

KitForma.

PRACTICAL FIELD GUIDE

Work Files You Can Find and Restore

Choose an accepted version, check its bytes and rehearse a restore into a separate folder

01

02

03

SAMPLE EDITION

ENGLISH / 1.0 / 2026

How to use this guide

Individuals and small teams organizing local project files and testing a bounded recovery task.

Inspect a fictional project with misleading filenames and one deliberately damaged copy. Use actual files, an exact SHA-256 manifest and a read-only local checker to select, restore and verify two accepted documents.

- Build a scoped file inventory.
- Separate version authority from byte equality and readability.
- Verify exact file digests without rewriting the reference.
- Record a restoration rehearsal and its limits.

Scope and limitations

- No cloud backup setup, credential vault or encryption certification is supplied.
- The exercise does not test device failure or independently authenticated provenance.
- No original file deletion or overwrite is required.
- The helper requires Python 3.9 or newer because it uses `Path.is_relative_to`.

Published by KitForma. Edition 1.0. Sources checked 1 October 2026. People and businesses in worked examples are fictional. This is an independent publication, not endorsed by the platforms mentioned.

Copyright 2026 KitForma. Usage, support and refund terms are included in `LICENSE.txt` in the purchased package.

Contents

How to use this guide	2
Contents	3
Define what you need to recover	4
Sources and tools	6

Define what you need to recover

File organization becomes useful when it answers a concrete recovery question. “Keep everything safe” is too broad to test. “Find the accepted workshop brief and restore the matching materials list into a new folder” is small enough to rehearse. This workbook helps you define that kind of task, inspect a limited project collection, record what matters, and verify a recovery copy without overwriting the working files.

The fictional project is Fern Studio's paper workshop. Its folder contains an early brief, an accepted brief, a second file with a reassuring but misleading name, a materials list and a copy that has been deliberately damaged for the exercise. None of these files contains real customer records. The complete contents are supplied as text fixtures, so you can inspect them without installing a document editor or connecting a cloud account.

The requested recovery is specific: recover the accepted version two brief and the matching materials list from the supplied archive folder. The acceptance note identifies those two paths. It does not say that the newest file is always correct or that a file named final must be authoritative. You will use the note, file contents and byte-level checks together, because each answers a different question.

This is a small local rehearsal, not a backup service, security certification, regulated retention policy or disaster-recovery plan. It does not test a failed physical disk or restore a remote account. Those boundaries make the exercise useful rather than trivial: you can actually verify every supplied file, detect the deliberately wrong candidate, and explain exactly

what the successful result establishes. Later chapters show how to record the additional risks that remain outside this rehearsal.

Sources and tools

[S1] **Library of Congress: Personal Digital Records**

<https://www.digitalpreservation.gov/personalarchiving/records.html>

Checked: 2026-10-01

[S2] **NIST FIPS 180-4 Secure Hash Standard**

<https://csrc.nist.gov/pubs/fips/180-4/upd1/final>

Checked: 2026-10-01

[S3] **Python hashlib documentation**

<https://docs.python.org/3/library/hashlib.html>

Checked: 2026-10-01

Inside the complete guide

Inventory a bounded collection before moving anything

Choose names that carry limited, honest meaning

Separate authority, equality and readability

Create a manifest you can reproduce

Test a deliberately wrong candidate

Restore into a new destination

Record access and dependency limits

Capstone: recover the accepted workshop packet

Answer key and a sustainable habit