file with contents like those given earlier, Hazel changes the Finder comments for that file to look like this:



SEE ALSO

Using Match Patterns in Conditions

Custom Text Attributes

Custom Date Attributes

Custom List Item Attributes

Custom Table Attributes

Reusing Custom Attributes in Condition Patterns

Using Patterns in Actions

Using Custom Attributes in Actions

Custom List Item Attributes

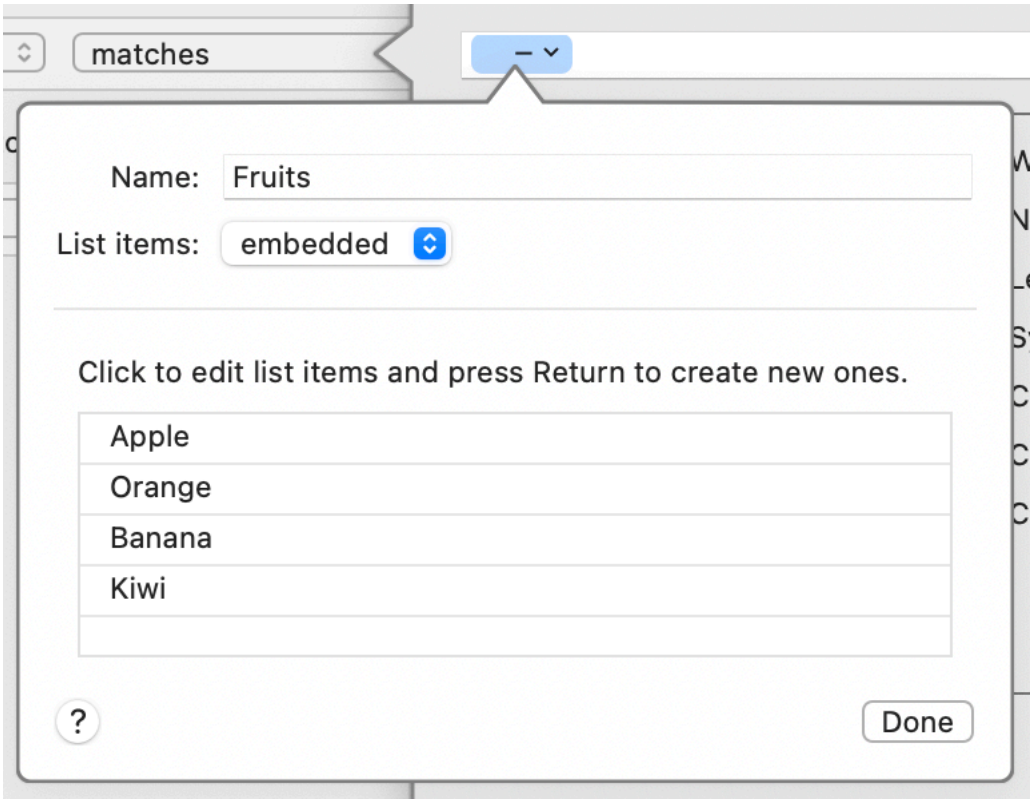
Although conventional attributes make it easy to create a condition that matches exact text (for example, in a filename) or a user-defined pattern, sometimes you may want your pattern—or an element of the pattern—to match any item from a list. For example, if you frequently work with files that include color names, you might want one condition that matches any of the words `red`, `green`, or `blue` while another condition matches any of the words `cyan`, `magenta`, or `yellow`. Or, you might want a rule to take action on all files that include the name of any employee in your department, or the name of any musical instrument in an orchestra. For situations like these, you can create your own list item attribute. Any item on the list then matches that attribute in your match pattern.

To create a custom list item attribute:

1. Create a condition with an attribute (such as “Name,” “Extension,” or “Contents”) that uses a match-related verb in the second pop-up menu (“matches”/“does not match,” “match”/“do not match,” or “contain match”/“do not contain match”).
2. Drag the “Custom List Item (—)” token to the pattern field.
3. Type a name for your custom list item attribute.
4. From the “List Items” pop-up menu, choose one of the following:
    - **“embedded”:** You enter the list items manually within Hazel.
    - **“from file”:** Hazel uses the contents of an existing text file for the list contents.
5. Follow the instructions below for the method you chose.

To enter embedded list items:

1. Rename “Item 1” as your first list item. To do this, select “Item 1,” wait a second, and then click it again to make it editable. Type in the first list item.
2. Add more list items. To do this, either select the first list item or hover over it and click the add icon. Type the next list item; repeat as needed.
3. Your finished list may look something like the image below. When you’re finished, click “Done,” and then make any further desired changes to your match pattern.



To use list items from a file:

1. Click “Choose,” navigate to a text file containing your list, and click “Open.”
2. By default, Hazel assumes each list item is on a separate line. If the list items in the text file are separated with a Tab character, choose “Tab” from the “Items separated by” pop-up menu. You can also type a character, such as a comma, in the “Items separated by” field.
3. If you select “Trim blank characters,” Hazel ignores whitespace characters on either end of any text from a field in a list item.
4. Click “Done,” and then make any further desired changes to your match pattern.

To rename a list item:

- Select the existing list item, wait a second, and then click it again to make it editable. Then type a new name and press Return.

To add a list item, do either of the following:

- Hover over an existing item and click the add ⊕ icon.
- With an existing item selected, press Return.

To reorder items in a list:

- Drag them up or down.

To remove a list item, do either of the following:

- Hover over it and click the remove ⛔ icon.
- Select it and press Delete.

SEE ALSO

Using Match Patterns in Conditions

Custom Text Attributes

Custom Date Attributes

Custom List Attributes

Custom Table Attributes

Reusing Custom Attributes in Condition Patterns

Using Patterns in Actions

Using Custom Attributes in Actions

Custom Table Attributes

A custom list item attribute matches any item from a list. But in some situations, you need the more sophisticated matching capabilities of a table (which you can think of as a list of rows). Each column in your custom table acts essentially like a list item attribute, with the added bonus that you can refer to other columns corresponding to the matching row in later conditions or in a rule's actions.

For example, suppose you had a table with just two columns: "Name" and "ID Number." If your rule condition looks for a match in the "Name" column, a "Rename" action could then rename the file with the person's corresponding ID number from the second column. You could also make a more complex rule with multiple conditions. For instance, say your table had a third column: "Favorite Color." Your first condition could match against the "Name" column, and a second condition could take action only if the text in the same row of the "Favorite Color" column is "purple." (For yet another example, see "Sort Files into Subfolders Based on Extension" in Using Custom Attributes in Actions.)

To create a custom table attribute:

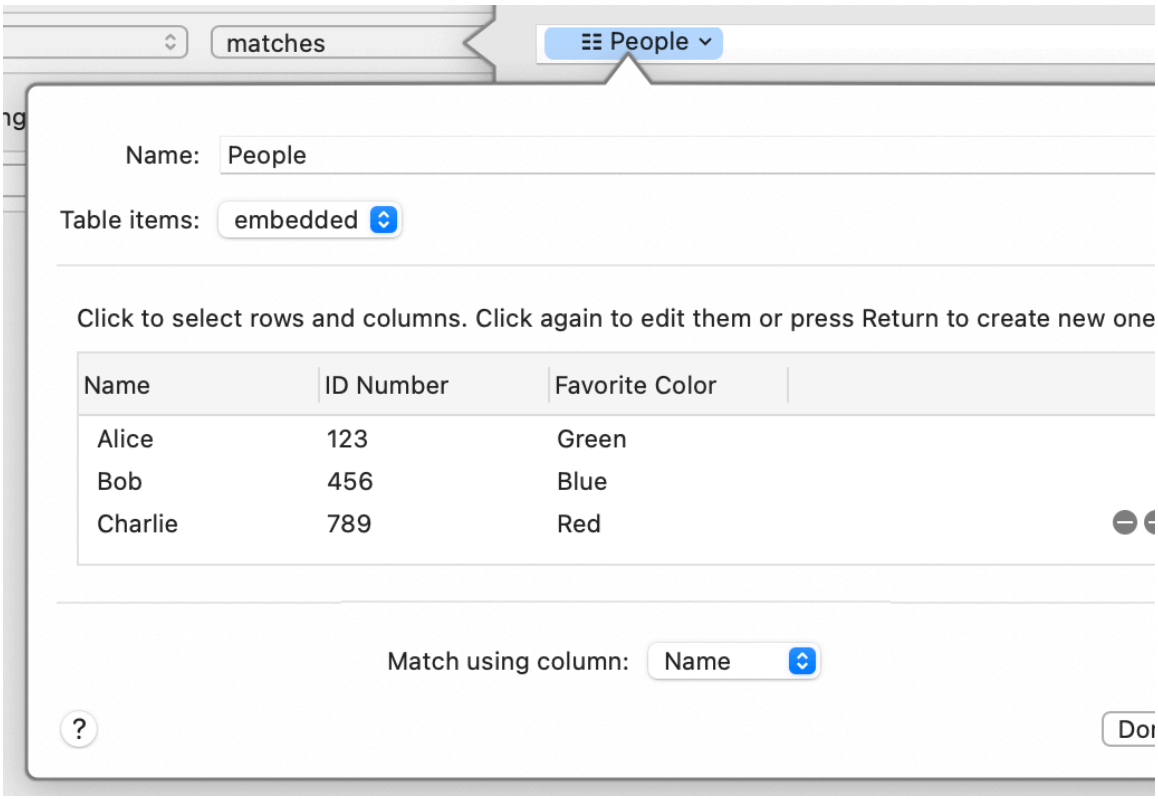
1. Create a condition with an attribute (such as "Name," "Extension," or "Contents") that uses a match-related verb in the second pop-up menu ("matches"/"does not match," "match"/"do not match," or "contain match"/"do not contain match").
2. Drag the "Custom Table (☷)" token to the pattern field.
3. Type a name for your custom table.
4. From the "Table Items" pop-up menu, choose one of the following:
    - **"embedded":** You enter the table items manually within Hazel.
    - **"from file":** Hazel uses the contents of an existing text file for the table contents.
5. Follow the instructions below for the method you chose.

To enter an embedded table:

1. Rename "Column 1" as your first column name. To do this, select "Column 1," wait a second, and then click it again to make it editable. Type in the first column name, and press Return.
2. To add another column header, with the first column selected, press Return. Then type a name for the second column and press Return. Repeat as needed

to create more columns.

3. In the first row, rename “Item 1” with your desired value. To do this, select “Item 1,” wait a second, and then click it again to make it editable. Type in the text you want, and press Return to accept it, or Tab to enter the text for the second column. Repeat for each item in that row.
4. Add more rows. To do this, either select the first row and press Return, or hover over the row and click the add icon. Enter the items in the new row; repeat as needed.
5. From the “Match using column” pop-up menu, choose which column this particular instance of the table should match against in your rule condition.
6. Your finished table may look something like the image below. When you’re finished, click “Done,” and then make any further desired changes to your match pattern.



To use a table from a file:

1. Click “Choose,” navigate to a text file containing your table, and click “Open.”
2. Hazel attempts to determine which character (such as a comma or tab) separates columns within in a row. If you need to change this character

manually, choose either “Newline” or “Tab” from the “Items separated by” pop-up menu. You can also type a character, such as a comma, in the “Items separated by” field. A preview shows how Hazel interprets the contents of your text file.

3. If the first line of the file contains column names, select “First line contains column names” to use those as the column names in your table. Otherwise, name the columns manually (if desired) by selecting a column header, waiting a second, clicking, and typing a new name.
4. From the “Match using column” pop-up menu, choose which column this particular instance of the table should match against in your rule condition.
5. If you select “Trim blank characters,” Hazel ignores whitespace characters on either end of any text from a field in a list item.
6. Click “Done,” and then make any further desired changes to your match pattern.

To rename a column:

- Select the existing column header, wait a second, and then click it again to make it editable. Then type a new name and press Return.

To add a column, do either of the following:

- Select an existing column header and press Return.
- Hover over an existing column header and click the add ⊕ icon.

To remove a column, do either of the following:

- Hover over it and click the remove ⊖ icon.
- Select it and press Delete.

To add a table row, do either of the following:

- Hover over an existing row and click the add ⊕ icon.
- With an existing row selected, press Return.

To remove a table row, do either of the following:

- Hover over it and click the remove ⊖ icon.
- Select it and press Delete.

SEE ALSO

Using Match Patterns in Conditions

Custom Text Attributes

Custom Date Attributes

Custom List Attributes

Custom List Item Attributes

Reusing Custom Attributes in Condition Patterns

Using Patterns in Actions

Using Custom Attributes in Actions

🧹 Reusing Custom Attributes in Condition Patterns

If you define a custom attribute in a condition, that attribute is available to subsequent conditions. The attribute is effectively shared between all the conditions that use it. You can even specify different patterns for each case.

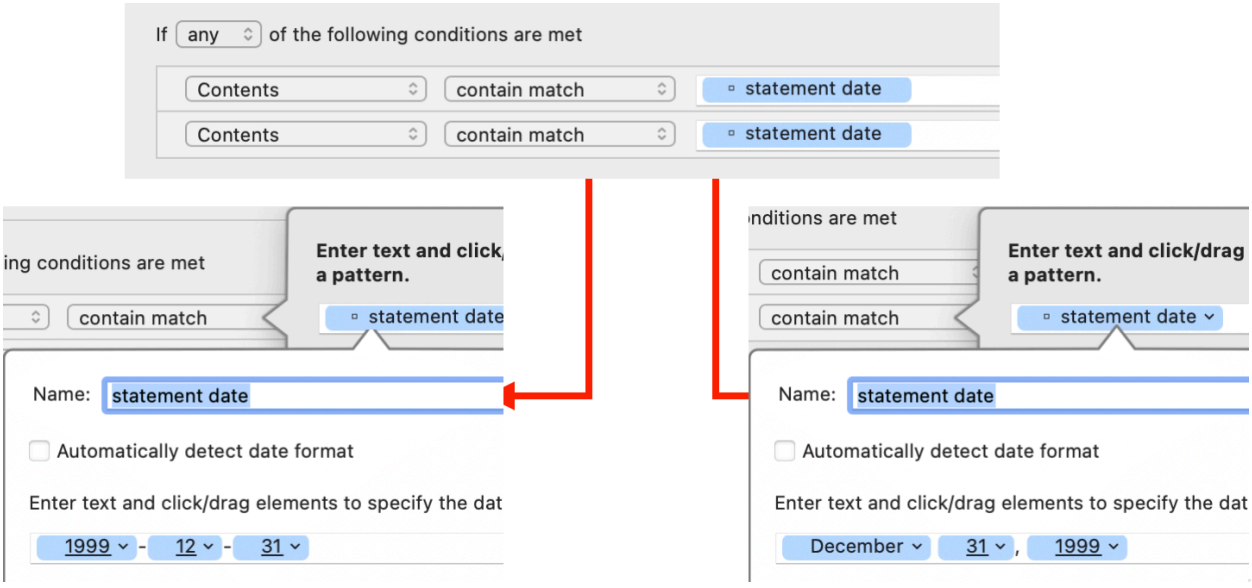
The first time the custom attribute successfully matches, it retains that value for the rest of the rule. That means subsequent uses of that attribute in the same rule must match the same value for any given file or folder.

For a text attribute, the exact text must match in subsequent usage. (The pattern for that token is ignored.) For a date attribute, a pattern can still be used to transform the text into a date, but the resulting date must match the previously matched date.

Note: You may also want to learn about reusing custom attributes in *actions*—see Using Custom Attributes in Actions.

Example 1: Using Two Different Date formats in a Custom Date Attribute

Take, for example, the following:

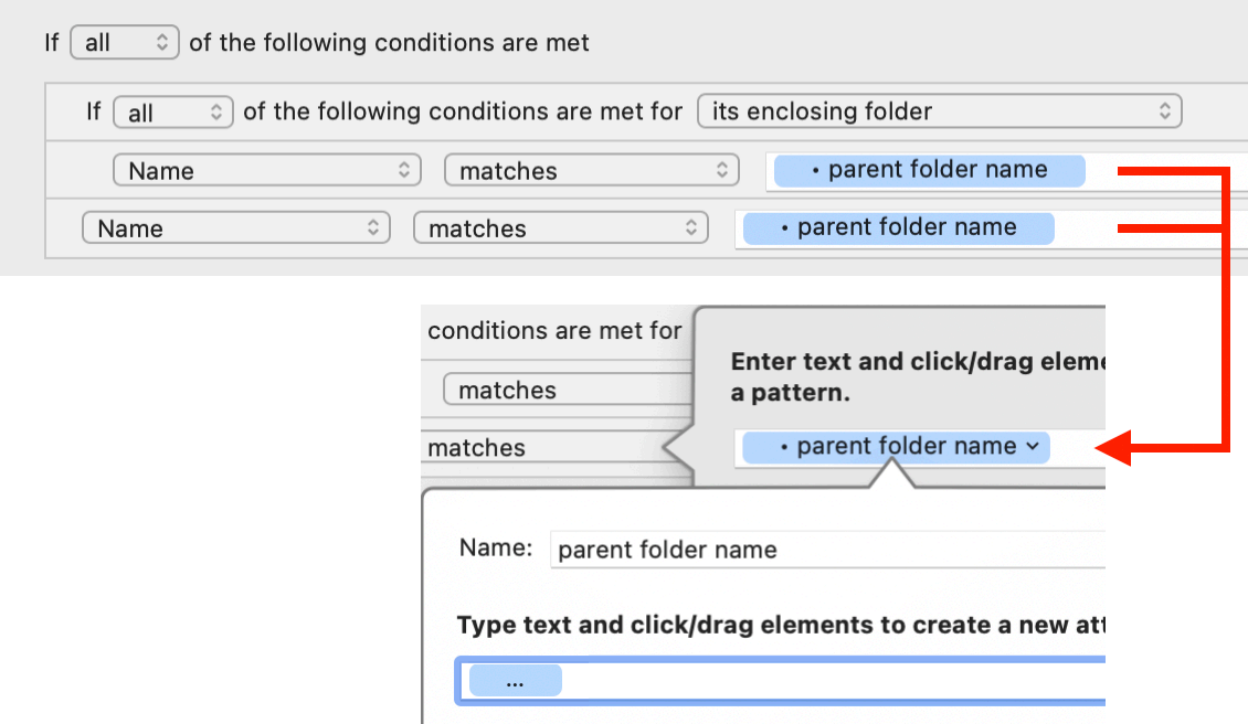


Although you can't see it in the main rule interface, each use of the custom attribute has a different date format. The top one is formatted as `1999-12-31` (below, left), and the bottom one is formatted as `December 31, 1999` (below, right).

Using a custom attribute like this enables you to match dates that may appear in different formats. In this example, you can match dates like `August 31, 2015` or `2015-08-31` in the same rule. Since the pop-up menu above the conditions is set to "Any," only one of the conditions needs to match. If you use the same custom attribute in an action for this rule, it will have the value of the first condition that matched.

Example 2: Matching a Filename with the Name of Its Parent Folder

Now let's look at another example:



Again, it's not shown in the main rule interface, but the custom attribute "parent folder name" here is defined as matching anything (…). The first instance matches the name of the file's parent folder, as indicated by the pop-up menu ("its enclosing folder") in the nested condition. The second instance matches the name of the file itself. The first time it matches, the name of the parent folder is captured into the custom attribute. When the custom attribute is used again in the second case, it must match the text that was captured before (note that the rule is set to match "All"). So, this example essentially

matches a file if its name is the same as its parent folder's name.

Using custom attributes in this way opens up new possibilities for correlating a file withs its parents or siblings, or correlating a folder with the files inside it.

SEE ALSO

Using Match Patterns in Conditions

Custom Text Attributes

Custom Date Attributes

Custom List Attributes

Custom List Item Attributes

Custom Table Attributes

Using Patterns in Actions

Using Custom Attributes in Actions

Action Reference

Each rule can have one or more actions that Hazel performs if the condition(s) match a given file or folder in the monitored folder. Hazel provides an extensive list of actions from which you can choose; most of these can be customized to meet your exact needs. (You may need to click the Options icon next to the action to display some of its options.)

Hazel offers the following actions:

Move

Move the file or folder to the specified destination folder.

Click the Options icon to display additional settings:

- In the “If file exists” section, select “rename the file,” “replace the existing file,” or “throw the file away” to tell Hazel how to treat the newly processed file if a file of the same name appears at the destination. Renaming appends a number to the newly processed file.
- Select the “Throw away if a duplicate” checkbox to move the new file to the Trash only if it is identical to (not merely named the same as) a file at the destination.
- If the matched item is a folder, you can also specify that some of the folder structure be copied over, as described in

This action will fail if the source and destination folders are the same.

Copy

Copy the file or folder to the specified destination folder.

Click the Options icon to display additional settings:

- In the “If file exists” section, select “rename the file,” “replace the existing file,” or “do not copy the file” to tell Hazel how to treat the newly processed file if a file of the same name appears at the destination. Renaming appends a number to the newly processed file.
- Select the “Do not copy if a duplicate” checkbox to refrain from copying the file if it is identical to (not merely named the same as) a file at the destination.
- If the matched item is a folder, you can also specify that some of the folder structure be copied over, as described in

Note that subsequent actions in the same rule apply to the copy and not the original.

Rename

Rename the file or folder. You can specify a pattern to use for the renaming, as described in . Patterns for renaming can also include user-configurable counters; see Using the Counter Attribute.

Sort into subfolder

Sort the file or folder into a subfolder of the monitored folder based on a specified pattern, as described in . If the folder does not exist, Hazel will create it as needed. Note that you can indicate additional levels of folders by using the ▸ token. Also, the “Anything” (…) attribute can be used to match against partial names of existing folders, although Hazel can’t create folders based on partial names.

Click the Options icon to display additional settings:

- In the “If file exists” section, select “rename the file,” “replace the existing file,” or “do not upload the file” to tell Hazel how to treat the newly processed file if a file of the same name appears at the destination. Renaming appends a number to the newly processed file.
- Select the “Do not copy if a duplicate” checkbox to refrain from copying the file if it is identical to (not merely named the same as) a file at the destination.

Sync

Sync the file or folder to the destination. This is similar to “Copy,” except that “Sync” copies only those items that have been added since the last time the rule ran. Syncing is one-way; files changed in the destination folder are not copied back to the monitored folder. Deletions from the source folder can be synced to the destination, but qualifications apply; see Syncing Folders for more information.

Upload

Upload the file or folder to a server via FTP, SFTP, or WebDAV; see Specifying Upload Options for details.

Click the Options icon to display additional settings:

- In the “If file exists” section, select “rename the file,” “replace the existing file,” or “do not upload the file” to tell Hazel how to treat the newly processed file if a file of the same name appears at the destination. Renaming appends a number to the newly processed file.
- If the matched item is a folder, you can also specify that some of the folder

structure be copied over, as described in

Add tags

Add the specified tags to the file or folder. You can also add Dynamic Tags, in which the tags are created on the fly based on file metadata and custom attributes you create. See for more details on how to use patterns to create Dynamic Tags.

Remove tags

Remove the specified tags from the file or folder. You can remove all the existing tags on a file removed by using the “Any Existing Tags” option. You can also specify Dynamic Tags, in which the tags are created on the fly based on file metadata and custom attributes you create. See for more details on how to use patterns to create Dynamic Tags.

Set color label

Set the color label on the file or folder. Click the ✕ icon to remove the color label. This action sets the tags associated with the given color, and removes any other colored tags.

Add comment

Add a Spotlight-searchable comment to the file or folder. If you include the “Comment” attribute in your pattern, it will include any existing comments in your new comment; otherwise, Hazel replaces the existing comment with your new comment.

Toggle extension

Set whether the file’s extension is hidden or visible in the Finder.

Toggle lock

Unlock or lock a file; choose “off” or “on” from the second pop-up menu.

Archive

Convert the file or folder to a Zip file, just as if you used the Finder’s Compress feature. Note that subsequent actions in the same rule operate on this new archive; the original item is moved to the Trash.

Unarchive

If the file is an archive, it is decompressed. Note that subsequent actions in the same rule operate on the decompressed file or folder; the original archive file is moved to the

Trash. This action will fail if the file is not an archive.

Open

Open a file or folder. You can specify a specific application or use “Default Application” to open the file with the application that would normally open it when you double-click it. You should exercise caution when opening files from untrusted sources; structure your rule conditions appropriately. Also, if you use this action, you should disable any browser options that will cause it to open files; otherwise, files may open twice. Click the Options icon to display an additional setting: “Bring to front.” Select this checkbox to bring the newly opened item to the front; otherwise, Hazel opens it but leaves it in the background.

Show in Finder

Open a new Finder window with the file or folder selected. Click the Options icon to display an additional setting: “Bring to front.” Select this checkbox to bring the Finder to the front; otherwise, Hazel selects the newly processed item but leaves the Finder in the background.

Make alias

Create an alias to the file or folder in the specified folder.

Import into Music

Import the file into Music. The original file remains in place, so depending on your Music settings, you may want to move it to its proper place beforehand (if you have Music set to not copy files) or throw it away afterwards (if you have Music set to copy files). Note that Music will open if it is not already running.

Import into Photos

Import the file into Photos. The original file remains in place, so depending on your Photos settings, you may want to move it to its proper place beforehand (if you have Photos set to not copy files) or throw it away afterwards (if you have Photos set to copy files). Note that Photos will open if it is not already running.

Import into TV

Import the file into the TV app. The original file remains in place, so depending on your TV settings, you may want to move it to its proper place beforehand (if you have TV set to not copy files) or throw it away afterwards (if you have TV set to copy files).

Run Shortcut

Runs a Shortcut that’s already configured in the Shortcuts app. See Using Shortcuts.

Run AppleScript

Run an AppleScript. For details on writing AppleScript for Hazel, see Using AppleScript or JavaScript.

Run JavaScript

Run a JavaScript. For details on writing JavaScript for Hazel, see Using AppleScript or JavaScript.

Run Automator workflow

Run an Automator workflow. For details on writing Automator workflows for Hazel, see Using Automator.

Run shell script

Runs a shell script or any command-line program. For details on writing shell scripts for Hazel, see .

Pause

Wait for a specified number of seconds or minutes before moving on to the next action. This can be useful in situations where you want to perform multiple actions, but an earlier one (such as running an AppleScript) may take some time to complete.

Run rules on folder contents

This action causes Hazel to run the *full* rule list on the files and folders contained in this one. You should make sure you have a condition specifying “Type is Folder,” as this action will fail if run on a file. If subfolders match a rule with this action, Hazel continues to descend as needed. But be careful about allowing Hazel to descend too far, as this will delay processing of subsequent files. For more information, see .

Continue matching rules

This action tells Hazel to continue matching against subsequent rules instead of stopping. Normally Hazel stops once it finds a match, but this action indicates that rule evaluation should continue. Note that you cannot continue evaluation if the file or folder is moved out of the monitored folder. Therefore, you can’t use this action in conjunction with the “Move” or “Sort into subfolder” actions.

Display notification

Display a notification on screen using Notification Center (see Notifications). You can also set a sound to play along with the notification by clicking the Options icon.

Notifications can include not only static text but also dynamic patterns, as described in .

Ignore

This action does nothing. It's useful for ensuring that certain files are not processed. For instance, you can set a rule at the beginning of a list to exclude certain files from any subsequent rules. This action makes sense only when it is the sole action in a rule.

SEE ALSO

Attribute Reference

Using Patterns in Actions

Using AppleScript or JavaScript

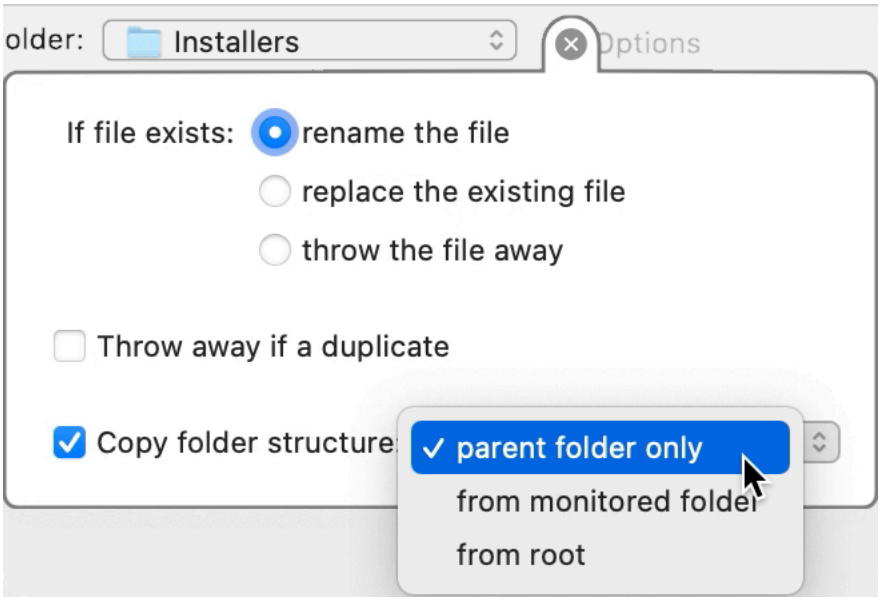
Using Automator

Using Shell Scripts

Copying Folder Structure

Certain actions, such as Move, Copy, and Upload, have an option to copy the folder structure, and not just the file or folder being processed. This option enables you to mirror the folder structure at the file's destination.

To enable this option, click the Options ⋯ icon in the action, select "Copy folder structure," and then choose the structure you prefer from the pop-up menu, as shown below. (By default, this option is disabled and no folder structure is copied, which means any file or folder processed by your rule is placed at the top level of the destination folder.)



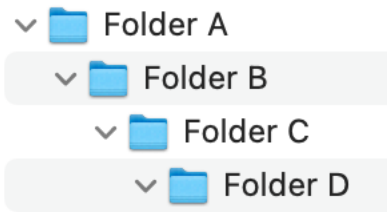
When "Copy folder structure" is enabled, the structures you can choose are:

- **"parent folder only":** Use only the name of the parent folder of the file or folder being processed. (This is the default choice.)
- **"from monitored folder":** Use the path from the monitored folder to the parent folder of the file or folder being processed.
- **"from root":** Use the full path from the top level of the volume (not including the volume name) to the parent folder of the file or folder being processed.

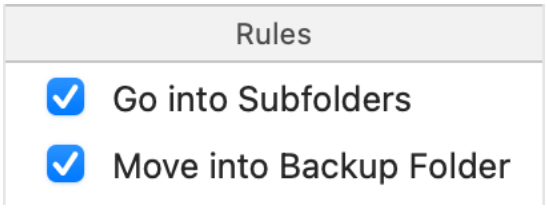
Folder Structure Examples

To illustrate how the three folder structure options work, let's look at an example (which you can duplicate on your own Mac and try yourself).

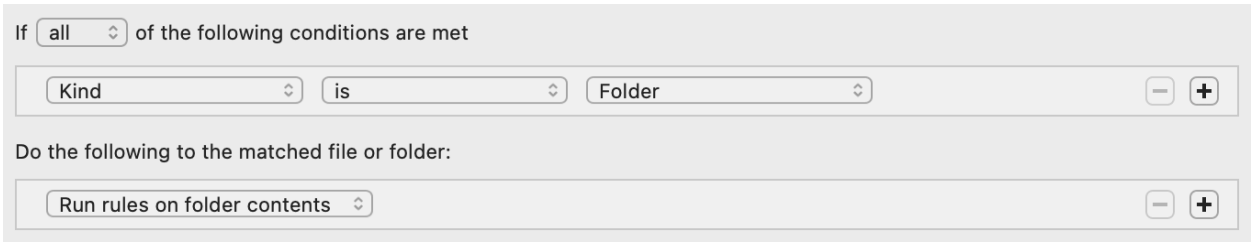
We begin with the following structure somewhere on your disk:



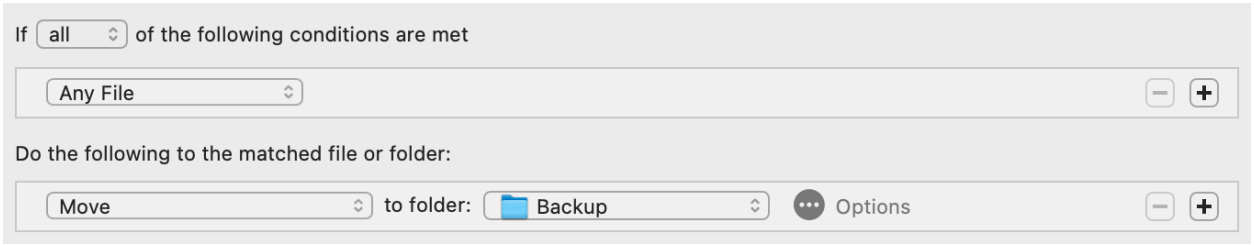
Next, we set up Folder A as a monitored folder and then create two rules:



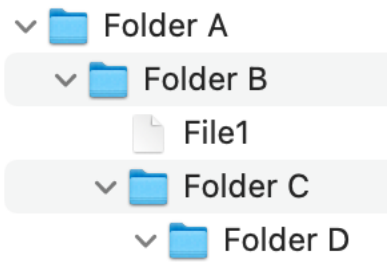
The first rule, which tells Hazel to process subfolders, looks like this:



The second rule uses the "Move" action (although "Copy" or "Upload" would also work), in this case using a folder named "Backup" as the destination:



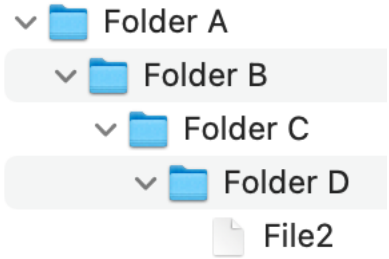
Now, we move File1 into Folder B (as shown below) and let the rule run.



File1 is moved to the Backup folder in one of three ways, depending on the option used:

- **“parent folder only”: Folder B**. That is, the new file’s parent folder (Folder B) is recreated within the Backup folder, and the file is moved to that folder.
- **“from monitored folder”: Folder B**. In this case, the results happen to be exactly the same as with the “parent folder only” option, because the monitored folder itself (Folder A) isn’t included in the path. (If File 1 were instead placed at the top level of Folder A, the file would instead be moved to the top level of Folder B.)
- **“from root”: Folder A ▸Folder B**. That is, the new file’s path from the monitored folder (Folder B, in side Folder A) is recreated within the Backup folder, and the file is moved to the new Folder B.

Next, suppose we move File2 into Folder D (as shown below) and let the rule run.

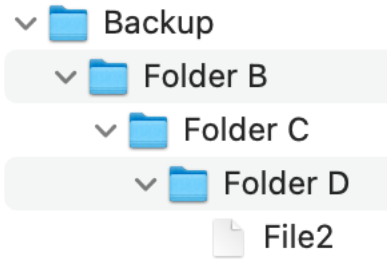


File2 is again moved to the Backup folder in one of three ways, depending on the option used:

- **“parent folder only”: Folder D**. That is, the new file’s parent folder (Folder D) is recreated within the Backup folder, and the file is moved to that folder.
- **“from monitored folder”: Folder B▸Folder C▸Folder D**. That is, the new file’s path from the monitored folder (Folder D, inside Folder C, inside Folder B) is recreated within the Backup folder, and the file is moved to the new Folder D.
- **“from root”: Folder A▸Folder B▸Folder C▸Folder D**. That is, the new file’s full

path (Folder D, inside Folder C, inside Folder B, inside Folder A) is recreated within the Backup folder, and the file is moved to the new Folder D.

So, if we are moving files to a folder called Backup and using the “from monitored folder” option (the second example just above), we would end up with a folder structure like this:



By getting the path up to (but not including) the monitored folder, Hazel can reproduce the path from Folder B downward in the new location.

SEE ALSO

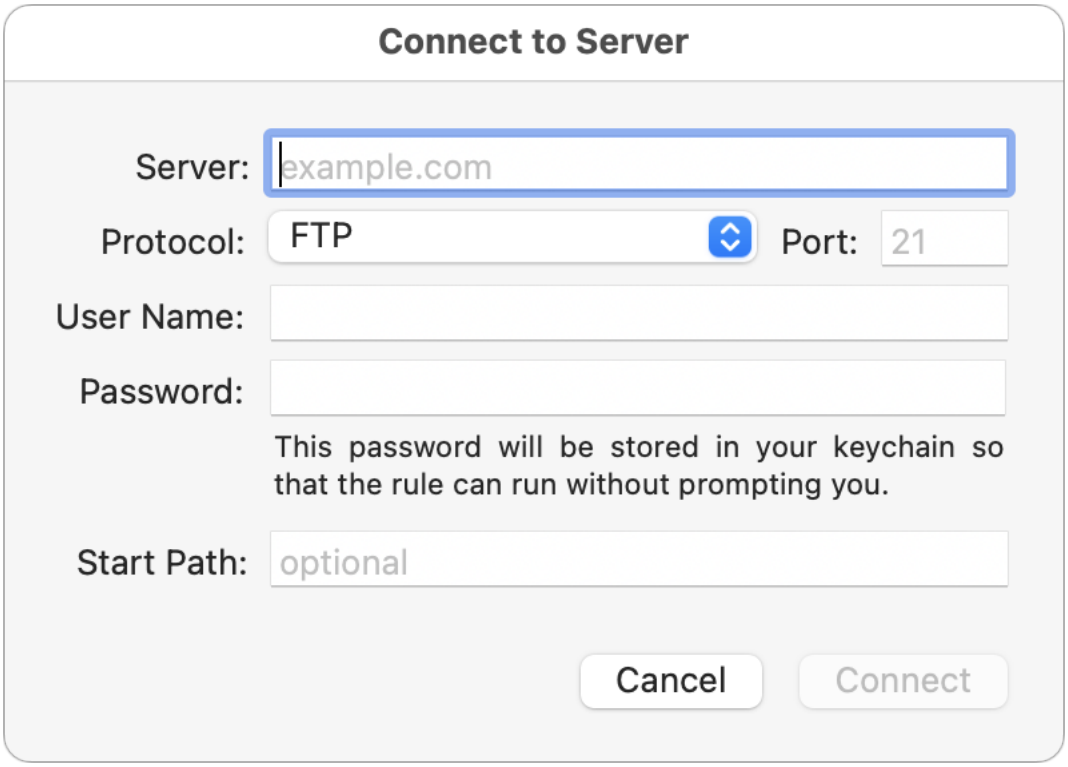
Action Reference

Relocating Folders

Specifying Upload Options

Hazel's Upload action lets you upload files or folders that match your conditions to an FTP, SFTP, or WebDAV server.

After you choose "Upload" from the first pop-up menu, choose "Add Server" from the second one. A dialog appears, in which you can fill in the details for the destination.



To set up a server:

1. Fill in the "Server," "User Name," and "Password" fields, which should be self-explanatory. (As the dialog says, Hazel stores the password in your Mac's keychain.)
2. Use the "Protocol" pop-up menu to choose your protocol—the options are:
    - "FTP"
    - "FTP (Implicit TLS/SSL)"
    - "FTP (Explicit TLS/SSL)"
    - "SFTP" (Note: If you choose SFTP as the protocol, a key button appears next to the "Password" field, which you can optionally click to

select an SSH key file in lieu of a password.)

- “WebDAV”
- “WebDAV (HTTPS)”

3. Hazel automatically fills in the default port based on the protocol you choose, but if your server is set up differently, you can enter a new number in the “Port” field to override it.
4. The “Start Path” is optional. If you already know the exact path to your desired destination folder on the server, you can enter it in this field, but you need not do so, because you’ll have the opportunity to navigate to it in the next step.
5. Click “Connect.” Hazel connects to the server with your credentials and, assuming the connection was successful, displays a dialog showing either the top-level folder in your account on the server or the folder of the “Start Path” (if you filled it in). If you want Hazel to save files anywhere other than the folder currently visible, navigate to it now. (Or, to create a new folder, click “New Folder.”)
6. Click “Open.”

Hazel adds the server to the action.

Once you’ve defined a server for any action, that server also appears as a potential destination in the pop-up menu for other “Upload” actions (either in this rule or in other rules).

To edit an existing server, select it from the “to server” pop-up menu in the rule action and then choose “Edit Server” from the same pop-up menu.

SEE ALSO

Action Reference

Copying Folder Structure

Using Patterns in Actions

Replacing Folders

The Move, Copy, and Sync commands require you to specify a destination folder. If you've selected a particular destination folder for more than one rule and you want to replace it with a different destination folder across all your rules that reference it, you can do so using Hazel's "Replace Folder" command. This allows you to change the location of a folder in all your rules at once instead of having to edit each rule individually.

Keep in mind that relocating a folder affects only which folders your rules use as destinations; it does not actually change any folders on your Mac, nor does it change which folders Hazel monitors.

To replace a folder across all your rules:

1. Choose Hazel > "Replace Folder…"
2. The list on the left contains all the folders used by your rules. Select the one you want to replace.
3. Click the "Choose…" button.
4. In the file dialog that appears, select the new folder to be used.
5. Select either "only this folder" if you want only the original folder to be replaced, or "this folder and its subfolders" if you want to also replace any subfolders of the original folder in use by any rules. This latter option allows you to relocate a tree of folders together.
6. Click the "Replace" button.

All rules using that folder are updated immediately.

SEE ALSO

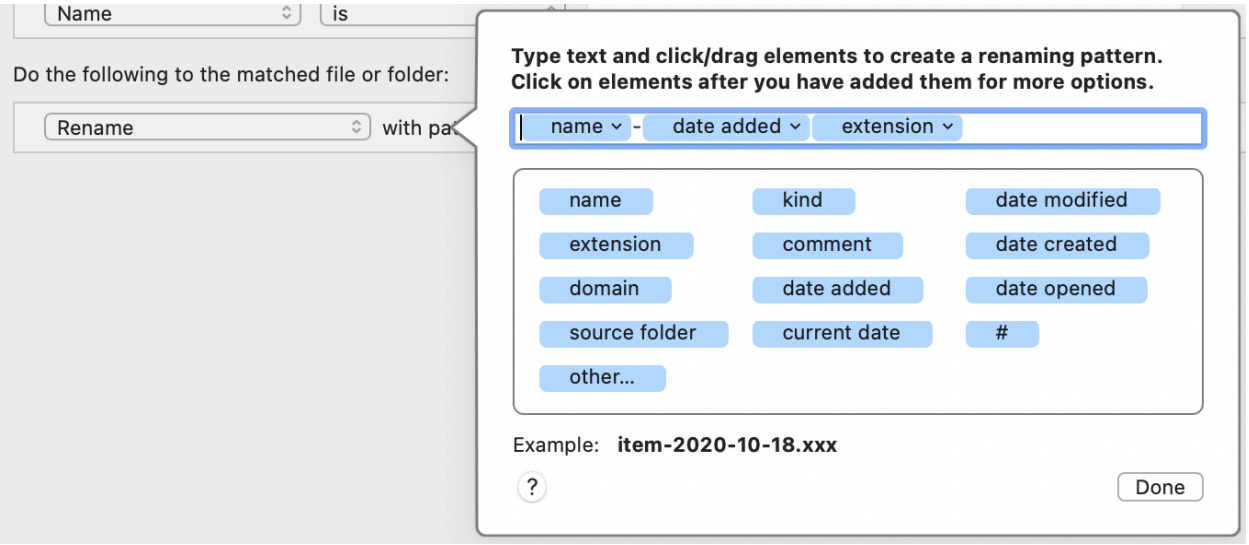
Action Reference

Copying Folder Structure

Using Patterns in Actions

For certain actions (“Rename,” “Sort into subfolder,” “Add tags,” “Remove tags,” “Add comment,” and “Display notification”), you must specify a pattern—a template that can include both static text and special attributes. The action uses your pattern to generate text when it runs. For example, when renaming a file, you can add the date the file was added. Since you don’t know what the date will be until the rule runs, and that the date varies from one file to the next, you can use an attribute in your pattern to represent that date. When the rule runs, it substitutes the actual date.

When you choose an action that requires a pattern, a popover appears as shown here. (Note that the available attributes vary from one action to another.)



Fill in the field at the top of this popover with the pattern you want. You can use a combination of plain text you type yourself (to include static elements as part of your pattern) and *tokens*—blue bubbles representing any of numerous attributes of the file being matched. Using a token in your pattern tells the rule to substitute that item’s actual value when the action runs. To add a token to your pattern, either click it (once) or drag it to the desired position in the pattern. The “Example” field at the bottom of the popover shows a sample of how the pattern will look when the action runs (of course, the actual text will vary). When you’re finished editing your pattern, click “Done.”

Special Attributes

A few attributes have special behaviors when it comes to patterns:

- **“other”:** “Other” is a stand-in for any available Spotlight attribute. When you add it to the field, a popover appears in which you can select which Spotlight attribute you want that token to represent. Depending on the attribute, you may have more options available after adding it to your pattern.
- **“Add tags,” “Remove tags”:** When you choose one of these actions and click in the field, a pop-up menu appears. You can choose either a specific tag name (click “Show All” if the one you want isn’t visible), Any Existing Tags (for the “Remove tags” action), or *Dynamic Tag*, which lets you build a pattern from text and tokens, much like the one shown above.
- **Custom attributes:** Custom text, date, list, and table attributes you defined in a condition can also be used in actions. For details, see Using Custom Attributes in Actions.

Attribute Formatting

If you hover your pointer over any attribute after adding it to the field, you’ll see that an arrow appears on its right side. This indicates you can click the token to display a pop-up menu with more formatting options. Depending on the type of attribute, you may see one or more of the following options:

- **For any attribute:**
  - Choose “Replace Text” to configure simple find-and-replace pairs (for example, “Replace `jpg` with `JPEG`”), which alter the text from what it would otherwise be. You can have more than one such pair if needed.
  - Choose “Set Default” to set up a default value for that attribute. If the attribute is blank/empty, then this default value will be used instead.
  - Choose “Set Maximum Length” to set the maximum length of an attribute, which can be useful, for example, if you need to limit filename length. To set a specific number of characters, use the arrow buttons or type in a number of characters; use “No limit” (the default) for any number of characters. Hazel truncates that element after the number of characters specified. Once you’ve set a maximum length, the token displays a double-arrow ↹ symbol next to the attribute name:

.
- **For text-based attributes:**
  - Use the commands “No Case Change,” “Title Case,” “lowercase,” or

"UPPERCASE" to change the case of the text.

- **Treat as File Path:** Available in Sort into Subfolder patterns, this allows you specify a character as a path separator so that the text can be split up into folders, from the top down. For example, if you had a text attribute with the value "some/example" and you specified "Unix (/)" as the path separator, the resulting file would be sorted into the "example" folder within the "some" folder. This would be the equivalent of the pattern "some ▸ example."

- **For number-based attributes:**
  - **"Number Format":** Use the commands "0," "00," "000," or "0000" on this submenu to set the minimum number of digits in a number to one, two three, or four, respectively; Hazel adds leading zeros, if necessary, to achieve that length. (Remember, this is a minimum; the counter can go as much higher than the number of digits you select as necessary.) To use a larger number of digits, choose "Edit Number Format," enter a number of digits, and click Done. Choose "No Digit Change" to leave the number of digits unchanged.
  - **"Thousands Separator":** Choose "Hide" or "Show" from this submenu to hide or show the thousands separator for numbers with four or more digits, as set in System Settings > "Language & Region" > "Number format." (For users in North America, the thousands separator is usually a comma; other regions may use a period or a space.) To leave the thousands separator, if any, unchanged, choose "No change."
  - **"Adjust Number":** If you want the adjust the actual number up or down, you can add or subtract any number from what the attribute would otherwise be. To do this, choose Adjust Number and then use the arrow buttons to set a positive or negative value.

- **For attributes representing a list (such as "Other" > "Attachment Names," "Keywords," or "Performers"):**
  - **"First Item,…,Last Item"; "First Item"; "Last Item"; "Select List Item":** Choose one of these commands to use only use the first item, the last item, a specific list item (such as the second from the beginning or ninth from the end), or all items, separated by commas (the "First Item,…,Last Item" option).
  - **"Text Format":** This submenu displays the same capitalization options that appear for text-based attributes. Note that this format applies across to items in the list.

- **For attributes representing dates:**
  - **“Edit Date Pattern”:** Change the way the date is formatted by using a wide variety of tokens (plus manually entered characters) to create your own pattern. See for more information.
  - **“Adjust Date”:** Choose this command to add to or subtract any number of years, months, days, hours, and/or minutes from the date.
- **For domains:** The “domain” attribute has special options to indicate how much of the domain to show: **“www.example.com”** (the whole thing), **“example.com”** (bare domain only, no subdomain), or **“example”** (the portion of the domain before the dot and top-level domain).

SEE ALSO

Using Match Patterns in Conditions

Using Custom Attributes

Custom Text Attributes

Custom Date Attributes

Custom List Attributes

Custom List Item Attributes

Custom Table Attributes

Reusing Custom Attributes in Condition Patterns

Action Reference

Using the Counter Attribute

Editing Date Patterns

Using Custom Attributes in Actions

Using the Counter Attribute

When editing a pattern for the "Rename" action, the counter # attribute becomes available.

This attribute represents a number that increases by one each time the action runs. You can use this in a renaming action to number files. For example:

name - # extension

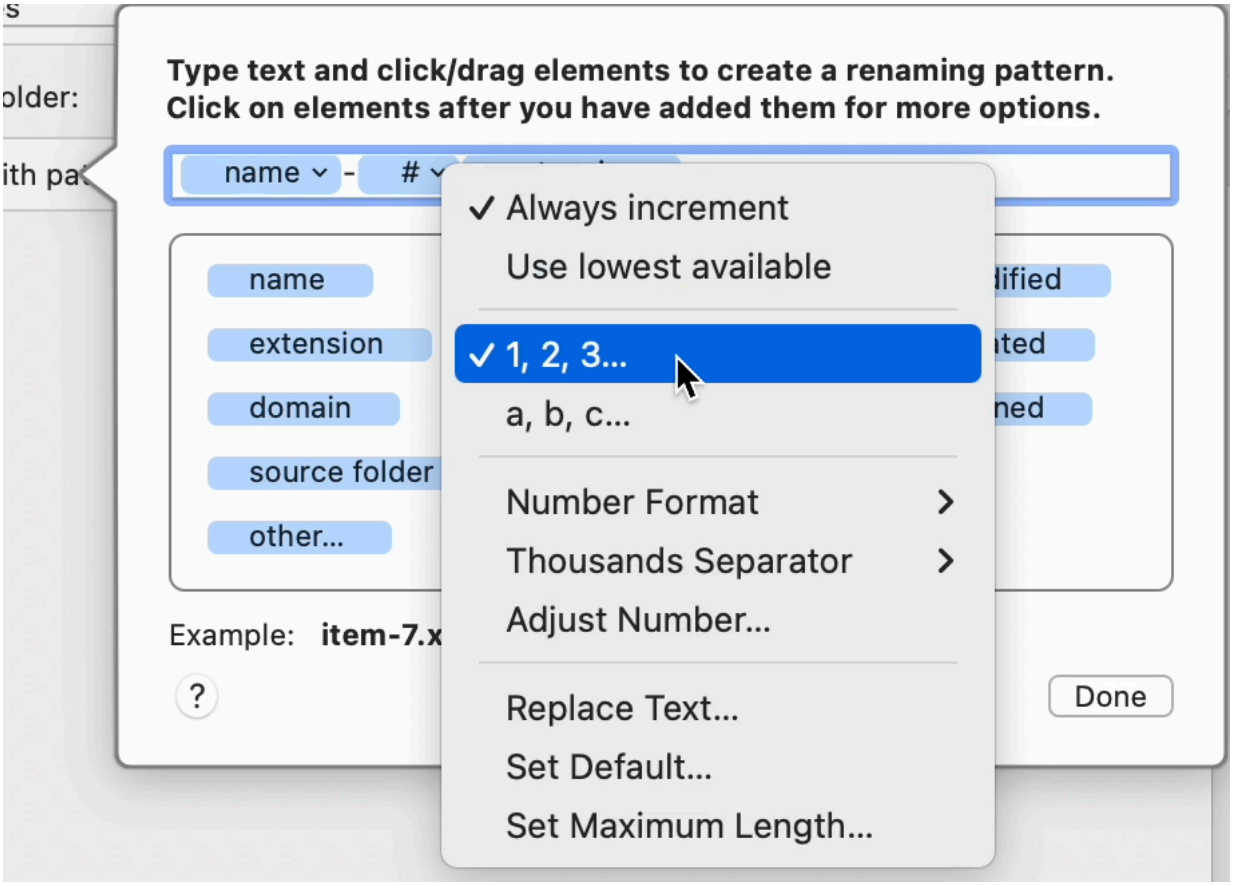
In the pattern above, the counter follows the filename and a hyphen, and comes before the extension. So, if your rule repeatedly encounters and matches new files named `somefile.doc`, this action creates files named `somefile-1.doc`, `somefile-2.doc`, and so on. But it works the same way even when filenames differ. For example, suppose your condition matches files whose names begin with Z, and you drag the files `Zing.txt`, `Zip.doc`, and `Zoom.xls` into the monitored folder. A “Rename” action with the pattern above would name them `Zing-1.txt`, `Zip-2.doc`, and `Zoom-3.xls`, respectively.

Note: Each folder has its own counter.

Customizing Counters

You can customize the numbers (or letters) used in a counter in several ways.

First, if you click the tiny arrow in the counter # token, a pop-up menu appears with numerous options.



Your options on this menu include “Number Format,” “Thousands Separator,” “Adjust Number,” “Replace text,” “Set Default,” and “Set Maximum Length” (as is true for any number-based attribute; see Using Patterns in Actions), plus:

- **“Always increment”/“Use lowest available”:** By default (using the “Always increment” setting), the counter always increases; it never backtracks or fills in holes in the sequence. But if you remove filename-1 and filename-2 from your monitored folder while leaving filename-3 there, you can instead choose “Use lowest available” to force Hazel to choose the lowest number that’s not already in use by a file in the monitored folder.
- **“1, 2, 3…”/“a, b, c…”:** Choose “1, 2, 3…” (the default) to use numbers, or “a, b, c…” for letters.

When you’re finished customizing your counter, click “Done.”

Resetting a Counter

If you want to reset that counter so that the next time it runs it goes back to 1, you can

right-click (or Control-click) the monitored folder in the Finder and choose Reset Counter from the contextual menu. (The command does not appear unless the folder already has a counter from a previous rename action.)

SEE ALSO

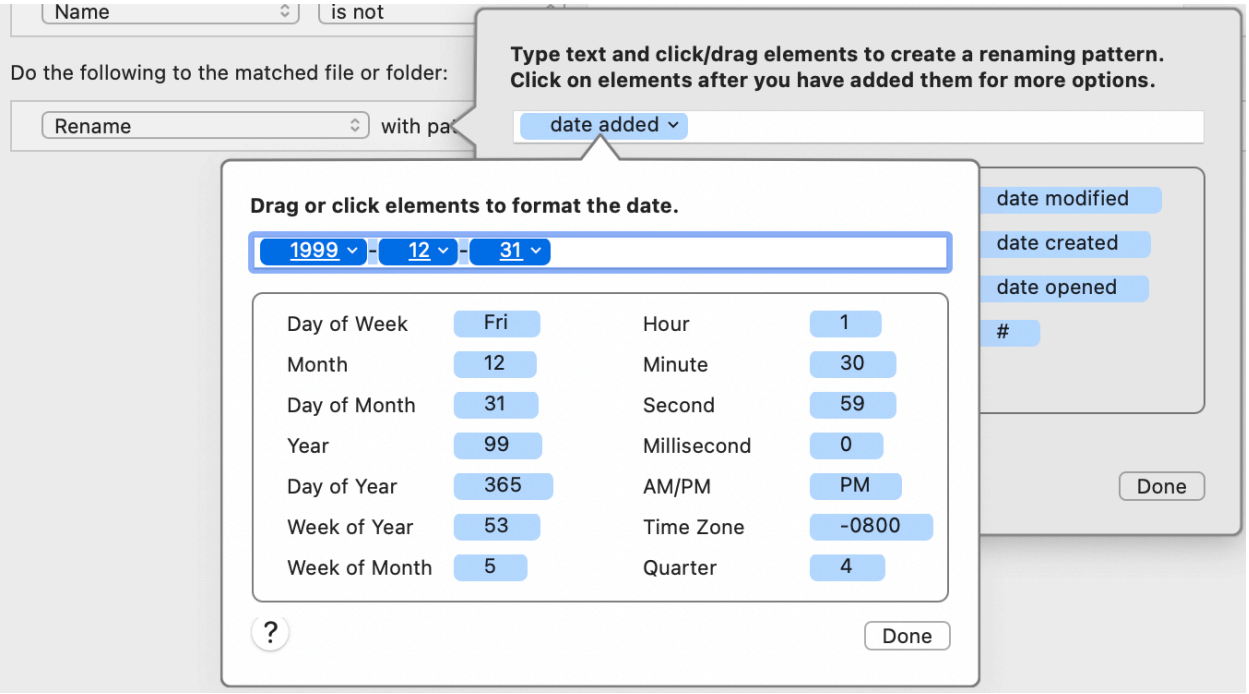
Using Patterns in Actions

Editing Date Patterns

Editing Date Patterns

Certain actions (such as “Rename” and “Sort into subfolder”) require patterns, and those patterns in turn can use any of numerous attributes—including several that are based on dates (“date added,” “date created,” “date modified,” “date opened,” and “current date”). You can edit any of these date attributes to express the date using any format you like. So, if the date is the last day of the year 1999, you can create a date pattern that represents that date as “1999-12-31,” “Dec 31 ’99,” “Day 365 of the year A.D. 1999,” or pretty much any other format you like.

To edit a date pattern, first put a date-based token (such as “date added”) in a pattern for an action. Then click the arrow on the right side of the token and choose “Edit Date Pattern” from the pop-up menu. You’ll see a popover like this one:



As in Hazel’s other popovers for creating patterns, you can type in static text or click date tokens to add them to your pattern; Hazel substitutes the real values for those tokens when the rule runs. For more information on editing patterns, see

.

Be aware of the following about date patterns:

- The date pattern need not contain every token to constitute a full date. You can specify just the year and month or just the hour, if you choose.
- If you hover your pointer over a date token after adding it to the field, a small

arrow appears on its right side. As with other tokens, the arrow indicates that you can click it to display a pop-up menu with formatting options. For example, if you drag in a “Month” token, you’ll see options “1” (the month number without a leading zero), “01” (the month number with a leading zero added if necessary), “Jan” (a shortened version of the month name), and “January” (the full month name).

- An underline indicates the minimum number of digits. For example, you can choose 1 or 01 for day or month, or 99 or 1999 for year. For days or months with only one digit, Hazel adds a leading zero when you choose 01; for years, Hazel includes all four digits with 1999 but only the last two with 99.

When you’re finished editing, click “Done.”

SEE ALSO

Using Patterns in Actions

Using the Counter Attribute

Using Match Patterns in Conditions

Custom Date Attributes

Using Custom Attributes in Actions

Actions can refer back to custom attributes from match patterns (see Using Match Patterns in Conditions and Using Custom Attributes). This enables you to do things like:

- Rename files with only part of the original filename.
- Display a notification that includes the first line of text in a file.
- Sort files into any of several subfolders based on their extension.
- Apply tags based on the contents of comments, or vice versa.
- Open with a different app depending on the domain name in the source URL.

Let's look at three examples to illustrate the process of reusing a custom attribute in an action.

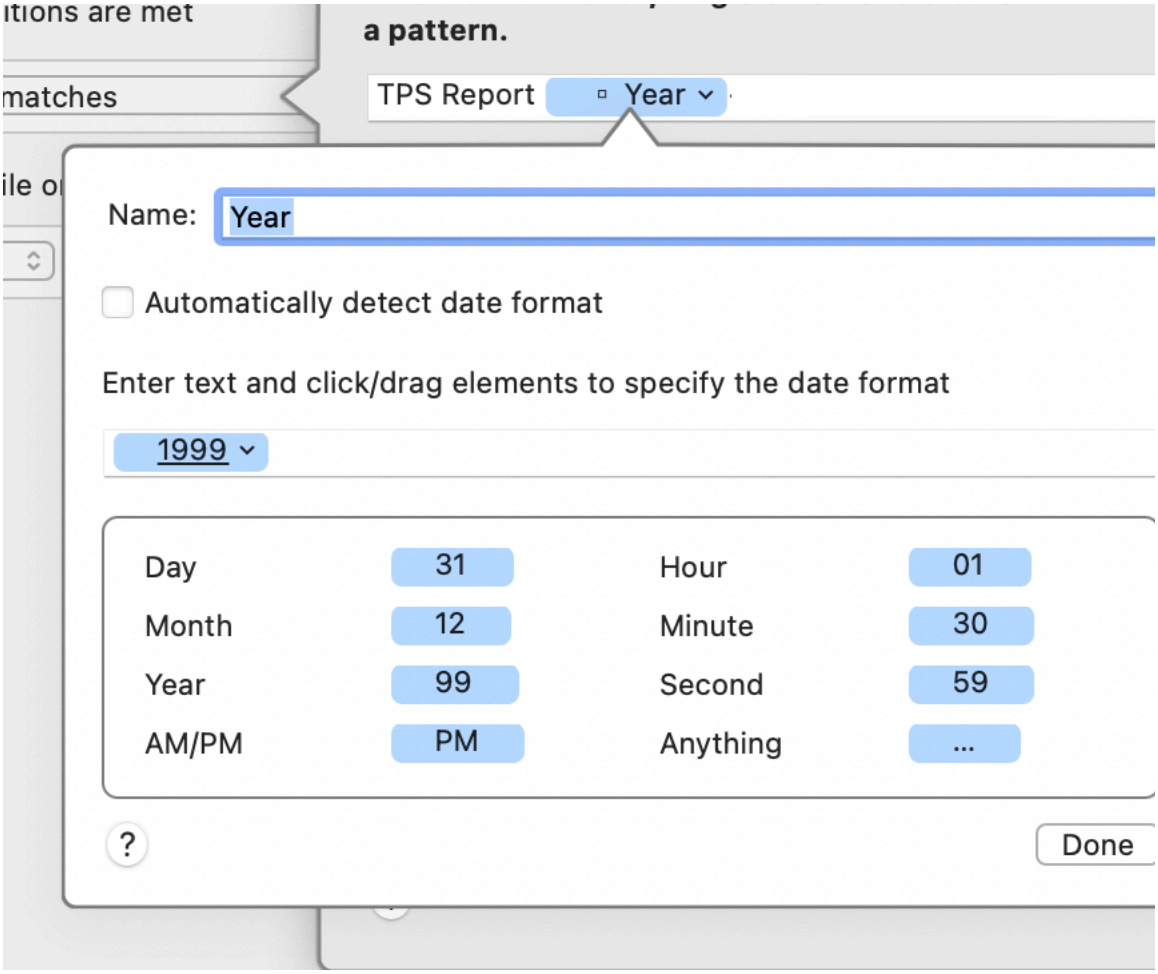
Note: You may also want to learn about reusing custom attributes in *other conditions*—see Reusing Custom Attributes in Condition Patterns.

Rename Files with Part of the Original Filename

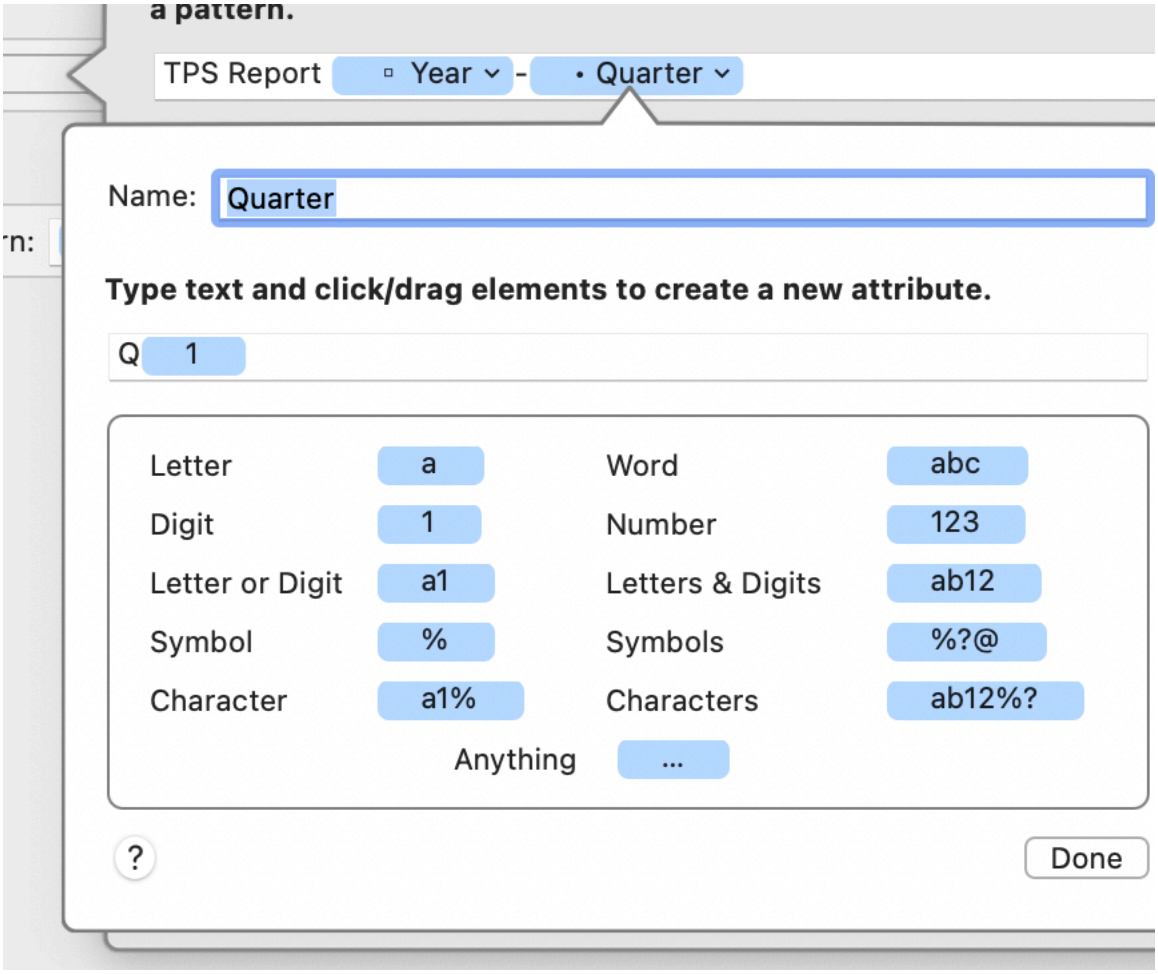
Suppose a monitored folder contains files with names like `TPS Report 2023-Q3.pdf` and `TPS Report 2024-Q2.pdf`, and you want a rule to rename them with just portions of the original filenames, but rearranged—like `Q3 2023.pdf` and `Q2 2024.pdf`. Here's how you might do that.

First, set up the condition:

- Choose "Name" from the first pop-up menu and "matches" from the second.
- Click in the pattern field and type the text `TPS Report` (followed by a space).
- Drag the "Custom Date ▫" token into the pattern, type in the name "Year," deselect the checkbox "Automatically detect date format," and drag the "Year (99)" token to the attribute field.
- Click the arrow on the right side of the 99 token and choose "1999" from the pop-up menu to specify that we're looking for four-digit dates. The custom Year token should then look like this:

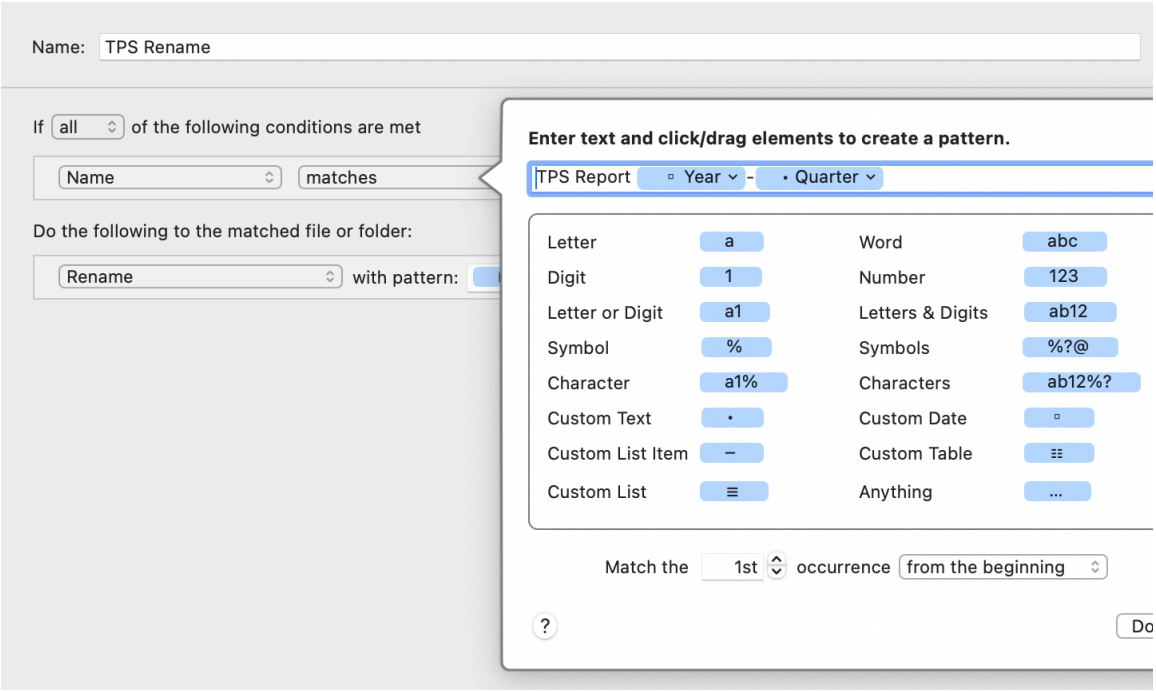


- Click “Done” to dismiss the “Year” popover, then type a hyphen.
- Drag the “Custom Text •” token into the pattern, type in the name `Quarter`, and in the attribute field type the letter `Q` and insert the “Digit (1)” token. It should look like this:



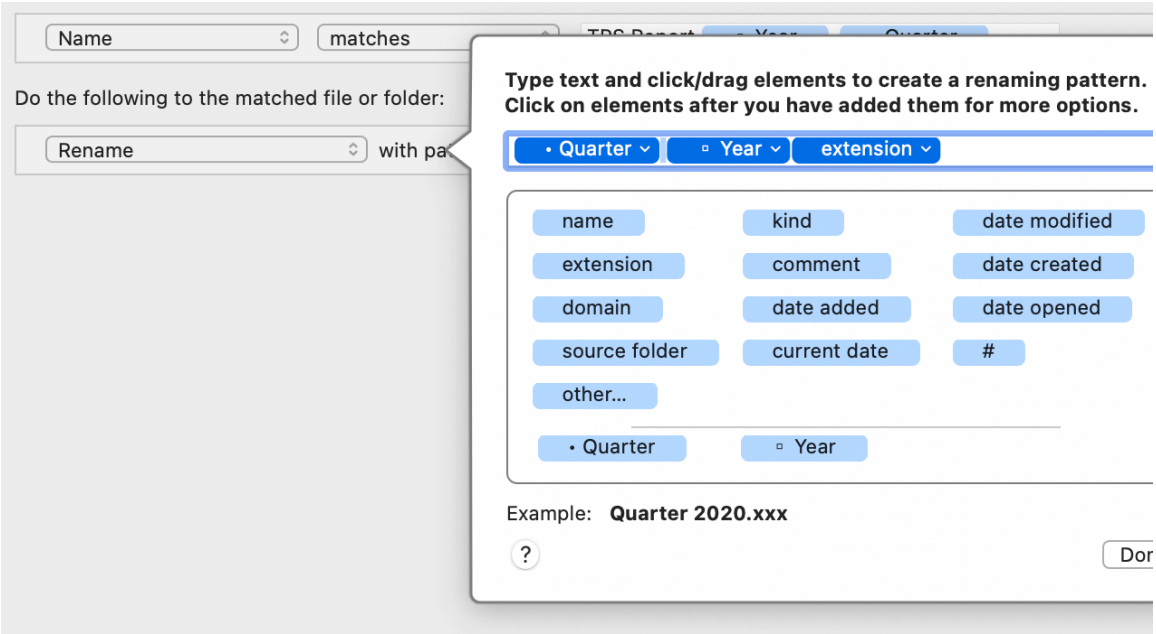
- Click “Done” to dismiss the “Quarter” popover. then click “Done” again to dismiss the match pattern popover.

  The finished condition will look like this:



Next we move on to the action:

- In the action row, choose “Rename” from the first pop-up menu.
- Click in the pattern field and delete the “name” attribute (but leave “extension” in place).
- From the bottom of the attribute list, drag the “• Quarter” custom text token to the beginning of the pattern field, then type a space and drag the “▫ Year” token to the pattern field, just to the left of “extension,” like so:



- Click the arrow on the right side of the "▫ Year" token and choose "1999" from the pop-up menu to specify that we're using four-digit dates in the replacement name.
- Click "Done," and then click "Save."

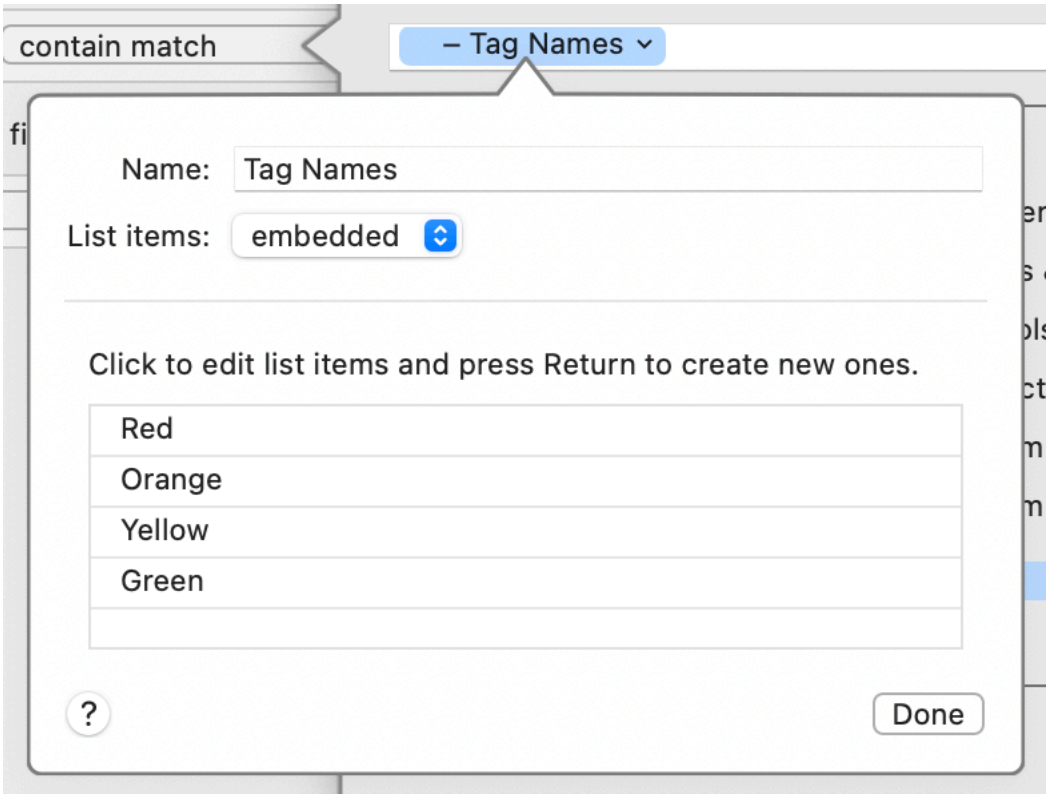
This rule will now rename files with names like `TPS Report 2023-Q3.pdf` to something like `Q3 2023.pdf`.

Rename a File or Folder Matching a List of Tags

Next, we use a list item attribute to match files or folders tagged with any word on a list, and then add the tag to the filename. For example, if the file `nothing.txt` has the tag "Red," and "Red" is on your list, that file will be renamed `nothing Red.txt`.

First, create the condition:

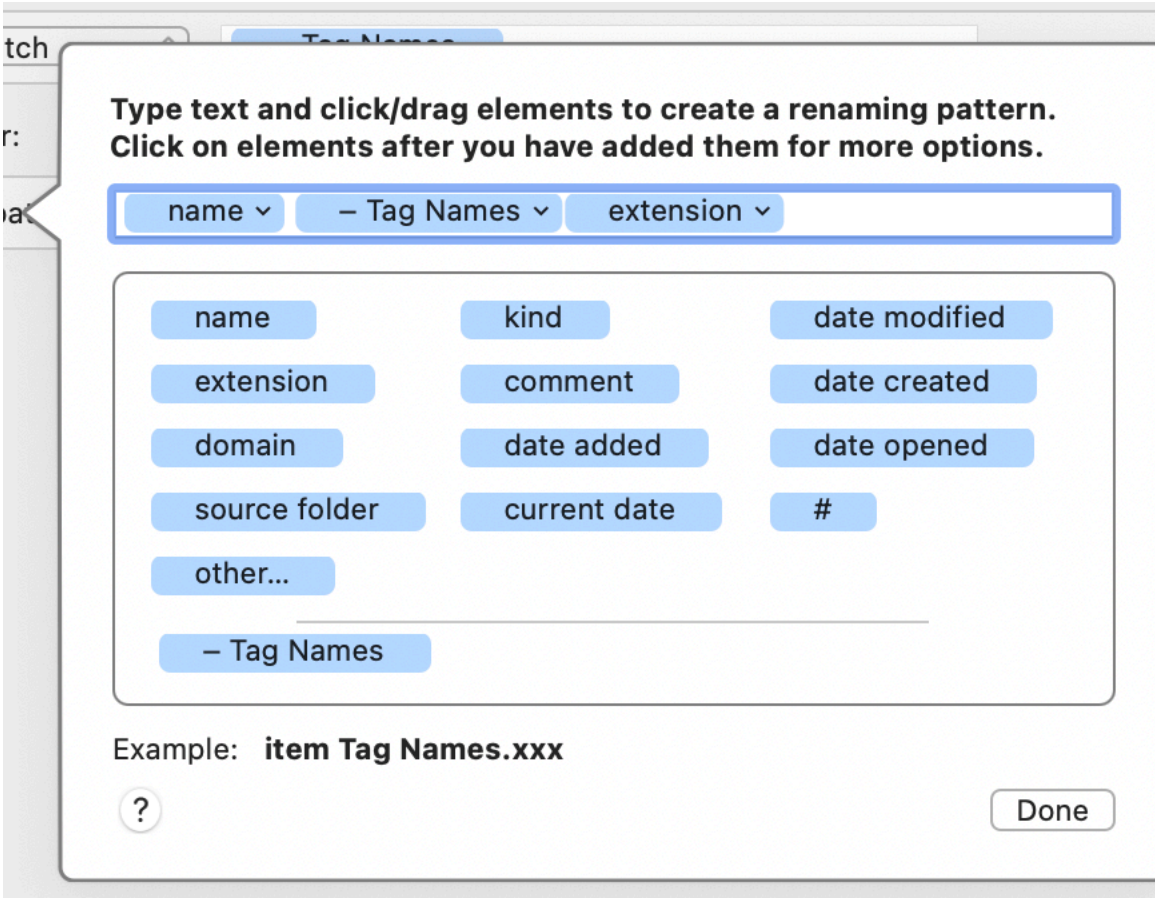
- Choose "Tags" from the first pop-up menu and "contain match" from the second.
- Click in the pattern field, and click the "Custom List Item (—)" token to add it to the pattern.
- Name the custom list item "Tag Names," and following the instructions in Custom List Item Attributes, fill in the list as shown below. (For this example, we're deliberately including only a portion of the default Finder tag names.)



- Click “Done” to dismiss the custom list attribute item popover, then click “Done” again to dismiss the match pattern.

Now create the action:

- From the first pop-up menu, choose “Rename.”
- Click in the pattern field and type a space after the “name” token.
- Drag the “— Tag Names” custom list item attribute up to the pattern field, inserting it after the space and before the “extension” token.



- Click “Done,” and then click “Save.”

This rule will now add the text of any tag on your list to the filename of items in the monitored folder.

Sort Files into Subfolders Based on Extension

The final example rule sorts files into one of three subfolders: “Images” if the extension is `.jpg`, `.jpeg`, or `.png`; “PDFs” if the extension is `.pdf`, or “Text” if the extension is `.txt`.

To set this up:

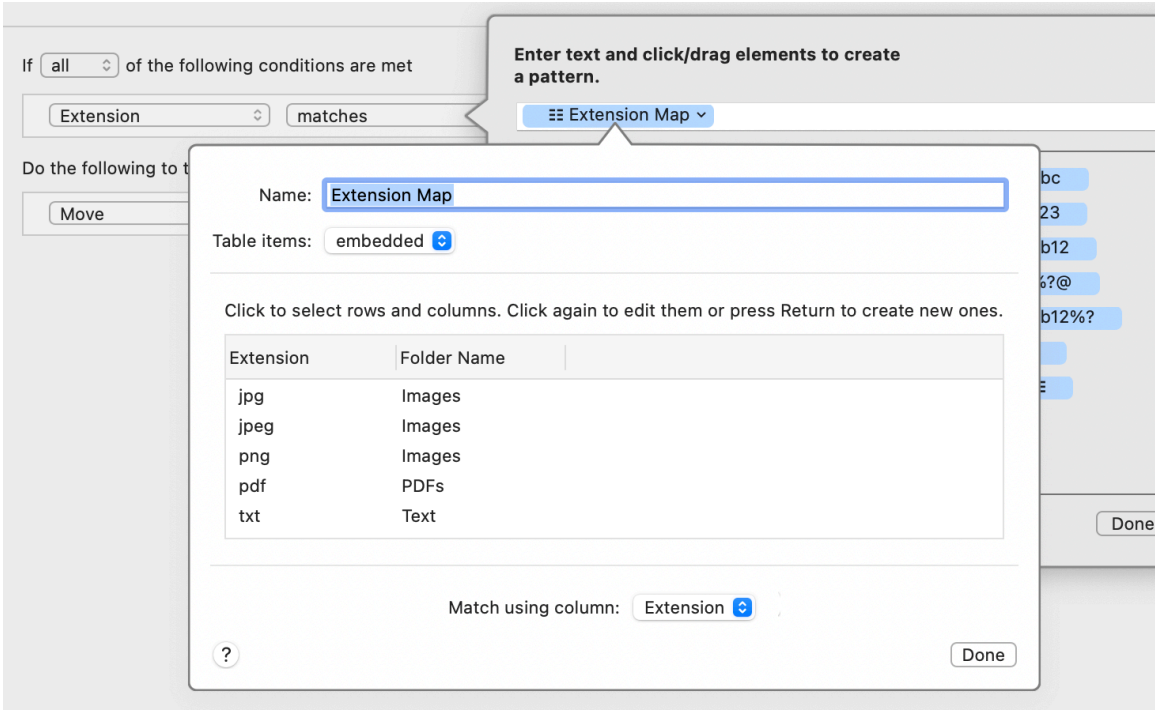
- Begin by creating folders with the above names (“Images,” “PDFs,” and “Text”) inside the folder you’re monitoring, so you can refer to them later.

Next, create the condition:

- Choose “Extension” from the first pop-up menu and “matches” from the

second.

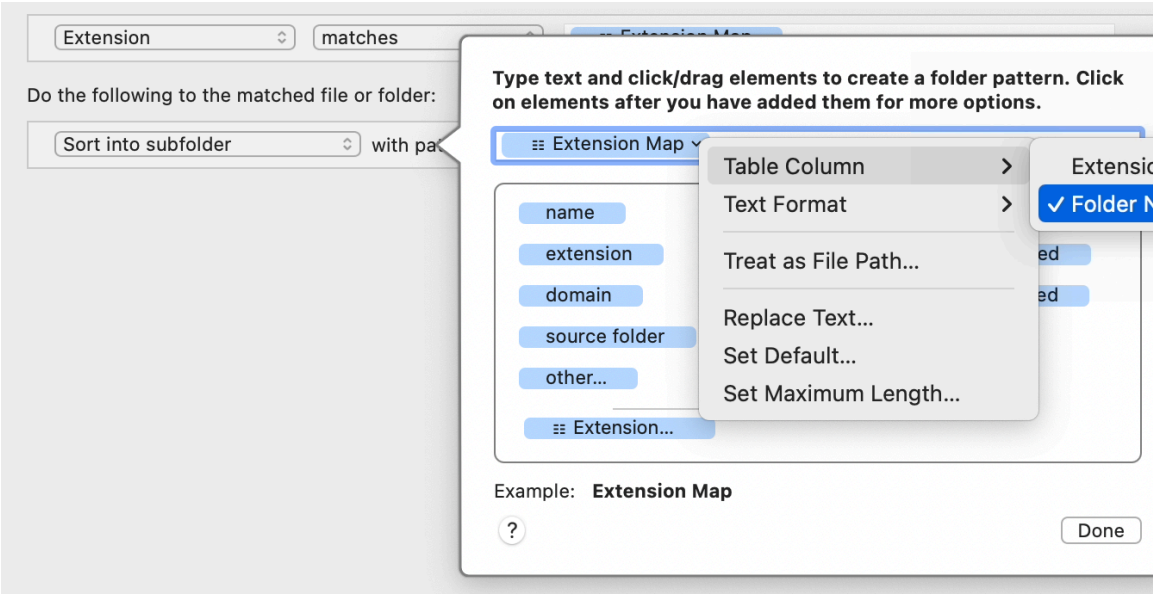
- Click in the pattern field, and click the “Custom Table (☷)” token to add it to the pattern.
- Name the custom table “Extension Map,” and following the instructions in Custom Table Attributes, fill in the table as shown below. (You’ll note that we have three different rows that say “Images” for the Folder Name, since there are three extensions we want to match against for that folder.)



- Click “Done” to dismiss the custom table popover, then click “Done” again to dismiss the match pattern.

Now create the action:

- From the first pop-up menu, choose “Sort into subfolder.”
- Click in the pattern field and drag the “☷ Extension Map” custom table up to the pattern field.
- Click the arrow on the right side of the “☷ Extension Map” token and choose “Table Column” > “Folder Name” from the pop-up menu to indicate that you want the names from the “Folder Name” column to be used for sorting. Your action should now look like this:



- Click “Done,” and then click “Save.”

This rule will now sort items into subfolders based on the extensions and folder names you specified in the table.

SEE ALSO

Using Patterns in Actions

Using the Counter Attribute

Using Match Patterns in Conditions

Using Custom Attributes

Custom Text Attributes

Custom Date Attributes

Custom List Attributes

Custom List Item Attributes

Custom Table Attributes

Reusing Custom Attributes in Condition Patterns

Using Patterns in Actions

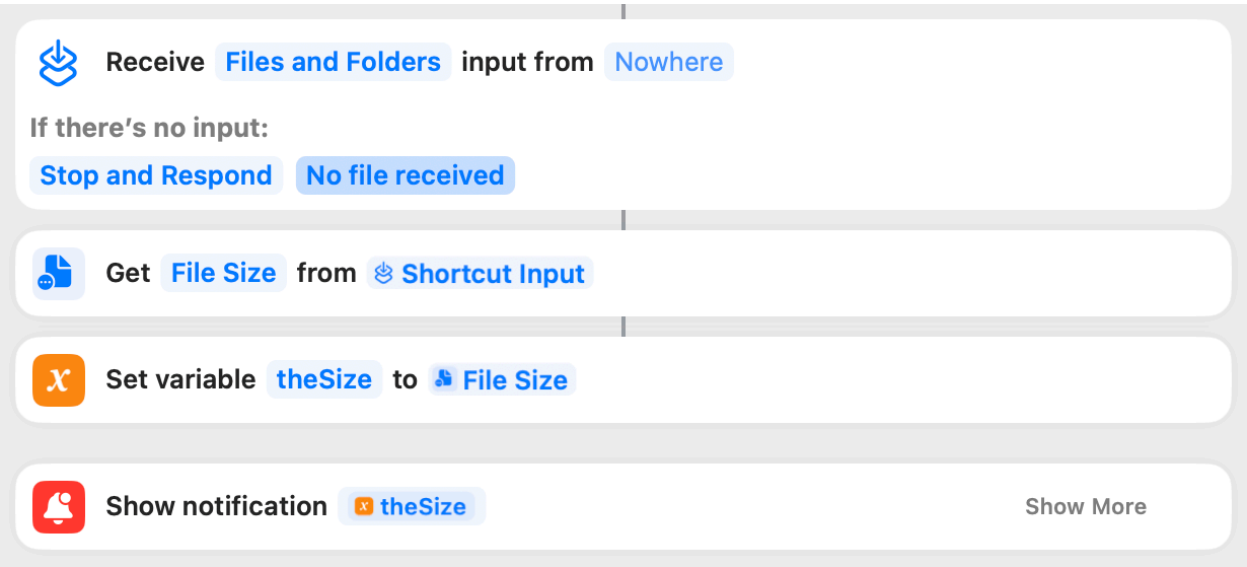
Using Shortcuts

A Hazel rule can run a Shortcut if you choose the “Run Shortcut” action and then choose one of your existing Shortcuts from the second pop-up menu. (Shortcuts cannot be used in a rule condition, only in an action.)

Standalone shortcuts that run without any input require no special modifications. However, if you want your Shortcut to act on whatever file or folder Hazel just processed, you must configure it to receive files and folders and then get information from that input. You can use any combination of input sources, or none at all:

1. Choose View > “Show Shortcut Details” and select the Use as Quick Action checkbox to display the Receive action at the top of your shortcut. Click “Images and 18 more” then deselect everything except Files and Folders.
2. Under “If there’s no input,” click Continue, choose Stop and Respond from the pop-up menu, and add a response (such as “No file received”).
3. Add your shortcut actions as usual, being sure to refer to Shortcut Input for information about the file or folder that was processed.
4. Optional: When you’re finished, you can return to View > “Show Shortcut Details” and deselect the Use as Quick Action checkbox (and any others that are also selected), so the Receive action lists “Nowhere” as the source of input.

This simple example Shortcut displays a notification showing the size of the file or folder that was just processed.



SEE ALSO

Using AppleScript or JavaScript

Hazel rules can use AppleScript or JavaScript in either of two ways: to provide logic for a condition (the “Passes AppleScript” or “Passes JavaScript” attribute), or to perform an action when a rule matches (the “Run AppleScript” or “Run JavaScript” action). If you need to accomplish a task that Hazel’s conditions or actions alone don’t cover, AppleScript or JavaScript may provide a solution.

Note: Other than the fact that the two languages have different syntax, their overall capabilities within Hazel are essentially the same.

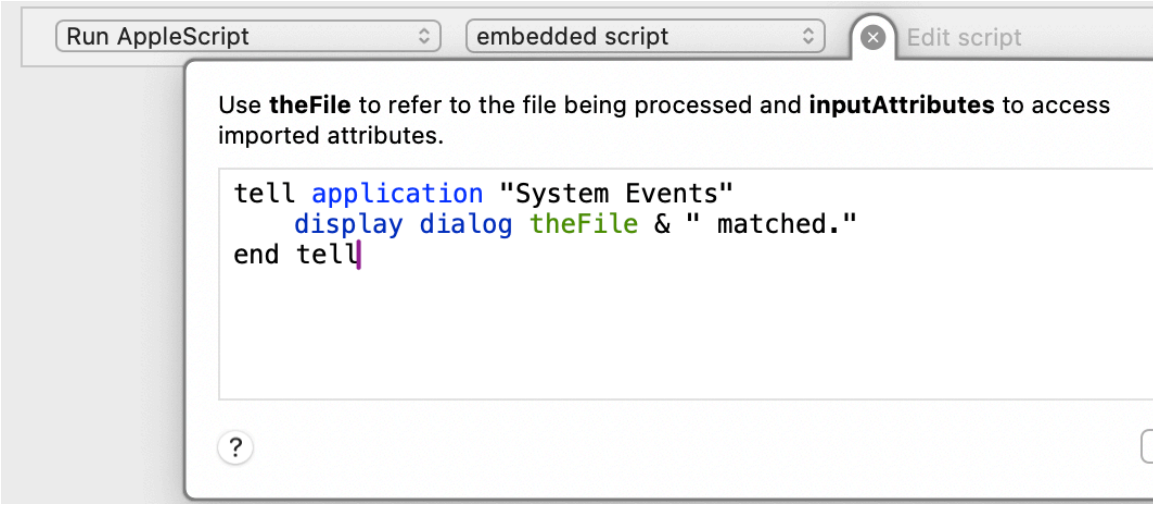
- **AppleScript/JavaScript in conditions:** AppleScript or JavaScript can also enable you to construct even more elaborate logical tests than Hazel natively offers with nested conditions. In addition, although Hazel can examine almost any attribute of a file or folder inside the watched folder, it can’t check the state of something *elsewhere* on your Mac—but AppleScript or JavaScript can. For example, you could write a script that checks for conditions such as the following:
    - The number of messages in my Inbox is greater than 100.
    - I have a certain URL open in a Safari tab.
    - The amount of free space on my disk is less than 10 GB.
    - I’m connected to the library’s free Wi-Fi network.
- **AppleScript/JavaScript in actions:** Similarly, while Hazel can move, copy, rename, and tag files and folders (among other actions), there are many tasks you might want to accomplish that Hazel can’t perform without a bit of help (in this case, from AppleScript or JavaScript). For example:
    - Sending an email message
    - Inserting a value into a specific cell in an Excel or Numbers spreadsheet
    - Opening a webpage
    - Performing mathematical calculations

Embedded vs. External AppleScripts/JavaScripts

Hazel can run AppleScripts or JavaScripts in either of two ways:

- **Embedded:** You can type (or paste) your script directly into your Hazel rule. To

do this, choose “Passes AppleScript”/“Passes JavaScript” as the attribute for a condition, or “Run AppleScript”/“Run JavaScript” for an action, and then click the Edit Script icon. Enter your script in the popover that appears (as illustrated below), optionally click the Build button to verify your syntax, and click the X icon when you’re done. Embedded scripts are stored with your rules and therefore do not need to be transferred separately when copying, moving, sharing, or syncing your rules.



- **External:** If your AppleScript or JavaScript already exists as a standalone script (`.scpt`) file, you can simply point Hazel to that script. To do this, choose “Passes AppleScript”/“Passes JavaScript” as the attribute for a condition, or “Run AppleScript”/“Run JavaScript” for an action, choose “Other” from the second pop-up menu, navigate to the file on disk, and click “Open.” Because external scripts are stored separately from Hazel rules, they must be transferred separately when copying, moving, sharing, or syncing rules to another Mac.

AppleScript/JavaScript in Rule Conditions

You can use AppleScript or JavaScript to supply custom logic for a rule condition. The primary requirement is that your script return `true` or `false` (as a value or a record/object—see “Returning Values from the Script” below).

Keep in mind that because rule conditions are evaluated frequently, you should make sure your script runs quickly. Time-consuming scripts will greatly decrease Hazel’s performance. In addition, the script must not modify any files or folders underneath the folder Hazel is monitoring, as that can result in a loop in which Hazel never stops processing the folder.

Passing Values to the Script

The variable `theFile` carries an alias to the file or folder currently being processed, which Hazel passes to the script.

For **embedded** scripts, no handler (AppleScript) or function (JavaScript) is needed. For **external** scripts, Hazel requires a handler (AppleScript) or function (JavaScript) called `hazelMatchFile`. The handler or function takes two arguments—`theFile` (as described just above) and `inputAttributes`, a list of input attribute values (see the section just ahead on .

Below is a template you can use for external AppleScripts:

```
on hazelMatchFile(theFile, inputAttributes)

    -- Add your logic here.

    -- 'theFile' is an alias to the file that matched.

    -- 'inputAttributes' is an AppleScript list of the values of
any attributes you told Hazel to pass in.

    -- Be sure to return true or false (or optionally a record)
to indicate whether the file passes this script.

end hazelMatchFile
```

And here is the corresponding JavaScript template:

```
function hazelMatchFile(theFile, inputAttributes)

{

    -- Add your logic here.

    -- 'theFile' is an alias to the file that matched.

    -- 'inputAttributes' is a JavaScript array of the values of
any attributes you told Hazel to pass in.

    -- Be sure to return true or false (or optionally an object)
to indicate whether the file passes this script

}
```

Returning Values from the Script

To tell Hazel whether the file or folder being processed matches your condition, have your script return `true` or `false`. For example:

```
-- your supporting code here
```

```
if (your logical test here)
    return true
else
    return false
end if
```

You can optionally return a *record* (AppleScript) or an *object* (JavaScript) instead of `true` or `false`. This method allows your script to send values back to Hazel. An AppleScript record looks something like this:

```
{key1:value1, key2:value2,…}
```

Currently, Hazel recognizes the following keys:

- **hazelPassesScript:** This is the same as returning `true` or `false` by itself: the `hazelPassesScript` key tells Hazel whether the file passes the script or not. For example, the following two AppleScript fragments do the same thing.

  Returning a true/false value (AppleScript):

  ```
  -- your supporting code here
  if (your logical test here)
      return true
  else
      return false
  end if
  ```

  Returning a record (AppleScript):

  ```
  -- your supporting code here
  if (your logical test here)
      return {hazelPassesScript:true}
  else
      return {hazelPassesScript:false}
  end if
  ```

- **hazelOutputAttributes:** A record (AppleScript) or object (JavaScript) of keys/values for the custom attributes this script exports. See the section below on Custom Attributes. Here's an example in AppleScript:

  ```
  -- your supporting code here
  return {hazelPassesScript:true, hazelOutputAttributes:
  { color:"blue", flavor:"raspberry"}}
  ```

AppleScript/JavaScript in Rule Actions

You can also use an AppleScript or JavaScript as the “Run AppleScript”/“Run JavaScript” action.

Passing Values to the Script

The variable `theFile` carries an alias to the file or folder that matched the rule, which Hazel passes to the script.

For **embedded** scripts, no handler (AppleScript) or function (JavaScript) is needed. For **external** scripts, Hazel requires a handler (AppleScript) or function (JavaScript) called `hazelProcessFile`. The handler or function takes two arguments—`theFile` (as described just above) and `inputAtttributes`, a list/array of input attribute values (see the section just ahead on).

Below is a template you can use for external AppleScripts:

```
on hazelProcessFile(theFile, inputAttributes)

    -- Add your logic here.

    -- 'theFile' is an alias to the file that matched.

    -- 'inputAttributes' is an AppleScript list of the values of any attributes you told Hazel to pass in.

    -- Be sure to throw errors if you do not want Hazel to consider this action to have run successfully.

end hazelProcessFile
```

And here is the corresponding JavaScript template:

```
function hazelProcessFile(theFile, inputAttributes)

{

    -- Add your logic here.

    -- 'theFile' is an alias to the file that matched.

    -- 'inputAttributes' is a JavaScript array of the values of any attributes you told Hazel to pass in.

    -- Be sure to throw errors if you do not want Hazel to consider this action to have run successfully.

}
```

Returning Values from the Script

Your script can optionally return a record (AppleScript) or object (JavaScript). (If your script returns anything besides a record/object, that value is ignored.) This method allows your script to send values back to Hazel. An AppleScript record looks something like this:

```
{key1:value1, key2:value2,…}
```

Currently, Hazel recognizes the following keys (of which your record/object can contain any or all):

- **hazelStop:** If set to true, instructs Hazel to stop processing. Subsequent actions for this rule do not run, and Hazel considers the rule as a whole to have been successful. You can use this to stop a rule from continuing based on what button a user clicks in a dialog, for example. Hazel will not try to run this rule again.
- **hazelSwitchFile:** If set to a valid file, Hazel applies subsequent actions to that file. You can use this to retarget what file Hazel is acting on, such as switching to the parent folder. Or, if the script transforms the file into a new one, you can point this at the new file.
- **hazelOutputAttributes**: A record of values for the custom attributes this script exports (just as in rule conditions). See the section below on Custom Attributes.

So, an AppleScript's `return` line may look something like this:

```
return {hazelStop:false, hazelSwitchFile:myFile,
hazelOutputAttributes:{ color:"blue", flavor:"raspberry"}}
```

Reporting Errors

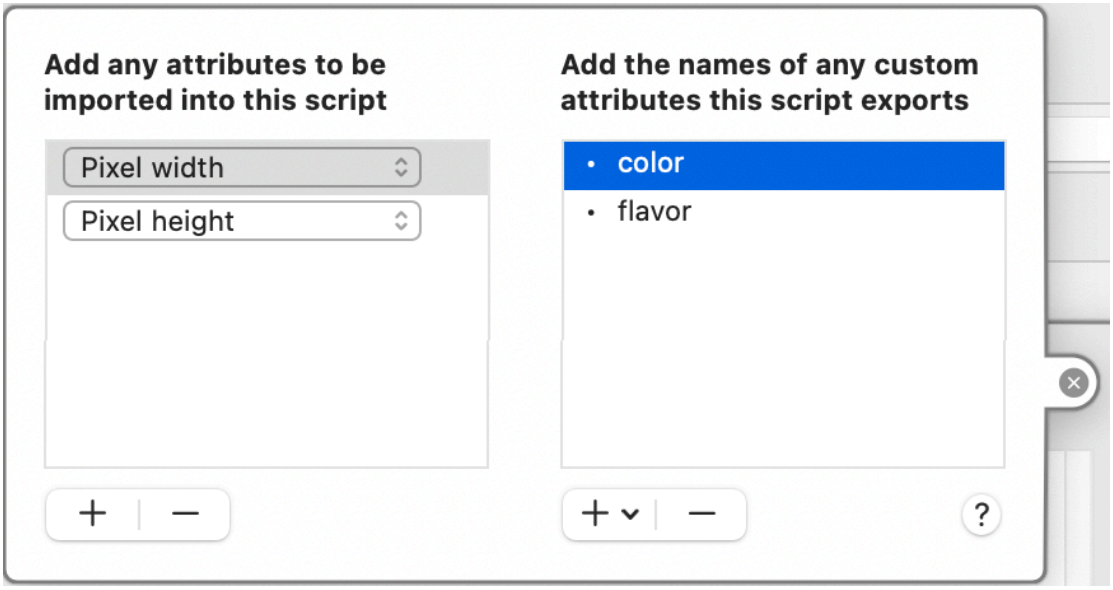
If you want the script to terminate **un**successfully, you must throw an *error* (AppleScript—see Apple's page Working with Errors in the AppleScript Language Guide) or an *exception* (JavaScript). If the script encounters an error/exception, Hazel considers the rule as a whole to have been unsuccessful, and will try to run the rule again later. If the script does not throw an error/exception, Hazel considers the script to have been successful regardless of what is returned.

Input Attributes

An AppleScript or JavaScript can take a list of input attribute values as its second argument. This enables you to pass most of Hazel's built-in attributes, as well as any

Spotlight attribute or any custom attribute you defined earlier in the rule, to your script for further processing.

To specify these attributes for an *external* script, click the Edit Script icon in the AppleScript/JavaScript condition or action. For an *embedded* script, first click the Edit Script icon, and then click the Custom Tokens icon in the script editor popover.



On the left side of this popover, you can specify which attributes to pass into your script. To add an attribute, click the plus + button and choose an attribute from the pop-up menu; repeat as needed to add more attributes. (The pop-up menu also includes any custom attributes that were defined earlier in the rule.) The right side (exporting attributes) is covered just ahead.

The second (`inputAttributes`) argument that Hazel passes to your script automatically is a list of attributes, in the order they were defined in the popover above. So in this example, the first item in the list, item 1 of `inputAttributes`, is the attribute "Pixel width," and the second item in the list, item 2 of `inputAttributes`, is the attribute "Pixel height."

Therefore, to use these attributes in your script, you can assign variables to the relevant list/array items. Here's how you might do it in AppleScript:

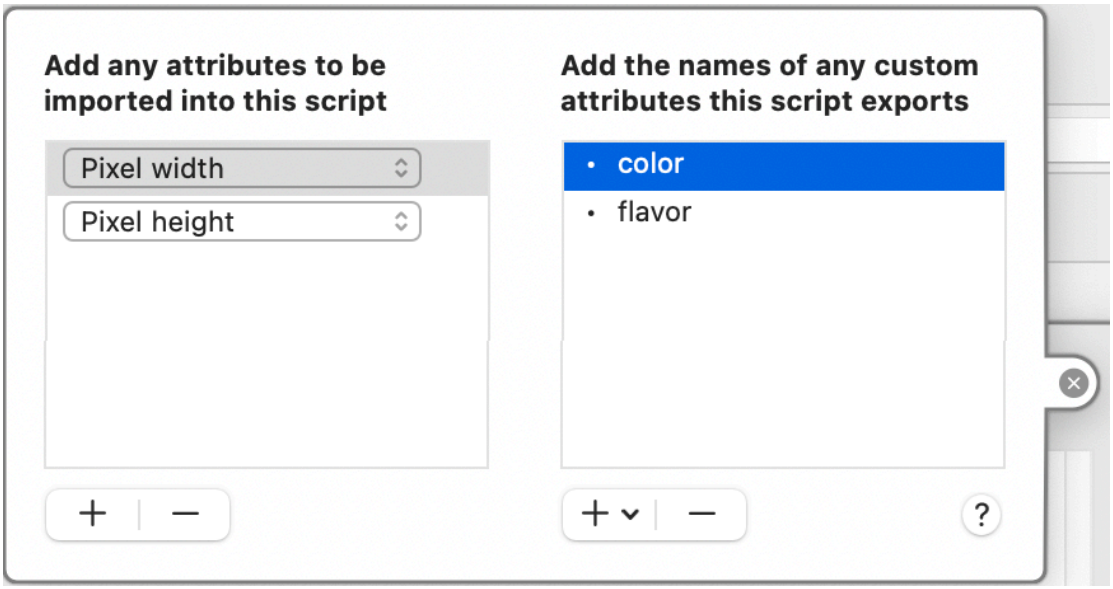
```
set width to item 1 of inputAttributes
set height to item 2 of inputAttributes

-- Use width and height as needed
if width is equal to height then
```

. . .

Custom Attributes

Just as you can pass attributes into a script, you can also export custom attributes defined by your script. Once you've done so, those custom attributes become available for patterns in conditions or actions that after the condition or action in which you exported them.

To make these attributes accessible to your rule, from an *external* script, click the Edit Script icon in the AppleScript/JavaScript condition or action. For an *embedded* script, first click the Edit Script icon, and then click the Custom Tokens icon in the script editor popover. (It's the same popover shown just above.)



On the right side of this popover, you can add the names of any custom attributes your script exports. To add an attribute, click the plus button and choose "New Text Attribute" or "New Date Attribute" from the pop-up menu. Select the newly added attribute, press Return, and type a name for it. Repeat the process as needed to add more attributes.

```
return {hazelOutputAttributes:{ color:"blue", flavor:"
raspberry"}}
```

These values will then be substituted wherever the attribute is used (in conditions or actions later in the rule).

SEE ALSO

Using Shortcuts

Using Automator

Using Shell Scripts

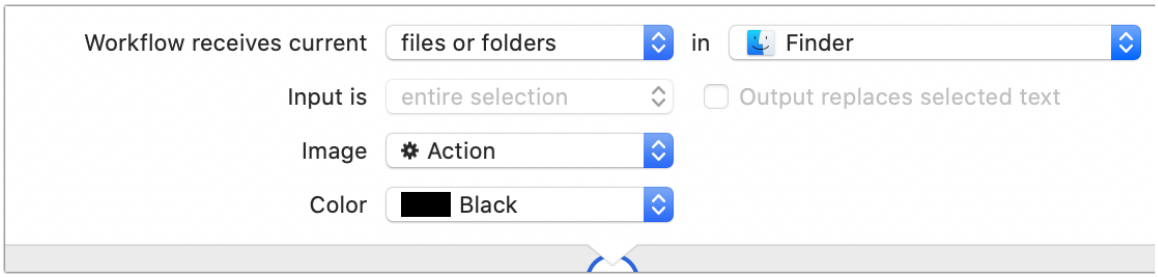
Using Automator

Hazel rules can use Automator to process matched files if you choose the “Run Automator workflow” action. (Unlike AppleScript, JavaScript, and shell scripts, Automator cannot be used in a rule *condition*, only in an action.) The file or folder that was matched is passed to the Automator workflow as input.

Using Automator to create actions enables you to extend Hazel’s built-in capabilities without having to write any code. Automator workflows can, for example, rotate or resize images, send files as attachments to email messages (see example below), make changes to your Contacts or Calendar data, or save text to an audio file—all things Hazel can’t do on its own.

To create an Automator workflow and use it in a Hazel rule:

1. Open Automator (in Macintosh HD ▸ Applications ▸ Utilities).
2. Click “New Document.” (Or, if Automator is already running, choose “File” > “New”.)
3. Select either Workflow or Quick Action as the document type and click “Choose.”
4. Build your workflow (see below for an example). If your document type is Quick Action, make sure the top of the workflow says “Workflow receives selected files or folders in Finder,” as shown below. (This option does not appear if your document type is Workflow, but that doesn’t prevent the workflow from accepting files and folders from the Finder.)

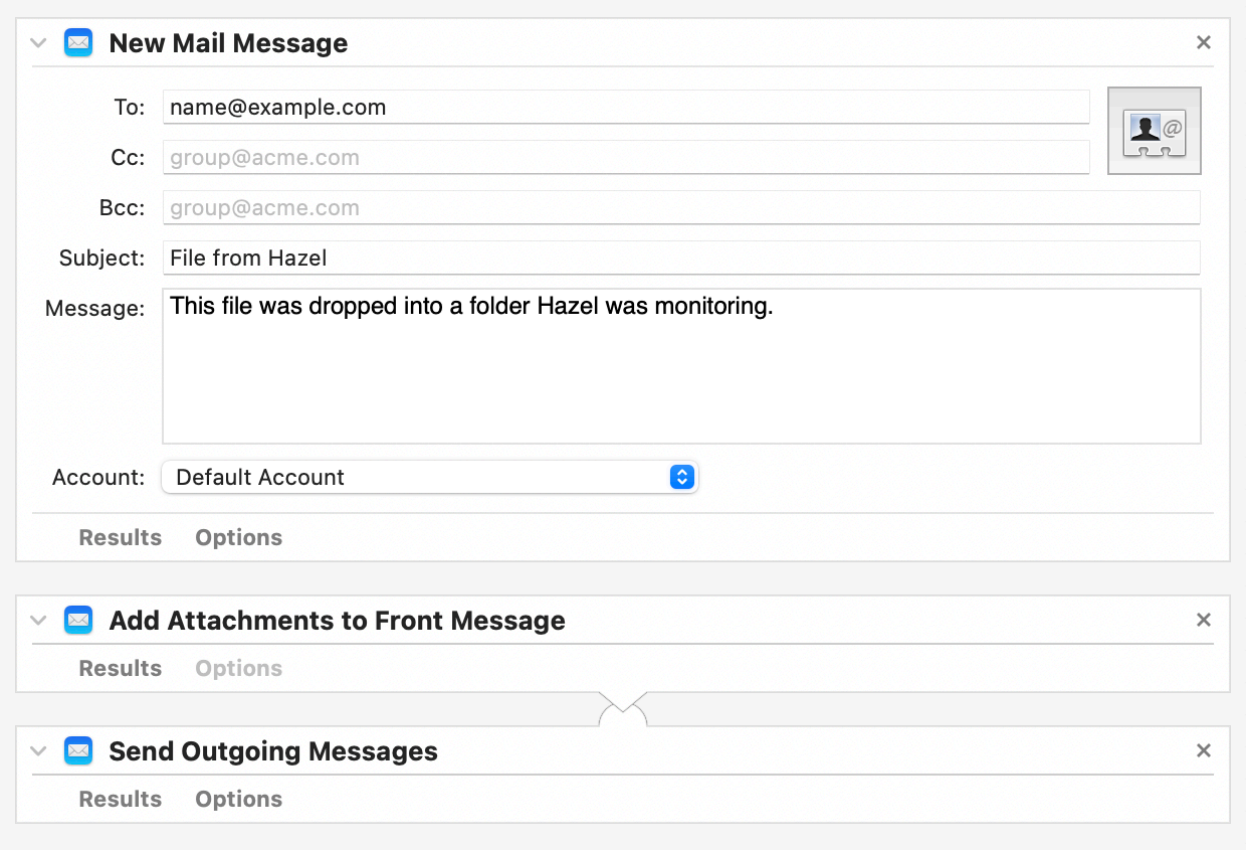


5. Choose File > “Save” to save your workflow. (If you selected Service as your document type, the workflow is saved to Macintosh HD ▸ Users ▸ your-username ▸ Library ▸ Services. Otherwise, navigate to the location you want to save the workflow and click “Save.”)
6. Now, in Hazel, open or create a rule, and choose “Run Automator workflow” as an action, choose “Other” from the second pop-up menu, navigate to the

workflow you saved in the previous step. Click “Open.”

7. Click “OK” to save the rule.

The following example workflow (shown here as Workflow document type, rather than Quick Action) sends an email message (with whatever address, subject, message, and account you fill in), with the processed file as an attachment:



SEE ALSO

Using Shortcuts

Using AppleScript or JavaScript

Using Shell Scripts

Using Shell Scripts

Hazel can use shell scripts, or even standalone command-line programs, in either conditions or actions. In a condition, the shell script tells Hazel whether or not the file or folder being processed matches. In an action, the shell script can do pretty much anything (which may involve the file or folder being processed, but doesn't have to). Scripts can be written in any language (including Perl or Python), as long as you tell Hazel which shell or interpreter to use.

Like Shortcuts, AppleScript, JavaScript, and Automator, shell scripts enable Hazel to do things it's unable to do on its own. For example, a shell script in a condition could employ the `grep` command to perform sophisticated pattern-matching on the file using regular expressions. A shell script in an action, likewise, can process the matched file or folder using any of innumerable command-line utilities. If you're already comfortable writing shell scripts, using them in Hazel should be quite natural.

Unlike AppleScript and JavaScript, shell scripts in Hazel can neither accept import attributes nor export custom attributes. Shell scripts take just one argument, `$1`, which is the full path of the file or folder being processed.

When working with shell scripts in Hazel, keep the following in mind:

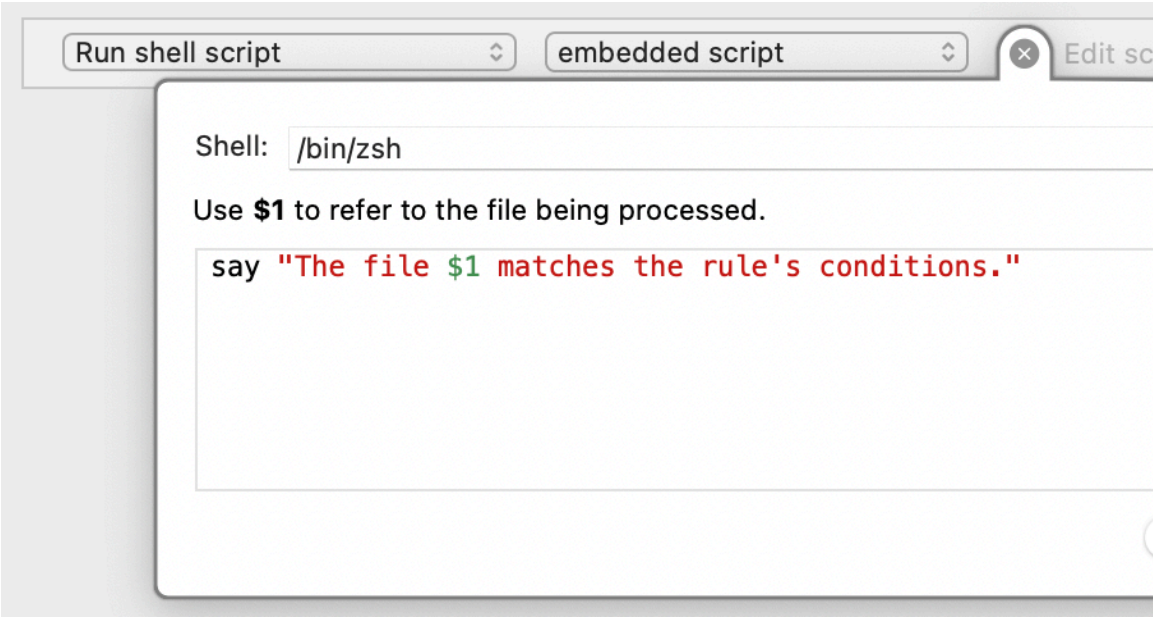
- The scripts execute in a different environment than in Terminal, which means that certain environment variables (such as a custom PATH and aliases) may not be set. Therefore, you should always specify the *full path* to any program or script, even though you might not need to do so in Terminal.
- Along the same lines, keep in mind that `$1` returns the *full* path, not a relative path, and that the path may contain spaces (which can cause errors in many commands if not properly accounted for). To prevent problems in the sh family of shells (zsh, tcsh, bash, etc.), enclose the path (or at least the `$1`) in quotation marks.

Embedded vs. External Shell Scripts

Hazel can run shell scripts in either of two ways:

- **Embedded:** You can type (or paste) your script directly into your Hazel rule. To do this, choose "Passes shell script" as the attribute for a condition, or "Run shell script" for an action, and then click the Edit Script icon. Enter your script in the popover that appears (as illustrated below), being sure to specify the shell or interpreter in the field at the top (the default is `/bin/zsh`), and click

the X icon when you're done. Embedded scripts are stored with your rules and therefore do not need to be transferred separately when copying, moving, sharing, or syncing your rules.



- **External:** If your script (or other command-line program) already exists as a standalone file, you can simply point Hazel to that file. To do this, choose "Passes shell script" as the attribute for a condition, or "Run shell script" for an action, choose "Other" from the second pop-up menu, navigate to the file on disk, and click "Open." As is the case in Terminal, the file must be executable. (That is, for user-created scripts, you should already have run `chmod ugo+x script_name`.) In addition, scripts must begin with a shebang (`#!`) line to indicate which interpreter to use—for example, `#!/bin/bash` or `#!/usr/bin/perl`. Because external scripts are stored separately from Hazel rules, they must also be transferred separately when copying, moving, sharing, or syncing rules to another Mac.

Shell Scripts in Rule Conditions

The goal of any script used in a rule condition (via the "Passes shell script" attribute) should be to perform a logical test that either succeeds (the condition matches) or fails (the condition does not match). Bear in mind the following:

- Scripts used for rule conditions should return an exit status of 0 to indicate that the file or folder being processed has matched the condition. (Normally, any

script or command-line program returns an exit status of 0 when it runs without errors.) Any other exit status tells Hazel that the file or folder being processed does not match the condition. This page provides various exit status examples for bash scripts; these should also work with zsh.

- Because rule conditions tend to be evaluated many times, you should make sure the scripts runs quickly. Time-consuming scripts will greatly decrease Hazel's performance.
- The script must not modify any files or folders underneath the folder Hazel is monitoring, as that can result in a loop in which Hazel never stops processing the folder.

Shell Scripts in Rule Actions

When you use the "Run shell script" action, a successful match can cause any script to run—even one that has nothing to do with the file or folder you're processing. However, most of the time you will probably want to use the shell script to perform an action on the matched file or folder, in which case your script can refer to it with the argument `$1`, which represents its path.

For example, this simple script uses text-to-speech to say the entire path of whatever file or folder has matched the rule condition(s):

```
#!/bin/zsh

say "The file $1 matches this rule's conditions."
```

As is the case when using shell scripts in conditions, scripts used in the "Run shell script" action should return an exit status of 0 to indicate that the action was performed successfully. Any other value indicates an error, which causes Hazel to execute the rule again repeatedly until the action succeeds. (Once again, see this page for exit status examples for bash or zsh scripts.)

SEE ALSO

Using Shortcuts

Using AppleScript or JavaScript

Using Automator

Advanced Topics

The following topics are for more advanced Hazel users:

- Processing Subfolders: Learn how to make Hazel descend into subfolders of the folder being monitored.
- Using Nested Conditions: Build complex logical tests in which multiple conditions are nested inside one another.
- Syncing Folders: Understand the uses and limitations of Hazel's "Sync" action.

Processing Subfolders

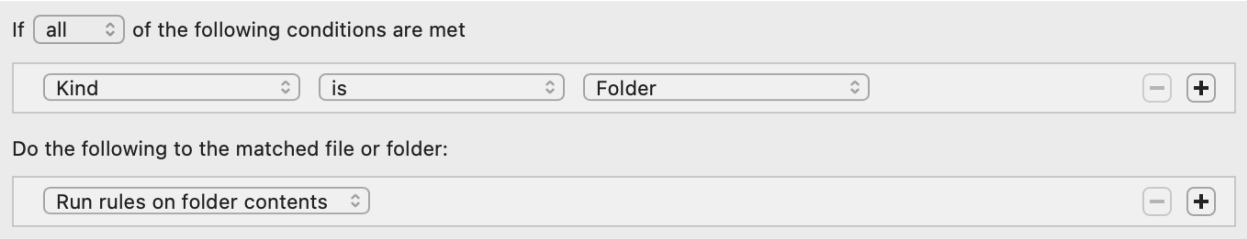
When Hazel checks your rules against the item in a monitored folder, it normally treats files and folders the same—for example, a Move action moves either a file or a folder if it matches your conditions, and a Rename action renames either a file or folder if it matches.

As a result, Hazel normally ignores specific changes inside subfolders of the monitored folder. So, suppose you're monitoring your Downloads folder, and inside that is a folder called Research. If you have a rule that matches any PDF document and opens it in Preview—and you download a PDF to your Downloads folder, that file will open. But if you download a PDF directly to the Research subfolder, the rule won't match, because Hazel ignores what goes on inside subfolders, unless you expressly tell it to examine them.

If you want Hazel rules to process the subfolders within a monitored folder, you can of course add those subfolders as monitored folders and create special rules just for them. But that's not always practical—especially if you don't know in advance what the subfolders' names will be, or if there are quite a few of them and you want the same rules to apply to all of them.

To solve this problem, Hazel offers a special action: "Run rules on folder contents." If a subfolder inside your monitored folder matches a rule containing this action, then the other rules in the list will also apply to that subfolder's contents.

The easiest way to make a "Go into subfolders" rule is to configure it as follows. (For your convenience, you can download this rule here: subfolder.hazelrules. After you've downloaded the rule, open Hazel, select the folder you want to monitor, and drag the rule into its Rules list.)



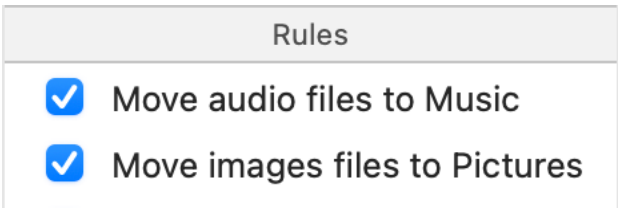
This rule has just one condition (it matches any folder) and one action ("Run rules on folder contents").

In many situations, this can be the whole rule. If it looks as though this rule doesn't actually perform any actions, that's because it doesn't need to. Instead, it effectively modifies the behavior of your other rules, making their actions apply to items in subfolders. Therefore, it's usually best to write all your rules as though they were

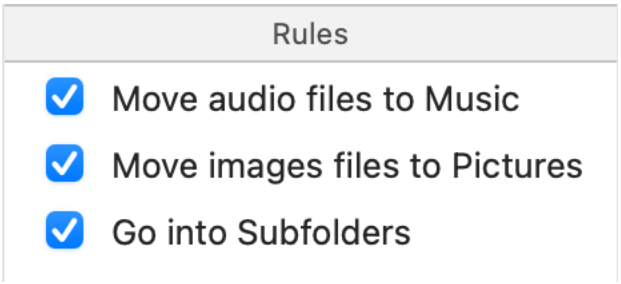
operating on items at the top level of your monitored folder, and test them to make sure they function as intended. Then add the “Go into subfolders” rule to extend their behavior to items in subfolders of the monitored folder as well as the top level of the monitored folder.

To demonstrate why and how you might use the “Run rules on folder contents” action, let’s walk through a series of examples.

Imagine Hazel is monitoring a folder in which you often place music and image files. You initially have two rules: one that moves audio files to your Music folder and another that moves image files to your Pictures folder. Your rules might look like this:

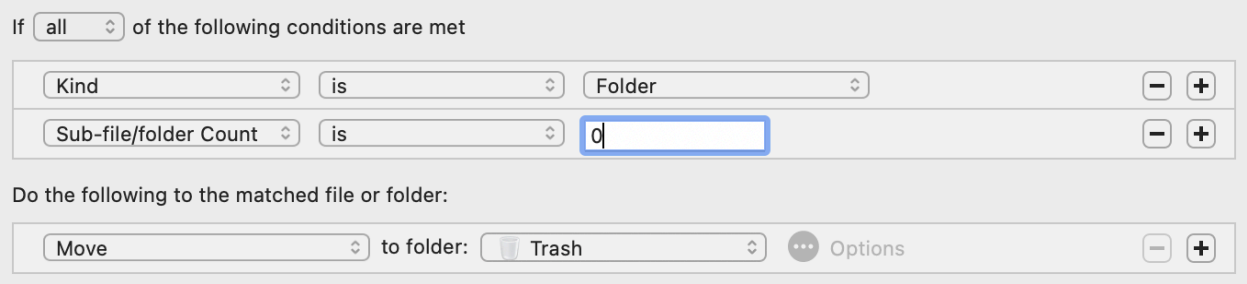


These rules work fine for individual files, but what if you drop a folder full of images or audio files into your monitored folder? Nothing would happen, because by default, Hazel matches only top-level items in the monitored folder, and the folder containing images or audio files does not, itself, match your rules. So you can add a third rule—the “Go into subfolders” rule described above:

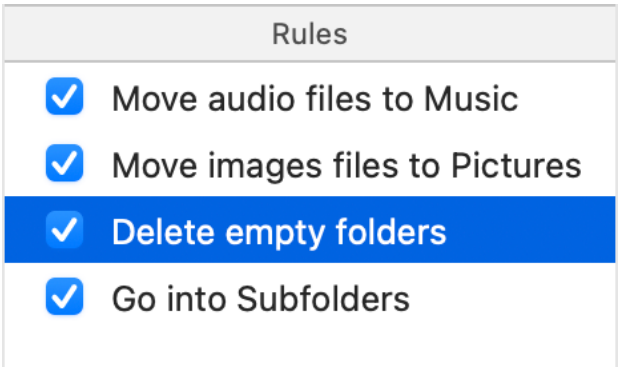


This new rule makes the other rules for this folder apply to subfolders as well. So now if you drag a folder of photos into your monitored folder, they’re all moved to Pictures, and if you drag a folder of MP3s into your monitored folder, they’re all moved to Music. Your existing rules did not have to change at all; they merely had their scope of operation expanded.

So far so good, but now you may find that your monitored folder contains empty subfolders! That’s because Hazel moved their contents (photos or audio files) to other locations. If you want to delete the empty subfolders, you can add a fourth rule—call it “Delete empty folders.” It would look like this:

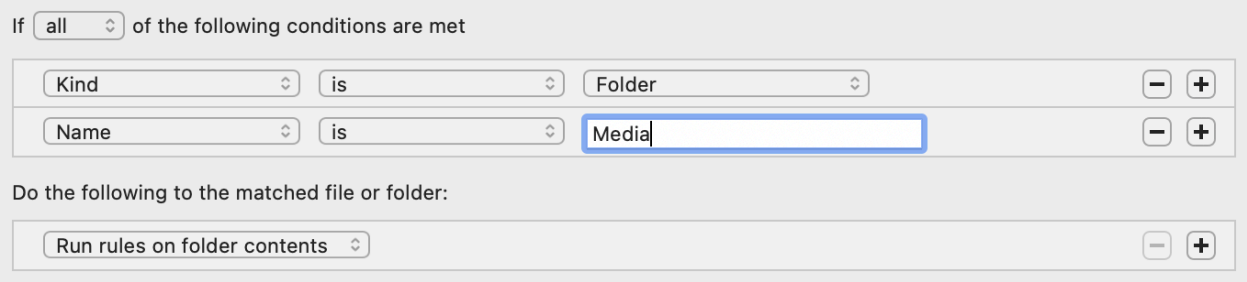


But wait! This new rule matches folders (because of the condition “Kind is Folder”) and so does the “Go into subfolders” rule. By default, Hazel applies only the first matching rule in the list to any given item (see Understand the Logic of Rules), so if “Go into subfolders” comes before “Delete empty folders,” then the “Delete empty folders” rule will never run, because all folders will already have been matched. So, since this new rule is more specific (it applies only to certain folders, not all folders), we put it before “Go into subfolders,” like so:



Let’s walk through how this will work. Suppose you drag a folder called “My Tunes” into your monitored folder. The presence of the “Go into subfolders” rule in the list means that each of the other rules will apply to folders within the monitored folder, so the “Move audio files to Music” rule runs and moves the files out of the “My Tunes” folder. Then the “Delete empty folders” rule matches the now-empty “My Tunes” folder and moves it to the Trash. But if the “Delete empty folders” rule had come after “Go into subfolders,” it would never have had a chance to run, because the “Go into subfolders” rule itself would already have matched that folder. (Another way to achieve the desired result would be to keep the “Go into subfolders” rule first, but add the “Continue matching rules” action to it, so that a match won’t block later rules from applying as well.)

To take this example even further, suppose we want only some folders in our monitored folder to be processed this way. For example, you want the first two rules to operate only on files inside folders whose names start with the word “Media.” To do this, you could modify the “Go into subfolders” rule by adding a second condition that specifies the folder name:



Once you've made that change, dragging a "Media Collection" folder into the monitored folder will trigger the rule, whereas dragging a "My photos and music" folder into the monitored folder will not.

Processing subfolders can get tricky, so here are a few tips you should keep in mind:

- Keep your "Go into subfolders" rule separate from your other rules—there's no need to combine everything into a single rule, and trying to do so is likely to produce undesirable results.
- The "Run rules on folder contents" action will fail if it acts on anything other than a folder. Be sure you use a Kind is Folder condition (as shown above) to ensure that non-folder items do not match. And don't add conditions or actions to your "Go into subfolders" rule that refer to files—leave those for other rules. (If the rule is matching a file and doing something with it, then it can't match a folder as well.)
- Running rules on folder contents is recursive—that is, Hazel will keep going into subfolders of subfolders of subfolders (and so on) unless you tell it not to. If you want a specific rule (that is, a rule that comes after "Go into subfolders" in your list) to stop at a certain subfolder depth, add a condition to that rule with the attribute "Subfolder depth"—for example, "Subfolder depth is greater than 2." (In this context, a depth of 0 means the top level of the monitored folder. An item at the top of a subfolder within the monitored folder will have depth 1, and so on.) If you want to be sure all your rules stop at a certain subfolder depth, add the "Subfolder depth" condition to the "Go into subfolders" rule above.
- Hazel runs rules on folder contents in order, from the top down. That is, if it matches Folder A at the top level of your monitored folder, it then works its way through everything in Folder A, and if one of those items is Folder B (which also matches), it then goes through everything in Folder B and so on. When it's finished with Folder B's contents, it pops back up to Folder A and finishes checking its remaining contents, and when it's finished with those, it pops up another level to finish checking whatever's left at the top level of the monitored folder.
- If the monitored folder has a very deep hierarchy of subfolders, processing them

repeatedly can take a long time and can negatively affect Hazel's performance. For best results, use this approach on relatively shallow folder hierarchies, or use the "Subfolder depth" attribute to limit how deep the rules go.

- As a reminder, a good rule of thumb is to first create rules that will process files as if they were at the top level of the monitored folder, and then add the rule above to tell Hazel to go into subfolders.

SEE ALSO

Attribute Reference

Action Reference

Understand the Logic of Rules

Using Nested Conditions

Using Nested Conditions

Most rules work just fine when you use “all” to match all the conditions, “any” to match at least one condition, or “none” to match no conditions—for example, “All of (Name contains Apple, Extension is .pages, and Date Created is Today)” or “Any of (Size is greater than 1 GB, Date Last Opened is not in the last 5 years, or Color Label is Red).” But what if you need more elaborate combinations, like “All of (Any of (Kind is PDF or Kind is Image)) and (Any of (Name contains Screenshot or Tags do not contain Ignore))”?

Hazel has just the solution for such cases: *nested conditions*, in which a single condition can contain one or more additional conditions (and so on as deep as necessary), with “all”, “any,” or “none” specified at each step. This capability lets you construct highly specific conditions that amount to various combinations of AND and OR operations. For example:

- “Name is A *or* Name is B, *and* the item is also a music file.”
- “Size is over 1 GB, *and* either the file is a move *or* it has a Red tag.”

Nested conditions come with an additional feature that gives them even more power—*condition targets*. For each level of nesting (even if you use only a single nested condition, so there’s effectively no hierarchy at all), Hazel provides a pop-up menu that lets you choose the scope to which that condition applies. Although that scope can simply be the file or folder that is currently being processed (the default), you can instead choose to examine subfiles and subfolders, or the enclosing folder, of the current item. For example:

- You can match an item based on properties of its parent folder, as in “Move this file only if it has the extension `.jpg` *and* it’s inside a folder called Pictures.”
- You can match a folder based on the properties of files it contains, as in “Rename this folder only if a file inside it was created within the last 10 minutes.”

Create Nested Conditions

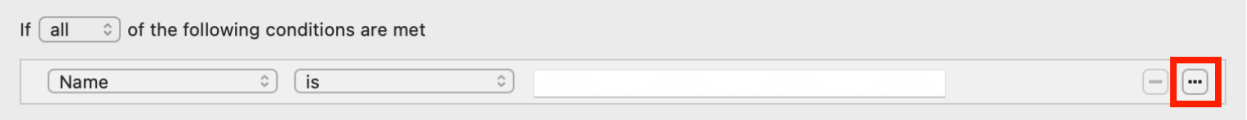
To create a nested condition, do either of the following:

- Hold down the Option key. The plus ⊕ button at the right of each condition row changes to an ellipsis ⋯. Click an ellipsis ⋯ button to add a nested condition below it.
- Hold down the plus ⊕ button for the condition immediately above where you

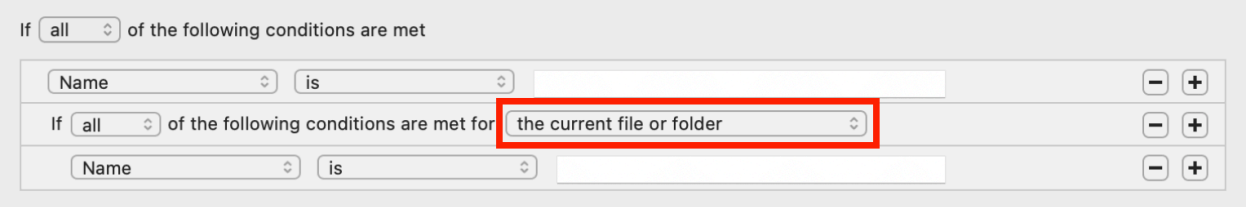
the nested condition to appear. After a moment, a pop-up menu appears; choose “Add nested condition” from this menu.

In either case, two new rows appear below that row. The first has its own “any/all/none” pop-up menu plus a second menu for condition targets, and the second is a blank condition that you can fill in as needed.

In other words, if you start with this (in this image, the Option key is being pressed) and you click the ellipsis ··· button at the right of the sole condition:



The nested condition then looks like the following (the condition target pop-up menu is outlined in red):



You can repeat this process as needed to create further nesting, or add new conditions at the same level by clicking the plus + button as usual.

Like any other condition, nested conditions can be dragged up or down, or deleted. You can even drag items in and out of nested conditions.

Condition Targets

Nested conditions have a second pop-up menu that specifies the condition target—in other words, what the condition should apply to. The available targets are as follows:

- **“the current file or folder”:** This is the default, and the one you will probably want most often. This refers to the file or folder referenced by the outer condition. If the outer condition is the top-level condition (that is, you have only one level of nesting), the conditions under this heading match the file or folder currently being processed. However, if the outer condition (a condition nested above this one) is set to “any of its files or subfolders,” then the conditions under this heading apply to those subfiles as well.
- **“its enclosing folder”:** This target matches the folder that contains the file or folder being processed. This can be used, for example, to perform actions on a

file based on whether its parent has a certain name.

- **“any file or folder in the same folder”:** Use this to apply the conditions to other items in the same folder as the current one. As long as one of the files matches the conditions under this one, this condition will match. This is useful in cases where files in the same folder are related, such as if you need to match a text file if there is a corresponding image file.
- **“all files and folders in the same folder”:** This is like the previous target in that it applies to other items in the same folder, but this time, it requires that *all* the other files match the conditions under this one. You can use this if, for example, you need to match a file that is a manifest/index for the rest of the files in the folder.
- **“any of its subfiles or subfolders”:** This target applies only to folders. It will not match if applied to a file. With this target set, the condition matches the folder being processed if at least one subfile or subfolder within it matches the conditions underneath it. For example, you can use this target to match a folder if any of its subfiles have an “Orange” tag.
- **“all of its subfiles and subfolders”:** This target is like the previous one in that it applies only to folders, and no individual file will trigger a match. With this target set, the condition matches the folder being processed if *every* subfile or subfolder within it matches the subconditions. For example, you can use this target to perform an action on a folder if all its subfiles are older than a certain age.

Keep in mind that regardless of your condition target, the rule’s *actions* apply to the file or folder currently being processed. So, even if the rules *conditions* are based on the subfiles, subfolders, or enclosing folder of the item being processed, the *actions* apply to the item itself. (If you need to apply actions to a folder's subfiles, see
for more information.)

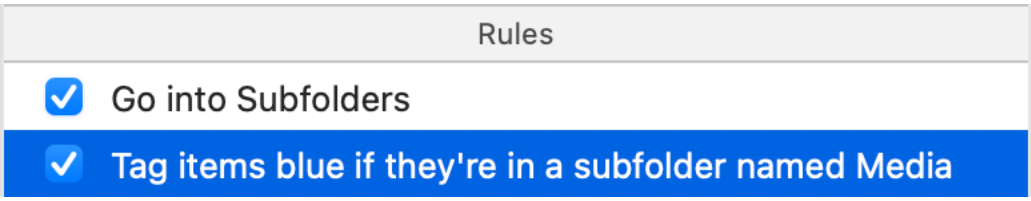
Condition Target Examples

Here are example rules that illustrate two of the capabilities of condition targets—matching an item based on properties of its parent folder, and matching a folder based on properties of the items it contains. In both cases, the rules contain only one nested condition, which exists simply to expose the condition target pop-up menu.

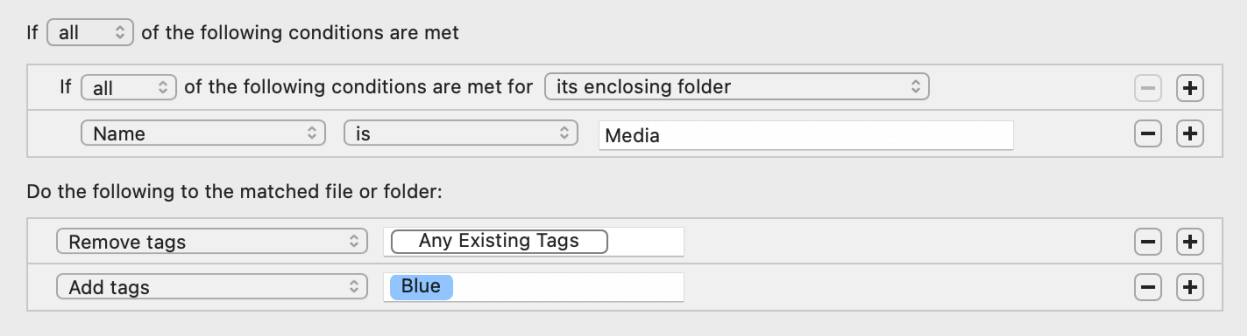
Matching Based on Parent Folder Properties

This example applies a Blue tag to any item whose parent folder is named Media.

Because the actions must apply to items in a subfolder, this rule must appear *after* a "Go into subfolders" rule that uses the "Run rules on folder contents" action (see for details):



The rule itself looks like this:

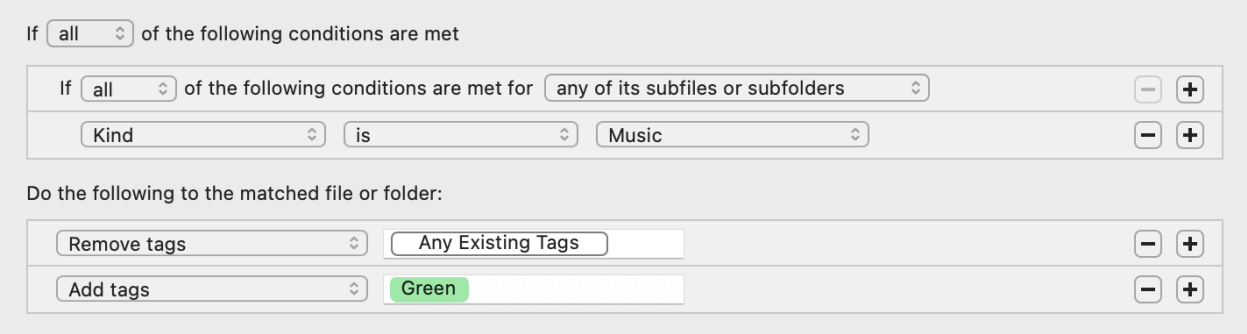


It reads as follows: If the single condition is met for an item's parent (enclosing) folder—the folder name is "Media"—then remove any existing tags from that item (not the Media folder) and apply the "Blue" tag (so that it will have *only* the "Blue" tag).

For another example of matching an item based on the properties of its parent folder, see "Example 2: Matching a Filename with the Name of Its Parent Folder" in Using Custom Attributes.

Matching a Folder Based on Its Contents

This example applies a "Green" tag to any folder that *contains* a music file. Because the actions apply to the folder itself, and not to its contents, it's unnecessary to use a "Go into subfolders" rule. The rule looks like this:



It reads as follows: If the single condition is met for a folder's contents—there is at least

one music subfile—then remove any existing tags from the folder and apply the "Green" tag (so that it will have *only* the "Green" tag).

SEE ALSO

Understand the Logic of Rules

Create a Rule

Using Custom Attributes

Processing Subfolders

Syncing Folders

Hazel's Sync action copies items from the monitored folder to another folder—but unlike the Copy action, Sync copies only those items that have been added since the last time the rule ran. So you can think of it as being like a smarter or more efficient Copy.

Syncing items in Hazel differs from how some utilities use the word "sync." You'll get better results with the Sync action if you understand its limitations. Specifically:

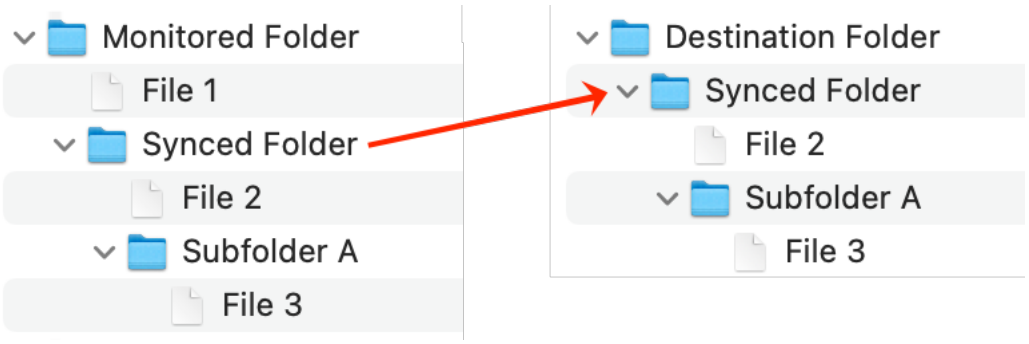
- **One-way syncing only:** Syncing in Hazel is one-way; files changed in the destination folder are *not* copied back to the monitored folder. Although you could in theory create mutual syncing rules for both source and destination folders, we recommend against doing this. Hazel was not designed for two-way sync, and it does not have the capability to handle conflicts and other complex situations the way a dedicated sync app such as ChronoSync can, nor will its syncing performance be anywhere close to that of a dedicated app.
- **Deletions are not synced (by default):** If a rule syncs files in the monitored folder with another folder, and then you delete an already-synced file from the monitored folder, that file will *not* disappear from the destination folder. That's because Hazel's rules normally operate only on files and folders that match your conditions, and the absence of a file that previously existed in the monitored folder is not something a condition can match.

  You can work around this limitation by using a rule that looks for any change in a *subfolder* of the monitored folder. This works because, from Hazel's point of view, the list of files in a folder is a property of that folder. Once the subfolder matches your condition(s) and syncs, its entire list of files is processed. And, as a result, any files deleted from the source will also disappear from the destination. However, a side-effect of this approach (which may or may not be to your liking) is that the destination folder must contain a copy of the entire subfolder being synced—not just its contents—as described below.

Sync a Subfolder (including Deletions)

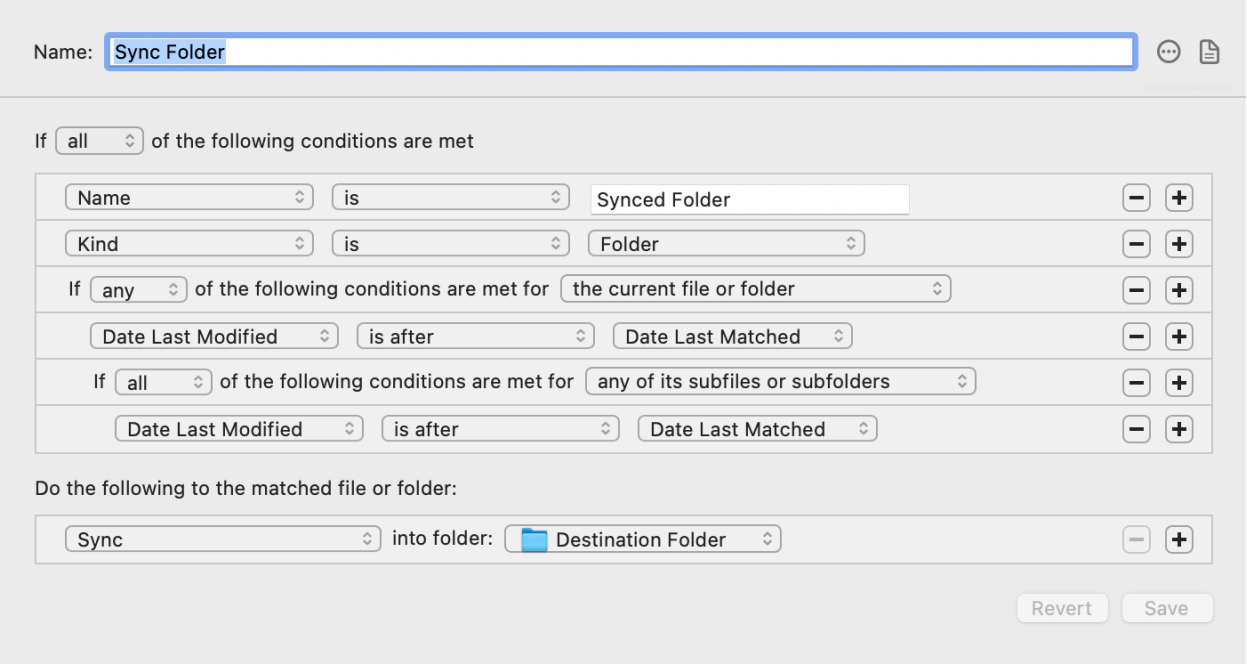
Here's how you can set up Hazel to sync an entire subfolder of a monitored folder, including deletions. To be clear, if the monitored folder is "Monitored Folder" in the example below, the folder you want to sync—inside the monitored folder—is called "Synced Folder" and the destination is "Destination Folder," the result after syncing will

look like this:



That is, the contents of “Synced Folder” don’t sync directly to the top level of “Destination Folder,” but rather to subfolder “Synced Folder” within “Destination Folder.” Anything deleted from “Synced Folder” in the monitored folder is deleted from the “Synced Folder” at the destination. Syncing is only one-way; changes in the “Synced Folder” subfolder of “Destination Folder” do not sync back to “Synced Folder” in the “Monitored Folder.”

The rule that performs this sort of sync looks like this:



You can create this rule manually (be sure to read Using Nested Conditions for help with the last four lines in the “conditions” section), or, if you prefer, you can download this rule using the following link: sync.hazelrules. After you’ve downloaded the rule, open Hazel, select the folder you want to monitor (that is, the folder that *contains* the folder you want to sync), and drag the rule into its “Rules” list. Then edit the rule and replace the name of the “Synced Folder” and the “Destination Folder” with the folders

you want to sync.

This rule looks for a folder inside the monitored folder with the name you specify. When Hazel detects changes in that folder—or in any of the subfiles or subfolders within it—the “Sync” rule runs, copying the contents of the synced folder to the destination folder.

Note that the conditions shown here merely tell Hazel *when* to sync (in this case, whenever the subfolder is modified). If you prefer, you can use entirely different conditions (such as Current Time is 12:00 AM any day) to specify a different schedule or frequency. Regardless of the conditions, the “Sync” action itself behaves the same way.

Tip: The conditions in this example rule are applicable to many actions, not just syncing. You can use the same conditions to trigger any action every time something is modified.

Because this rule runs every time “Synced Folder” or anything inside it changes, it can bog down your Mac a bit if that folder contains a large number of items. If you want to limit how often it runs, you can do so by adding “Date Last Matched is not in the last 10 minutes” (or however long you want the interval between runs to be) to the top of the conditions list.

SEE ALSO

Processing Subfolders

Using Nested Conditions

Action Reference

Copying Folder Structure

Specifying Upload Options

Viewing Logs

If your rules are not running as expected, you can view the logs to see what actually happened. Here you can see if Hazel encountered any errors when running your rules.

Note that it helps to preview your rules beforehand. See Preview a Rule for more details.

To view the logs, do either of the following:

- Choose Help > “View Logs.”
- If you enabled the Hazel status menu, choose “View Log” from that menu.

Either method above opens Hazel’s log file (Macintosh HD ▸ Users ▸ your-username ▸ Library ▸ Logs ▸ Hazel ▸ Hazel.log) in the Console utility (or in your default app for log files, if you changed it). You can use the search field to search for any folders or files of particular interest.

SEE ALSO

Stopping & Restarting Hazel

Preview a Rule

Enable, Disable, or Pause Rules

Run Rules Manually

Show Rule Status

Learn More

If you can’t find your answers here, then you can try the following:

- Search the .
- Participate in our where users come together to talk about Hazel.
- Peruse the page for articles written by others. (The more instructional articles are in the .)
- And of course, you can always

Credits

Updated icons for version 6 provided by Matthew Skiles (https://matthewskiles.com)

Numerous icons and layout work provided by Jono Hunt (https://jonohunt.com)

Parts of the user interface for Hazel 3 and Noodlesoft's website designed by Dave Keller at Made By Rocket

Some retina versions of graphics provided by John Marstall at Black Pixel

Original Hazel app icon design by Fernando Lins (http://www.devixdesign.com)

Documentation by Joe Kissell (https://www.takecontrolbooks.com)

Documentation created with Help Crafter by Putercraft

Hazel makes use of the following:

- AquaticPrime licensing framework ( )
- Sparkle software updating framework ( )
- PotionStoreFront in-app purchasing framework by Andy Kim (https://github.com/potionfactory/potionstorefront/tree/master)
- ConnectionKit by Karelia software (https://github.com/karelia/ConnectionKit)
- PSMTabBarControl by John Pannell
- unrar by Alexander L. Roshal (https://www.rarlab.com)
- rsync maintained by Wayne Davison (https://rsync.samba.org).

Except as noted otherwise, all documentation, the Noodlesoft website and the Hazel software itself:

Copyright © 2006–2025 Noodlesoft, LLC, All Rights Reserved

Notices

AquaticPrime

Copyright (c) 2005, Lucas Newman
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
- Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
  documentation and/or other materials provided with the distribution.
- Neither the name of the Aquatic nor the names of its contributors
  may be used to endorse or promote products derived from this
  software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED
TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,

INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

ConnectionKit

Licensed under the BSD License http://www.opensource.org/licenses/bsd-license
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE

OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

PSMTabBarControl

Copyright (c) 2005, Positive Spin Media
All rights reserved.
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

- Redistributions of source code must retain the above copyright
  notice,this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above
  copyright notice, this list of conditions and the following
  disclaimer in the documentation and/or other materials provided
  with the distribution.
- Neither the name of Positive Spin Media nor the names of its
  contributors may be used to endorse or promote products derived
  from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A

```
PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT,
INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED
AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.
```

PotionStoreFront

```
Copyright (c) 2011, Potion Factory LLC
All rights reserved.

Redistribution and use in source and binary forms, with or
without
modification, are permitted provided that the following
conditions are met:

    * Redistributions of source code must retain the above
copyright
      notice, this list of conditions and the following
disclaimer.
    * Redistributions in binary form must reproduce the
above copyright
      notice, this list of conditions and the following
disclaimer in the
      documentation and/or other materials provided with
```

the distribution.
    * Neither the name of the Potion Factory LLC nor the
      names of its contributors may be used to endorse or
promote products
      derived from this software without specific prior
written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY,
OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY
WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.

rsync

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://
fsf.org/>
Everyone is permitted to copy and distribute verbatim
copies

of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you
these rights or asking you to surrender the rights. Therefore, you have
certain responsibilities if you distribute copies of the software, or if
you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether
gratis or for a fee, you must pass on to the recipients the same
freedoms that you received. You must make sure that they, too, receive
or can get the source code. And you must show them these terms so they
know their rights.

Developers that use the GNU GPL protect your rights with two steps:
(1) assert copyright on the software, and (2) offer you this License
giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains
that there is no warranty for this free software. For both users' and
authors' sake, the GPL requires that modified versions be marked as
changed, so that their problems will not be attributed erroneously to
authors of previous versions.

Some devices are designed to deny users access to install

or run
modified versions of the software inside them, although the manufacturer
can do so. This is fundamentally incompatible with the aim of
protecting users' freedom to change the software. The systematic
pattern of such abuse occurs in the area of products for individuals to
use, which is precisely where it is most unacceptable. Therefore, we
have designed this version of the GPL to prohibit the practice for those
products. If such problems arise substantially in other domains, we
stand ready to extend this provision to those domains in future versions
of the GPL, as needed to protect the freedom of users.

  Finally, every program is threatened constantly by software patents.
States should not allow patents to restrict development and use of
software on general-purpose computers, but in those that do, we wish to
avoid the special danger that patents applied to a free program could
make it effectively proprietary. To prevent this, the GPL assures that
patents cannot be used to render the program non-free.

  The precise terms and conditions for copying, distribution and
modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making

available to the
public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other
parties to make or receive copies. Mere interaction with a user through
a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices"
to the extent that it includes a convenient and prominently visible
feature that (1) displays an appropriate copyright notice, and (2)
tells the user that there is no warranty for the work (except to the
extent that warranties are provided), that licensees may convey the
work under this License, and how to view a copy of this License. If
the interface presents a list of user commands or options, such as a
menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work
for making modifications to it. "Object code" means any non-source
form of a work.

A "Standard Interface" means an interface that either is an official

standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free

programs which are used unmodified in performing those activities but
which are not part of the work. For example, Corresponding Source
includes interface definition files associated with source files for
the work, and the source code for shared libraries and dynamically
linked subprograms that the work is specifically designed to require,
such as by intimate data communication or control flow between those
subprograms and other parts of the work.

  The Corresponding Source need not include anything that users
can regenerate automatically from other parts of the Corresponding
Source.

  The Corresponding Source for a work in source code form is that
same work.

  2. Basic Permissions.

  All rights granted under this License are granted for the term of
copyright on the Program, and are irrevocable provided the stated
conditions are met. This License explicitly affirms your unlimited
permission to run the unmodified Program. The output from running a
covered work is covered by this License only if the output, given its
content, constitutes a covered work. This License

acknowledges your
rights of fair use or other equivalent, as provided by
copyright law.

  You may make, run and propagate covered works that you do
not
convey, without conditions so long as your license
otherwise remains
in force. You may convey covered works to others for the
sole purpose
of having them make modifications exclusively for you, or
provide you
with facilities for running those works, provided that you
comply with
the terms of this License in conveying all material for
which you do
not control copyright. Those thus making or running the
covered works
for you must do so exclusively on your behalf, under your
direction
and control, on terms that prohibit them from making any
copies of
your copyrighted material outside their relationship with
you.

  Conveying under any other circumstances is permitted
solely under
the conditions stated below. Sublicensing is not allowed;
section 10
makes it unnecessary.

  3. Protecting Users' Legal Rights From Anti-Circumvention
Law.

  No covered work shall be deemed part of an effective
technological
measure under any applicable law fulfilling obligations

under article
11 of the WIPO copyright treaty adopted on 20 December 1996, or
similar laws prohibiting or restricting circumvention of such
measures.

When you convey a covered work, you waive any legal power to forbid
circumvention of technological measures to the extent such circumvention
is effected by exercising rights under this License with respect to
the covered work, and you disclaim any intention to limit operation or
modification of the work as a means of enforcing, against the work's
users, your or third parties' legal rights to forbid circumvention of
technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you
receive it, in any medium, provided that you conspicuously and
appropriately publish on each copy an appropriate copyright notice;
keep intact all notices stating that this License and any
non-permissive terms added in accord with section 7 apply to the code;
keep intact all notices of the absence of any warranty; and give all
recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that

you convey,
and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to
produce it from the Program, in the form of source code under the
terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified
it, and giving a relevant date.

b) The work must carry prominent notices stating that it is
released under this License and any conditions added under section
7. This requirement modifies the requirement in section 4 to
"keep intact all notices".

c) You must license the entire work, as a whole, under this
License to anyone who comes into possession of a copy. This
License will therefore apply, along with any applicable section 7
additional terms, to the whole of the work, and all its parts,
regardless of how they are packaged. This License gives no
permission to license the work in any other way, but it does not

invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product
(including a physical distribution medium), accompanied by the
Corresponding Source fixed on a durable physical medium
customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product
(including a physical distribution medium), accompanied by a
written offer, valid for at least three years and valid for as
long as you offer spare parts or customer support for that product
model, to give anyone who possesses the object code either (1) a
copy of the Corresponding Source for all the software in the
product that is covered by this License, on a durable physical
medium customarily used for software interchange, for a price no
more than your reasonable cost of physically performing this
conveying of source, or (2) access to copy the
Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the
written offer to provide the Corresponding Source. This
alternative is allowed only occasionally and noncommercially, and
only if you received the object code with such an

offer, in accord
with subsection 6b.

d) Convey the object code by offering access from a designated
place (gratis or for a charge), and offer equivalent access to the
Corresponding Source in the same way through the same place at no
further charge. You need not require recipients to copy the
Corresponding Source along with the object code. If the place to
copy the object code is a network server, the Corresponding Source
may be on a different server (operated by you or a third party)
that supports equivalent copying facilities, provided you maintain
clear directions next to the object code saying where to find the
Corresponding Source. Regardless of what server hosts the
Corresponding Source, you remain obligated to ensure that it is
available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided
you inform other peers where the object code and Corresponding
Source of the work are being offered to the general public at no
charge under subsection 6d.

A separable portion of the object code, whose source code

is excluded
from the Corresponding Source as a System Library, need not be
included in conveying the object code work.

  A "User Product" is either (1) a "consumer product", which means any
tangible personal property which is normally used for personal, family,
or household purposes, or (2) anything designed or sold for incorporation
into a dwelling. In determining whether a product is a consumer product,
doubtful cases shall be resolved in favor of coverage. For a particular
product received by a particular user, "normally used" refers to a
typical or common use of that class of product, regardless of the status
of the particular user or of the way in which the particular user
actually uses, or expects or is expected to use, the product. A product
is a consumer product regardless of whether the product has substantial
commercial, industrial or non-consumer uses, unless such uses represent
the only significant mode of use of the product.

  "Installation Information" for a User Product means any methods,
procedures, authorization keys, or other information required to install
and execute modified versions of a covered work in that User Product from
a modified version of its Corresponding Source. The information must

suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates

the rules and
protocols for communication across the network.

  Corresponding Source conveyed, and Installation Information provided,
in accord with this section must be in a format that is publicly
documented (and with an implementation available to the public in
source code form), and must require no special password or key for
unpacking, reading or copying.

  7. Additional Terms.

  "Additional permissions" are terms that supplement the terms of this
License by making exceptions from one or more of its conditions.
Additional permissions that are applicable to the entire Program shall
be treated as though they were included in this License, to the extent
that they are valid under applicable law. If additional permissions
apply only to part of the Program, that part may be used separately
under those permissions, but the entire Program remains governed by
this License without regard to the additional permissions.

  When you convey a copy of a covered work, you may at your option
remove any additional permissions from that copy, or from any part of
it. (Additional permissions may be written to require their own

removal in certain cases when you modify the work.) You may place
additional permissions on material, added by you to a covered work,
for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you
add to a covered work, you may (if authorized by the copyright holders of
that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the
terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or
author attributions in that material or in the Appropriate Legal
Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or
requiring that modified versions of such material be marked in
reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or
authors of the material; or

e) Declining to grant rights under trademark law for

use of some
    trade names, trademarks, or service marks; or

    f) Requiring indemnification of licensors and authors of that
    material by anyone who conveys the material (or modified versions of
    it) with contractual assumptions of liability to the recipient, for
    any liability that these contractual assumptions directly impose on
    those licensors and authors.

  All other non-permissive additional terms are considered "further
restrictions" within the meaning of section 10. If the Program as you
received it, or any part of it, contains a notice stating that it is
governed by this License along with a term that is a further
restriction, you may remove that term. If a license document contains
a further restriction but permits relicensing or conveying under this
License, you may add to a covered work material governed by the terms
of that license document, provided that the further restriction does
not survive such relicensing or conveying.

  If you add terms to a covered work in accord with this section, you
must place, in the relevant source files, a statement of the
additional terms that apply to those files, or a notice indicating

where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the
form of a separately written license, or stated as exceptions;
the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly
provided under this License. Any attempt otherwise to propagate or
modify it is void, and will automatically terminate your rights under
this License (including any patent licenses granted under the third
paragraph of section 11).

However, if you cease all violation of this License, then your
license from a particular copyright holder is reinstated (a)
provisionally, unless and until the copyright holder explicitly and
finally terminates your license, and (b) permanently, if the copyright
holder fails to notify you of the violation by some reasonable means
prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is
reinstated permanently if the copyright holder notifies you of the

violation by some reasonable means, this is the first time you have
received notice of violation of this License (for any work) from that
copyright holder, and you cure the violation prior to 30 days after
your receipt of the notice.

  Termination of your rights under this section does not terminate the
licenses of parties who have received copies or rights from you under
this License. If your rights have been terminated and not permanently
reinstated, you do not qualify to receive new licenses for the same
material under section 10.

  9. Acceptance Not Required for Having Copies.

  You are not required to accept this License in order to receive or
run a copy of the Program. Ancillary propagation of a covered work
occurring solely as a consequence of using peer-to-peer transmission
to receive a copy likewise does not require acceptance. However,
nothing other than this License grants you permission to propagate or
modify any covered work. These actions infringe copyright if you do
not accept this License. Therefore, by modifying or propagating a
covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation

(including a cross-claim or counterclaim in a lawsuit) alleging that
any patent claim is infringed by making, using, selling, offering for
sale, or importing the Program or any portion of it.

  11. Patents.

  A "contributor" is a copyright holder who authorizes use under this
License of the Program or a work on which the Program is based. The
work thus licensed is called the contributor's "contributor version".

  A contributor's "essential patent claims" are all patent claims
owned or controlled by the contributor, whether already acquired or
hereafter acquired, that would be infringed by some manner, permitted
by this License, of making, using, or selling its contributor version,
but do not include claims that would be infringed only as a
consequence of further modification of the contributor version. For
purposes of this definition, "control" includes the right to grant
patent sublicenses in a manner consistent with the requirements of
this License.

  Each contributor grants you a non-exclusive, worldwide, royalty-free
patent license under the contributor's essential patent claims, to
make, use, sell, offer for sale, import and otherwise run,

modify and
propagate the contents of its contributor version.

  In the following three paragraphs, a "patent license" is any express
agreement or commitment, however denominated, not to enforce a patent
(such as an express permission to practice a patent or covenant not to
sue for patent infringement). To "grant" such a patent license to a
party means to make such an agreement or commitment not to enforce a
patent against the party.

  If you convey a covered work, knowingly relying on a patent license,
and the Corresponding Source of the work is not available for anyone
to copy, free of charge and under the terms of this License, through a
publicly available network server or other readily accessible means,
then you must either (1) cause the Corresponding Source to be so
available, or (2) arrange to deprive yourself of the benefit of the
patent license for this particular work, or (3) arrange, in a manner
consistent with the requirements of this License, to extend the patent
license to downstream recipients. "Knowingly relying" means you have
actual knowledge that, but for the patent license, your conveying the
covered work in a country, or your recipient's use of the covered work

in a country, would infringe one or more identifiable patents in that
country that you have reason to believe are valid.

  If, pursuant to or in connection with a single transaction or
arrangement, you convey, or propagate by procuring conveyance of, a
covered work, and grant a patent license to some of the parties
receiving the covered work authorizing them to use, propagate, modify
or convey a specific copy of the covered work, then the patent license
you grant is automatically extended to all recipients of the covered
work and works based on it.

  A patent license is "discriminatory" if it does not include within
the scope of its coverage, prohibits the exercise of, or is
conditioned on the non-exercise of one or more of the rights that are
specifically granted under this License. You may not convey a covered
work if you are a party to an arrangement with a third party that is
in the business of distributing software, under which you make payment
to the third party based on the extent of your activity of conveying
the work, and under which the third party grants, to any of the
parties who would receive the covered work from you, a discriminatory
patent license (a) in connection with copies of the covered work

conveyed by you (or copies made from those copies), or (b) primarily
for and in connection with specific products or compilations that
contain the covered work, unless you entered into that arrangement,
or that patent license was granted, prior to 28 March 2007.

  Nothing in this License shall be construed as excluding or limiting
any implied license or other defenses to infringement that may
otherwise be available to you under applicable patent law.

  12. No Surrender of Others' Freedom.

  If conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License. If you cannot convey a
covered work so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you may
not convey it at all. For example, if you agree to terms that obligate you
to collect a royalty for further conveying from those to whom you convey
the Program, the only way you could satisfy both those terms and this
License would be to refrain entirely from conveying the Program.

  13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have
permission to link or combine any covered work with a work licensed
under version 3 of the GNU Affero General Public License into a single
combined work, and to convey the resulting work. The terms of this
License will continue to apply to the part which is the covered work,
but the special requirements of the GNU Affero General Public License,
section 13, concerning interaction through a network will apply to the
combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of
the GNU General Public License from time to time. Such new versions will
be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

Each version is given a distinguishing version number. If the
Program specifies that a certain numbered version of the GNU General
Public License "or any later version" applies to it, you have the
option of following the terms and conditions either of that numbered
version or of any later version published by the Free Software

Foundation. If the Program does not specify a version number of the
GNU General Public License, you may choose any version ever published
by the Free Software Foundation.

If the Program specifies that a proxy can decide which future
versions of the GNU General Public License can be used, that proxy's
public statement of acceptance of a version permanently authorizes you
to choose that version for the Program.

Later license versions may give you additional or different
permissions. However, no additional obligations are imposed on any
author or copyright holder as a result of your choosing to follow a
later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY
APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT
HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY
OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO,
THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM
IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU

ASSUME THE COST OF
ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS
THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY
GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE
USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF
DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD
PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS),
EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF
SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided
above cannot be given local legal effect according to their terms,
reviewing courts shall apply local law that most closely approximates
an absolute waiver of all civil liability in connection with the
Program, unless a warranty or assumption of liability accompanies a
copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied

```
warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General
Public License
    along with this program.  If not, see <http://
www.gnu.org/licenses/>.

Also add information on how to contact you by electronic
and paper mail.

  If the program does terminal interaction, make it output
a short
notice like this when it starts in an interactive mode:

    <program>  Copyright (C) <year>  <name of author>
    This program comes with ABSOLUTELY NO WARRANTY; for
details type `show w'.
    This is free software, and you are welcome to
redistribute it
    under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show
the appropriate
parts of the General Public License.  Of course, your
program's commands
might be different; for a GUI interface, you would use an
"about box".

  You should also get your employer (if you work as a
programmer) or school,
if any, to sign a "copyright disclaimer" for the program,
if necessary.
For more information on this, and how to apply and follow
```

the GNU GPL, see
<http://www.gnu.org/licenses/>.

  The GNU General Public License does not permit
incorporating your program
into proprietary programs. If your program is a subroutine
library, you
may consider it more useful to permit linking proprietary
applications with
the library. If this is what you want to do, use the GNU
Lesser General
Public License instead of this License. But first, please
read
<http://www.gnu.org/philosophy/why-not-lgpl.html>.