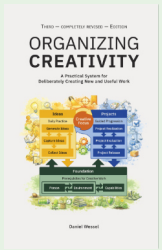
Backups

Mechanism

Imagine you wake up and your place is on fire — you have seconds to leave. Or you come home and find your place ransacked by thieves. Or one evening during a rather violent storm, a lightning strikes and you have an arc running across your room. In all these cases what you had — your idea collection, your works — are either damaged or gone. How would that affect you?

While ideas accumulate over time, the likelihood of survival of idea collections decreases over time. Fire, water, electric damages, data loss due to accident or user error, the cat spraying the competition, theft, normal failure or decay of storage media, even hacking can wipe out your digital data. Even paper is not protected, fire and water kill it as well. And if you store your data online (cloud, online service), you might lose access when the company folds, changes its terms of service, had an accident or was hacked itself.



Relevant Chapters

For background information, see Chapter 5: Environment and Chapter 9: Collecting Ideas.

All these nightmare scenarios are situations you can wake up from and realize it was only a bad dream — if you have backups. If these situations are not treated as catastrophes but as something you can actively plan for. To make your creative system resilient against data loss.

Applicability

This is relevant **if you have data you want to preserve.**

That's it.

If you have data you want to preserve, backups are a necessity.

While the applies primarily to the idea collection, it is also relevant for projects (archives, finished works). Ideas usually do not come back. Works that took weeks, months or longer to create are lost as well.

However, there are at least three major risk:

1. **Getting lost in making backups:** You can become paranoid about making backups, doing them immediately. That can undermine trust in your system and interfere with your work, when you assume nothing will stay stable for even a day. Use fixed dates (e.g., once per day) or specific moments for backups (e.g., after a major advancement).
2. **Backups actually destroying the work:** Normally backups put copies of your work in different places. But if you make a major error, you could overwrite your existing work. You would need a major error in a script, e.g., copying from the backup drive to your active drive or forgetting the encryption password, but it is possible. Computers do what you tell them to do, not what you want them to do.
3. **Putting your data in the wrong hands:** If you put your backups online or carry them with you (e.g., Box 2: Digital Lifeboat), it becomes more likely for the wrong people to get access to it. Even with encryption (see Box 4: Encryption). There is a trade-off here.

Intervention Variables

The practical aim of this is simple: define the data that matters, choose a maximum tolerable loss, create one backup routine, and restore-test it. The sections below give options for building that routine. The focus is on digital data, though paper should be backed up as well (see Backing up Paper).

Rule for Digital Files

As backups work on digital files, the following rules ensure that the backups have something to work with:

- **Save Early:** You cannot save too early. Ideally, you save a file immediately after creating it, which allows you to quick-save it via the Ctrl + S (or Cmd + S) shortcut (works with most programs).
- **Save Often:** You cannot save too often. If the file size is not too unwieldy for the computer, saving takes only an instant. Pressing Ctrl + S can be done while typing, without interrupting the work flow. It should become a habit that you hardly realize you do it. Even if you use the «auto-save» option available in many programs, make it a habit of saving manually immediately after you completed a significant step. You know when you have written something important that will not come back, the program merely counts down seconds.
- **Save Versions:** In rare instances, a file becomes corrupted. If you lose access to that file, the work is gone. So create multiple version of the same file. Use a standard naming scheme such as `FILENAME_YEAR_MONTH_DAY_CONSECUTIVENUMBER` , e.g., `projectX_2026-04-22_1.docx`, `projectX_2026-04-22_2.docx`, etc. Some apps do this automatically. Saving incrementally prevents you from losing all your work to a rare but very real and extremely volatile error. Whenever you have finished a major step, e.g., a good paragraph you would not get back, is a good moment. You can make saving incrementally easier if you display the folder listing of the save dialog as a list and order it according to the file date, with the newest file on top. This way you always see the last saved file and its number (e.g., _3, _4) when you enter the new incremental number.

Core Backup Rules

In contrast with only saving a file, even as a different files, a backup is defined as copying the file to another device (including online). In order for backups to actually work — preserve your data — they must be done in the following way:

- **Backup Regularly:** As long as the files only exist on one drive, they are at risk. Depending on the importance of the files,

DRAFT VERSION FOR FEEDBACK

backups should be done each day, week, or month. Set yourself a specific day to update the information on the device (e.g., the first of each month). Additionally update the information after (e.g., completed important step in a project) and before (e.g., vacation) significant events. Automation (see Box 1: Automation) makes even daily backups nearly effortless. For longer backups, schedule them when you do not need the computer for a while (e.g., overnight).

- **Use Different Storage Media:** Backups are only protected against a hard disk drive failure or theft of the device if they are on different medium. Putting backups in a different folder or another partition of the same drive means that when the drive is gone, all backups are gone. If your computer has an SD card slot, you can use it as «internal-external drive» (see Box 3: Internal-External Drive).
- **Store Backups in Multiple Places:** If the backups are in the same place, theft, fire, water or electricity damage can get them all. So store backups outside of the place you are living and working. Ensure you have at least one off-site storage of your backups. Online storage is cheap and data can be password protected. A safe-deposit box can be a good place to store an external drive. If you have two, one connected to your device and one in the safe-deposit box, you can swap them each month. Swapping the drives at the bank prevents you from having all your backup drives in your home at the same time. Doing an end-of-year backup of the key data on DVDs or M-Discs and storing them in the safe-deposit box can be useful as well (see Box 5: Savepoints).

As a rule of thumb, aim for at least three current copies of crucial data, with at least one off-site. For example, on an external hard disk drive on your desk (incremental backup roughly every day), in the cloud (e.g., encrypted Dropbox backup, every month), and in a safe deposit box (end-of-year backups).

More Risk Sensitive Backup Rules

The following rules are helpful, but more advanced.

- **If Possible, Keep on Backup on Your Person:** Key data and access data, e.g., logins and passwords for online backups, can be stored on your person. For example, an encrypted file on your smartphone, smartwatch (e.g., USB Disk App), or as a digital lifeboat (see Box 2: Digital Lifeboat). As loss, theft, or robbery is possible, strong encryption is a must.
- **Replace Storage Drives:** Regardless what you now use, the media you use to store your ideas will not survive. Partly due to time degrading the medium (USB-Sticks a few years, M-Discs a few decades), partly due to technical changes making old backup media obsolete. The media we use now will go the way of the 5 1/4 inch floppy discs, CDs, and the like. Keep this always in mind and make sure that when the physical media is gone you still have the ideas stored somewhere else.
- **Backups are Not for Active Use:** A

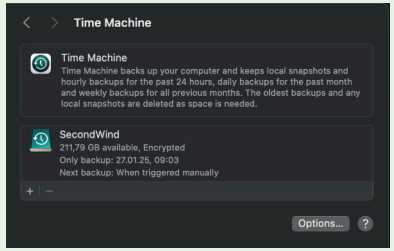
Box 1: Automation

The more frequently backups have to happen, the less they should depend on memory, motivation, or manual effort. A backup routine that requires you to remember, decide, connect devices, choose folders, and start the process every time will eventually be skipped.

Automation lowers the capture effort of backups: once configured, the routine happens at a fixed interval, when a drive is connected, or after a defined trigger. This is especially useful for data with a short max tolerable loss, such as an active idea collection or current project files.

Automation can take different forms, including more advanced ones (see Box 6: Advanced Automation):

Automation Type	Useful for	Watch for
Scheduled backups	Daily or hourly backup of active work	The device must be available at the scheduled time
Back up when drive is connected	External SSD or HDD used at desk	Easy to forget plugging in the drive
Versioned backup tools	Recovering older states after accidental deletion or corruption	Storage fills over time; check retention settings
Cloud/off-site backup tools	Remote copy of selected files	Sync is not the same as backup; check version history
Manual trigger with shortcut	Backup after major changes	Still depends on the trigger becoming habitual



On macOS, Time Machine is one example of automated, versioned backup software. It creates local snapshots and keeps hourly, daily, and weekly backups as space allows. You can add multiple external drives, start a backup manually, and exclude folders that do not need to be backed up, such as large temporary files.

Whatever tool you use, automation still needs occasional verification. A backup process can silently fail because the drive is full, disconnected, misconfigured, or excluded the wrong folder. Check the backup status regularly and restore-test a few files. Automation reduces effort — it does not remove responsibility. Put a monthly reminder in your calendar to check backup status and restore one random file.

DRAFT VERSION FOR FEEDBACK

Box 2: Digital Lifeboat

A digital lifeboat is a small storage device that contains your most relevant data and that you have almost always with you. It is for the case where your normal devices and local backups are unavailable: your home is destroyed, your laptop is stolen, you are traveling, or you have to leave quickly.

A digital lifeboat should contain only the data that would matter most in an emergency, for example:

- idea collection
- current projects
- essential documents
- recovery information for online backups
- password manager recovery information
- scans of crucial paper documents
- key finished works

It is not a working drive and not your only backup. It is an additional survival layer.

The storage medium must be:

Small enough to carry: Otherwise it will not be with you when needed

Encrypted: Loss or theft/robbery is more likely because you carry it

Large enough for the selected data: It must contain the actual emergency set, not just symbolic scraps

Rugged enough for everyday carry: Wallets, pockets, bags, heat, moisture, and pressure damage weak media

Easy to update: A lifeboat that is six months old may be comforting but useless

Small USB sticks, compact SSDs, or microSD cards can work. USB sticks can fit in a wallet. MicroSD cards can fit in a pendant or small container. An SSD can make sense in a bug-out bag or travel kit. Choose a medium you can actually update without making the routine annoying.

Because a digital lifeboat is carried around, encryption is mandatory for sensitive data. But encryption creates its own failure mode: if you forget the password or lose the recovery key, the lifeboat becomes inaccessible storage. Test that you can unlock it and restore files from it.

A digital lifeboat should be updated on a fixed rhythm, e.g., weekly, monthly, or after major project changes. After updating, restore-test at least one file. The point is not to feel safer because you carry a device. The point is to know that the device contains current, readable, accessible data.

A digital lifeboat is successful only if you can unlock it, open the files, and the contents are recent enough to matter.

Box 3: Internal-External Drive

Some notebooks and desktop computers have SD-card slots that are rarely used. With a flush microSD adapter, the slot can hold a microSD card almost permanently, turning it into a small separate backup drive. Physically, the card is distinct from the internal drive. Practically, it behaves like an always-available local backup medium.

This can be useful when you want quick backups while traveling or working away from your desk. If the internal drive fails, a recent copy may still be on the card. It also lowers friction: no cable, no external drive, no setup each time. Especially if you use (Advanced) Automation.

But this is only a limited backup layer. It does **not** protect against theft of the notebook, fire, water, electrical damage, malware, or mistaken backup scripts. If the card is mounted all the time, it may also be affected by some of the same failures as the main device.

Use an internal-external drive for:

- quick local backups while mobile
- temporary protection between larger backups
- active project or idea collection copies
- an additional layer, not the only layer

Do not use it as your only backup. It should be combined with at least one backup that is physically separate from the computer and preferably off-site.

MicroSD cards also vary strongly in speed and reliability. Use a reputable card, avoid counterfeit media, and restore-test files occasionally. If the backup contains sensitive data, encrypt it. A card hidden inside the device is not security — it is only easy to overlook.

Treat it as a convenient buffer between real backups, not as the backup scheme itself.

Figure 1: BaseQI USHII-420A with an 1 TB microSD Card inside in a MacBook Pro 14". The microSD is inside the SD card slot.

backup drive is not a second collection or a drive you use, e.g., at work. Have a single source of truth — e.g., the hard disk drive of your laptop. Everything else is a backup that is checked whether it actually works, but is never used actively unless your laptop drive is gone.

- **Never Use Sync as a Replacement for Backups:** Syncing data is often hailed as a better way of doing backups, but it is not.

DRAFT VERSION FOR FEEDBACK

Box 4: Encryption

Backups protect against data loss. Encryption protects against data exposure.

Use encryption whenever a backup contains sensitive personal data, valuable unpublished work, access data, private correspondence, contracts, financial documents, identity documents, or anything that would create damage if someone else could read it. This applies especially to backups that are online, off-site, or carried with you.

Encryption has a hard trade-off: if someone else gets the device, they should not be able to read it. But if you lose the password or recovery key, **you** may not be able to read it either. An encrypted backup without a recoverable key is just inaccessible storage.

So encryption requires two things:

1. **Strong encryption on sensitive backups:** Loss, theft, breach, or unauthorized access should not expose the data.
2. **Separate recovery information:** You must still be able to unlock the backup after stress, travel, device loss, or years of non-use

It sounds trivial but it happens: Do not store the only password or recovery key on the encrypted device itself. That is like keeping the combination for the safe in the safe.

Keep recovery information somewhere separate and secure, e.g., in a password manager, on paper in a protected place, or in another encrypted backup whose access you have tested.

There are several ways to encrypt backups (see Table 1)

On macOS, Disk Utility can create encrypted disk images (**.dmg**). A disk image works like a virtual drive: you unlock it, place files inside, then close it again. This can be useful for small sensitive collections, such as access data, contracts, or selected documents.

For larger data sets, encrypted disk images can become inefficient. If a 250 GB encrypted disk image changes, backup tools may need to copy the whole image again, even if only one small file inside changed. In that case, consider one of these alternatives:

- **Encrypt the whole backup device:** Useful for external drives, USB sticks, or microSD cards. On macOS, APFS with encryption can be used for suitable media.
- **Use full-disk encryption on the main device:** On macOS, FileVault encrypts the internal drive. Take the recovery warning seriously: if both password and recovery key are lost, the data is lost.
- **Separate sensitive data from large non-sensitive files:** Keep sensitive files in a smaller encrypted container and back up large media files separately. This avoids turning every small change into a huge backup operation.

Whatever method you use, test it:

1. Unlock the encrypted backup.
2. Restore a file.
3. Open the restored file.
4. Confirm that you still know where the recovery information is.

Encryption that has not been restore-tested is not protection. It is a bet. An encrypted backup only counts if you can still unlock it under bad conditions.

Method	Best for	Watch for
Encrypt the whole device	External backup drives, USB sticks, microSD cards, digital lifeboats	If the key is lost, the whole device is inaccessible
Encrypt a container / disk image	A selected folder or small set of sensitive files	Large containers may need to be copied in full after small changes
Encrypt before uploading	Online backups and cloud storage	You must manage the key yourself
Separate sensitive from non-sensitive data	Large collections with only some private material	Requires clear folder structure

Table 1: Encryption Options.

Syncing automatically keeps data on different devices the same. Meaning if you make a mistake or a file gets corrupted, that error will be synced automatically as well. A backup is putting all your crucial data in a place where it is secure from hardware failure, accidents, and other calamities.

- **Watch for Strange Behavior:** Some drives indicate in advance that something is wrong and they fail soon. For example, copying the same file sizes takes longer or hard disk drive give strange «clicking» sounds. If you notice anything strange distrust the device and use an additional backup drive. If you copy from the drive, spot-check the files to ensure they are not corrupted.
- **Context-Dependent: Encrypt Backups:** If you backup personal data or highly valuable ideas, encryption is recommended (see Box 4: Encryption). However, encryption comes with the risk of losing access to the data if you forget the password or lose access to the key.
- **Context-Dependent: Secure-Wipe or Destroy Backup Drives:** If you discard backup drives, e.g., they have become unreliable, crashed, or too small, secure-wipe (e.g., secure-format) or physically destroy them. There are programs that can wipe existing data by overwriting it again and again with random data. Otherwise even deleted data can be restored with the right tools.

DRAFT VERSION FOR FEEDBACK

Backups

Making Backups Easier on You

The less effort backups take, the more likely you are to do them. The following tips can help:

- **Do Incremental Updates:** Backup space is limited and copying a lot of files takes time, so it makes sense to save only the files that changed since the last backup. Apps such as macOS's Time Machine or commands such as rsync automatically recognize which files need to be updated on the backup drive.
- **Use a Fast Drive:** While large hard disk drives are rather cheap, a good SSD is much faster. There is a huge difference between a backup taking a couple of hours vs. a few minutes.
- **Get the Largest Backup Medium Possible:** Unless you are only dealing with small text files, a large backup drive (e.g., 2 TB for SSDs, 16 TB for hard disk drives) saves you the headache of deciding which of your data to backup.

Relevant Data to Back Up

Not all data needs the same backup treatment. Some data changes daily, some almost never changes. Some data would be painful to lose, some would be catastrophic, and some can simply be downloaded again.

If all your data fits comfortably on one backup drive, a full automated backup is easiest. Use a backup drive larger than your current data volume and let a tool such as Time Machine or another automated backup system handle it. Prioritization becomes important when the amount of data is large, the backup takes too long, or some data needs stronger protection than the rest. See Table 2.

Data Type	Examples	Why it matters	Backup priority
Access Data	Password manager recovery codes, account recovery keys, login information, encryption recovery keys, hints for where critical access information is stored	Without this, your backups may exist but remain inaccessible	Update immediately after every change; keep at least one (protected!) offline/paper fallback
Active Creative Core	Idea collection, current manuscript, active project folders, notes for ongoing work	This is the living part of the creative system. Lost ideas and unfinished work often cannot be recreated accurately	Back up daily or after major work sessions; use versioned backups
Critical Personal Data	Contracts, tax documents, certificates, legal documents, financial records, private photos, medical or identity documents	Loss creates practical, financial, legal, or personal damage	Back up regularly; encrypt; keep at least one off-site copy
Finished Works and Project Archives	Completed manuscripts, released works, final exports, source files for finished projects, documentation	These represent realized work and may be needed for reuse, proof, publication, or later development	Back up after completion and include in monthly or yearly savepoints
Reference and Working Library	PDFs, ebooks, research material, images, templates, reusable assets, collected examples	Useful for future work, but often less urgent than active projects	Back up weekly or monthly, depending on use
Large Media Collections	Photos, videos, audio recordings, music libraries, film material, scans	Often large, slow to back up, and partly irreplaceable	Separate irreplaceable originals from replaceable media; back up originals more aggressively
Replaceable or Low-Value Data	Downloaded movies, software installers, cache folders, temporary exports, files easily downloaded again	Losing it is inconvenient, not serious	Optional; exclude from backups if it makes the backup routine too heavy

Table 2: Relevant Data to Back Up

Back up the most irreplaceable and most frequently changing data first. Do not let large, replaceable files make backups of your idea collection or active projects harder.

Backing Up Paper

The problem with working on paper is that there is only the original. If that is destroyed, all ideas are lost. Single pages can be quickly digitized with a document scanner (see ▯ Digitizing Information). If you work with notebooks, then, e.g., a smartphone camera, can be used to make digital copies. That takes time but can be a life-saver if the physical object (e.g., idea notebook) is lost. If you work with paper notebooks a lot, then building a photography stand might make these backups easier.

Box 5: Savepoints

Savepoints are used in games as places to return to when you die or fuck up later. Applied to backups, a savepoint is a **deliberately frozen full copy** of important data at a specific point in time, e.g., at the end of the year, before a major system change, after finishing a large project, or before migrating to a new tool.

Unlike regular backups, savepoints are not continuously updated. That is their value. If your active files are corrupted, your backup script mirrors a deletion, or a reorganization turns out to be a mistake, a savepoint gives you an older known-good state to return to.

Create a savepoint by copying the important data to a separate medium, checking that the files can be opened, labeling it with the date and scope, and putting it away. Do not use it as a working drive. Do not update it casually. It exists for the moment when the normal backup process has failed or preserved the wrong state too faithfully.

Useful moments for savepoints:

- **End of year:** Creates a stable annual archive
- **Before major reorganization:** Lets you reverse a bad restructuring
- **Before changing tools or formats:** Protects against migration failure
- **After finishing a major project:** Preserves the completed work and source files
- **Before deleting or pruning large amounts of data:** Protects against overzealous cleanup
- **After restoring from failure:** Captures the recovered state before further changes

Large storage media are usually needed, e.g., an external HDD. For crucial smaller data, write-once media, DVDs, M-Discs, or a dedicated SSD/USB stick can be options. Whatever medium you use, the savepoint only counts if you have checked that files can be read.

Label it plainly, e.g., **Savepoint — 2026-12-31 — Idea Collection + Projects + Documents**. Then store it somewhere separate from your active system, preferably off-site.

Backup Options: Local

Local backup media are under your control. They are usually faster to restore from than online backups and do not depend on a company account, subscription, or internet connection. But local backups are not enough by themselves: fire, theft, water, or electrical damage can destroy all devices in the same place. At least one backup should eventually be off-site.

Choose the medium according to the role it has in your backup scheme — see Table 3.

For active work, speed matters because slow backups are avoided. For large archives, capacity matters more. For data you carry with you, size and encryption matter more than speed.

Do not choose one medium for everything. A good scheme often combines them: an SSD for frequent local backups, an HDD for large archives or off-site rotation, and a small encrypted USB stick or microSD card for key data you carry with you.

Medium	Best for	Strengths	Weaknesses
External Hard Disk Drives (HDDs)	Large stationary backups, media archives, monthly savepoints, off-site drive rotation	Large capacity for relatively little money; useful for many terabytes	Slower, audible, vibration-sensitive, easier to damage while running
External SSDs	Frequent backups, portable backups, active project backups, fast restore	Fast, silent, compact, no spinning parts; good when backups should take minutes, not hours	More expensive at large capacities
USB sticks	Small emergency backups, access data, digital lifeboats	Cheap, small, easy to carry	Easy to lose; variable quality; not ideal as only backup
SD and microSD cards	Internal-external drives, digital lifeboats, compact encrypted backups	Extremely small; can stay inside some devices; useful when portability matters	Easy to misplace; speed and reliability vary strongly; awkward for large backups
Optical media / write-once media	Rare long-term savepoints, finished works, yearly archives	Harder to accidentally overwrite; useful as frozen snapshot	Limited capacity; slower; requires compatible reader later

Table 3: Possible Backup Media

Backup Options: Online

Online storage can be useful because it is **off-site by default**. If your home is destroyed, robbed, flooded, or hit by electrical damage, an online backup can still survive. It also allows backups when you are away from your usual backup drives.

However, online storage is not fully under your control. You depend on the provider, your account access, the provider's terms of service, internet access, and the continued existence and reliability of the service. Accounts can be locked, companies can change conditions, services can fail, and data breaches can happen.

Use online backups as **one layer**, not as the whole backup strategy (see Table 4).

Backups Worksheet

Use case	Online backup is useful because...	Main risk	Countermeasure
Off-site copy	It survives local disasters	Account loss or provider failure	Keep local backups too
Frequent backup of active work	It can update automatically	Sync can copy mistakes, deletions, or corrupted files	Use versioned backup, not simple sync
Sensitive personal data	It can be stored remotely	Provider, breach, or account compromise exposes data	Encrypt before upload
Access while traveling	Data remains reachable without your backup drive	No internet or locked account means no access	Keep key data on an encrypted local lifeboat
Large archives	No physical drive handling	Upload/download may be slow; subscriptions continue	Use only for selected crucial data

Table 4: Online Backups

For sensitive data, do not rely only on the provider's promises of encryption. Encrypt the data yourself before uploading it, or use a system where only you control the key. This adds another responsibility — **if you lose the password or recovery key, the backup may become useless.**

The access information for online backups must itself be backed up. If the online backup survives but you cannot log in, it might as well be gone. Keep recovery codes, password manager access information, and encryption recovery information somewhere secure and offline, e.g., on paper or in another protected backup.

Online backups work best when combined with local backups:

- Local backup for fast recovery.
- Online backup for off-site survival.
- Encryption for sensitive data.
- Offline access information so you can actually restore it.

A backup you cannot access is not a backup — it is inaccessible storage.

Dealing with Accidents and Failures

Accidents will happen. The point of backups is not to prevent accidents, but to limit their impact.

If a drive fails, a laptop is stolen, or a lightning strike destroys your computer, the question is simple: What is the newest restorable copy, and where is it? If your backups are regular, versioned, and stored in different places, the damage is limited. You lose time, hardware, and perhaps the latest changes — not the whole idea collection.

The first rule after data loss is: **do not make the damage worse.** Do not overwrite backup drives. Do not immediately run unfamiliar recovery tools. Do not keep using a failing drive for ordinary work. Stop, identify what still exists, and work from copies wherever possible.

Core Sequence: **Stop. Preserve what remains. Find newest backup. Restore elsewhere. Check what is missing. Recover if possible. Rebuild what still matters. Fix the backup failure.**

A practical sequence:

1. **Stabilize the situation:** Protect what remains. Disconnect failing drives if continued use might damage them further. Do not erase, format, reinstall, or «clean up» anything until you know what is still recoverable.
2. **Identify the newest usable backup:** Check local backups, off-site backups, online backups, savepoints, and digital lifeboats. Note the date of the newest available version. Do not update or overwrite backups until recovery is complete.
3. **Restore to a separate location:** Restore recovered files to a new drive or temporary folder, not over the damaged or uncertain source. Open a sample of restored files and check whether they are readable.
4. **Check what is missing:** Compare the restored state with the work you remember doing since that backup. List the missing projects, files, versions, and access data. This gives you a concrete damage report instead of a vague sense of loss.
5. **Recover technically if needed:** If no usable backup exists, or if the backup is incomplete, try technical recovery. Use different cables, ports, readers, or computers. If the data is valuable, get external help before experimenting too much yourself.
6. **Rebuild the creative core:** If the idea collection or project material is truly gone, write down what you still remember. Start with the current project, then active ideas, then important older projects. Do not try to recreate the entire old collection at once. Rebuild the parts that still have force.
7. **Decide what is worth salvaging:** Loss also removes false obligations. Some projects are worth rebuilding; others were only being carried forward because they already existed. Decide which ideas and works still matter enough to restart.
8. **Change the system only after the emergency:** Once recovery or rebuilding is underway, examine why the loss was possible. Was there no backup? Was it too old? Was it in the same place? Was it encrypted but inaccessible? Was it never restore-tested? Fix the specific failure, not your whole system at once.

Recovering Data

- **If a storage medium failed:** Try different hardware: another cable, port, enclosure,

DRAFT VERSION FOR FEEDBACK

card reader, drive reader, or computer. Sometimes the storage medium is not dead — the connection or enclosure is.

- **If the drive behaves strangely (becomes slow, clicking sounds on HDDs):** Stop using it for normal work. Copy the most important files first, not the easiest or largest ones. A failing drive may only remain readable for a short time.
- **If the data is highly valuable:** Get professional help before running random recovery tools. Some attempts can make later recovery harder.
- **If you recover files:** Copy them to a separate drive and check them. A file that exists but cannot be opened is not recovered.

Afterward, update the backup scheme. The failure should produce a concrete correction, e.g., more frequent backups, off-site storage, versioned backups, access-data backup, encryption-key backup, or regular restore tests.

Optional: Damage Report

You can create a damage report to make your conclusions more accurate and valid. For an example see Table 5.

Item	Newest surviving version	Missing since backup	Recoverable?	Next action
Idea collection	2026-05-14 cloud backup	Three days of notes	Partly from memory	Restore, then idea dump
Current manuscript	Yesterday's local backup	Morning edits	Maybe	Check autosave/version history
Access data	Paper recovery sheet	None known	Yes	Verify logins
Project archive	Monthly savepoint	None	Yes	Restore later

Table 5: Example of a Damage Report.

Data	Source of truth	Approx. size	Max tolerable loss	Backup method	Location	Encryption?	Restore tested?

Table 6: Table of which data must be preserved.

Trial Definition

If you have never thought about backups, or only did them occasionally, establish a backup scheme before you need it.

Use an Integration Worksheet trial to test whether a backup routine actually integrates into your life. Backups only work if two things are true:

1. **The backups are current enough:** «Current enough» depends on the data. How much work can you tolerate losing: one hour, one day, one week, one month? The backup rhythm must be more frequent than that loss window.
2. **The backups can actually be restored:** A backup that has never been restored is only a hope. Test restoration by recovering one file, one folder, and one older version from each backup location used in the trial. Open the restored files and confirm that they contain the expected content. This also tests whether you still have the required passwords, recovery keys, devices, and software.

A backup scheme is good enough when the newest restorable copy is never older than the loss window you chose.

Default path: Back up your idea collection and active projects daily to an external SSD using automated backup software. Restore-test one random file each week. After four weeks, add one off-site layer. Start with the smallest set of data whose loss would seriously damage your work.

Establishing Your Routine

First, list the data that must be preserved. Include your idea collection, active projects, finished works, access data, and important paper documents that should be digitized (see Table 6).

Source of truth: the place where the active version lives and where changes are made.

Choose the shortest loss window that would still be acceptable (see Table 7). Then choose a backup rhythm that is more frequent than that window, because backups can fail, be skipped, or happen before rather than after the important change.

Practically, this can become a Backup Routine akin to Table 8.

The first table helps you classify data. The second table helps you design the actual routine. The trial tests whether the routine happens and whether the files can be restored.

First Trial

Do not try to build the perfect backup system in the first round. As final resilience target, aim for at least three current copies of crucial data, with at least one off-site. But as a first trial, test one layer well.

A good first trial:

- one automated local backup,

Backups Worksheet

Max tolerable loss	Use for	Backup rhythm	Practical setup	Minimum check
Minutes to 1 hour	Current manuscript, active idea collection, deadline work	Continuous/hourly versioned backup or snapshotting	Auto-save/version history + automated backup + manual version before risky changes	Confirm older versions can be restored
Half a day	Active creative work where losing a session would hurt	After each major work block	Shortcut, script, or backup tool triggered after significant changes	Restore one changed file weekly
One day	Most active projects and idea collections	End of each workday	Automated daily backup to external drive or local backup volume	Check backup status weekly
A few days	Personal documents, less active projects, reference material	2–3 times per week or after meaningful changes	Scheduled backup; manual backup acceptable if tied to fixed days	Restore-test monthly
One week	Stable archives, photos, finished works, personal library	Weekly	External drive, cloud backup, or rotating drive	Monthly spot-check
One month	Rarely changing archives, old projects	Monthly	Monthly savepoint or full archive copy, preferably off-site	Open sample files after copy
One year / major milestone	Completed works, yearly archives, tax/ legal documents	Yearly or after major project completion	End-of-year savepoint on separate medium, stored off-site	Verify readability before storing
Loss acceptable / replaceable	Downloaded media, installers, temporary exports	Optional	Do not back up, or keep one loose archive	None, unless replacement would be costly

Table 7: Tolerable Loss and Backup Routines.

- one clearly defined backup rhythm,
- one restore drill,
- one calendar reminder to check status.

Example

- **Change behavior:** At the end of each workday, the backup drive is connected and the automated backup runs. Every Friday, one random file is restored and opened.
- **Success:** For four weeks, the local backup completes on at least 20 of 28 days, and all four weekly restore tests succeed.

Data	Max tolerable loss	Source of truth	Backup rhythm	Backup destination	Restore test
Idea collection	One day	Laptop folder	Daily automated backup	External SSD + encrypted cloud	Test a random note weekly
Current project	Half a day	Project folder	After each major work block	External SSD	Test the latest version weekly
Finished works	One month	Archive folder	Monthly savepoint	Off-site drive	Open sample files after copy
Access data	Immediate / one day	Password manager + paper fallback	After every change	Encrypted backup + paper recovery note	Confirm access monthly

Table 8: Possible Backup Routine.

- **Abort:** Restore fails twice, backup requires more than two minutes of attention per day, or the process requires manual decisions each time.
- **Ambiguity:** Travel days count if the laptop was not used for relevant work or if the lifeboat/cloud backup was updated within 48 hours.

If the first layer works, add the next layer in a later trial: off-site drive, encrypted cloud backup, savepoints, or digital lifeboat.

Stop improving the backup scheme once the restore drill works and the routine is stable. Do not add another layer unless a concrete risk remains uncovered.

Major Resilience Tests

Once you have a working scheme, test it against imagined losses. This is not the first trial; it is a stress test for the backup architecture.

In increasing severity:

1. Your main computer goes up in smoke. All local active data is lost.
2. You wake up and your home is on fire. You have 10 seconds.
3. You return home to a burned-out building. Everything stored there is gone.
4. You have to leave the country now with what you have on you.

For each scenario, answer the Resilience Test Questions (see Table 9).

For a more valid test, set a calendar alarm — or ask someone else to send you a message at a random future time. When the message arrives, run one scenario immediately. Do not adjust the backup scheme first. The point is to see what would survive under the current system.

Question	Answer
What survives?	
What is lost?	
How old is the newest surviving version?	
Can you access passwords, keys, and recovery information?	
Which backup layer failed or was missing?	
What needs to change?	

Table 9: Resilience Test Questions

Hand-Off

Creative data is not just information. It often contains formulations, structures, unfinished arguments, captured ideas, and project states that cannot be reconstructed reliably once lost.

You may be able to rebuild after a loss, but you will not necessarily rebuild the same work. Backups are therefore not technical housekeeping. They are part of keeping the creative system viable.

Use the Integration Worksheet to trial one backup routine: choose the data, define the maximum tolerable loss, set up one backup layer, and restore-test it.

The standard is simple: **The newest restorable copy is never older than the loss window you chose.**

Once that works, stop improving the backup scheme unless a concrete risk remains uncovered.

The useful time to build this routine is before restoration is needed.

More Information

- **PhD Comics:** Jorge Cham has a nice comic series on backups. https://phdcomics.com/comics/archive.php?comicid=638 (and the next ones).

Box 6: Advanced Automation

A more controlled way of updating backup media is to use scripts. This can make backups fast and comfortable, especially when you use different backup drives for different purposes, e.g., desk SSD, off-site HDD, or digital lifeboat USB stick.

However, scripts are unforgiving. They do what you wrote, not what you meant. If you confuse source and target directories, you can overwrite or delete the wrong files. The **--delete** option is especially hazardous: it removes files from the target if they no longer exist in the source.

So this is an advanced option:

Do not use scripts on real data until you have tested them on disposable dummy folders and checked the source/target direction. For first trials, use non-destructive tools or versioned backup software.

A script like the following creates a **one-way mirror** from selected folders on the Mac to a backup drive — see Table 10.

The source folders are on the Mac. The target folders are on the backup drive. Files that are new or changed in the source are copied to the target. Files that were deleted from the source are also deleted from the target because of **--delete**.

This is useful when the backup drive should match the current state of selected folders. It is not the same as a versioned backup. If you accidentally delete a file from the source and then run the script, the deletion can be carried into the backup. For this reason, scripted mirror updates should be combined with versioned backups, savepoints, or another backup layer that preserves older states.

Before using a script like this on real data:

1. Test it on dummy folders.
2. Run it with **--dry-run** and read what would happen.
3. Confirm that the source is your active data and the target is the backup drive.
4. Confirm that the backup drive is mounted at the expected path.
5. Restore-test files afterward.

On macOS, the scripts for different backups can be started via Spotlight by wrapping them in a small Automator app (see Figure 2).

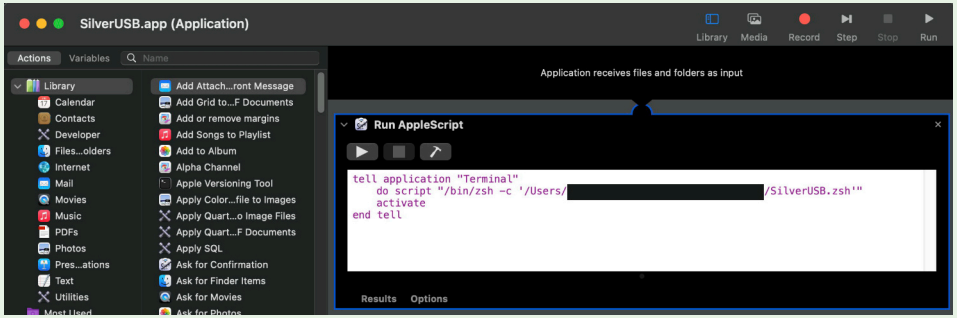


Figure 2: Automator to execute a backup script.

```
tell application "Terminal"
  do script "/bin/zsh -c '/Users/PATHTOSCRIPT/SilverUSB.zsh'"
  activate
end tell
```

This makes the backup action easy to trigger, while still leaving you in control of when it happens.

For backups on mobile devices (e.g., iPhones), an encrypted USB-C stick is usually simpler than scripting backups directly to mobile devices. If you use an iPhone with USB-C port, you can usually use USB-C sticks, even encrypted ones.

In any case, a one-way mirror protects against drive failure — it does not protect well against your own mistaken deletions unless another backup layer preserves older states.

```
#!/bin/zsh
set -euo pipefail
trap 'echo "Error at line $LINENO"; exit 1' ERR
rsync -avrh --update --delete /Users/USERNAME/Documents/IdeaCollection/ /Volumes/Backup/IdeaCollection/
rsync -avrh --update --delete /Users/USERNAME/Documents/Contracts/ /Volumes/Backup/Contracts/
rsync -avrh --update --delete /Users/USERNAME/Documents/PersonalLibrary/ /Volumes/Backup/PersonalLibrary/
echo "\n\e[32mBackup updated.\e[0m\n"
```

Table 10: Backup Script Example.