SYNC ALL — MIR library

Русский · User guide

Contents

- MIR scope
- Sources and completeness
- Reliability without old generations
- The plan in the data folder
- Commands and results
- Deleting ._runtime

MIR scope

SYNC ALL operates on selected DATA only. Managed library folders are replaced by verified server copies. Local edits, extra files, old Git state and repositories removed from Forgejo Mirror/automatic Repositories are deleted. Old generations of working folders are not accumulated.

Outside deletion: Tunnel, connection settings in config, complete recovery Snapshots, durable Recovery and explicit protectedLocal paths: which paths are protected is named by the plan in the data folder. These exceptions are not replaced by Git content. Other R2 buckets, KV/DO and cloud drives are outside this library's verification boundary.

Sources and completeness

The full repository catalog is read on MAIN VPS. Known projects retain familiar folder names; new projects use Repositories//. All refs/HEAD, git fsck and checkout files are verified. Matching verified local Git objects are reused; changed or corrupt copies are downloaded again.

The newer daily/manual R2 recovery archive is selected. Checks cover the outer SHA-256, complete inner manifest, all nested payloads, operational file/unit bytes and Unix metadata, and active snapshot

Git HEADs. New payloads are retained automatically. Snapshot and live Git have separate dates and freshness evidence.

Code comes from Git; published Portal/Crypto/Planning data comes from allowed snapshot subtrees. MAIN VPS Backup is a Git source master, not a flat overlay of installed files. SECOND VPS is not queried: its full transport-config.tar.gz comes from recovery. MIR removes obsolete flat transport configs in that folder.

Reliability without old generations

Stage and verify first, then replace transactionally on the same volume. before exists only temporarily to roll back an unfinished replacement. Failed swaps restore old folders; the next run first recovers an interrupted transaction. Changes during staging stop replacement; pre-existing local edits are removed by MIR.

After successful verification, temporary before, heavy staging and old SYNC ALL caches are deleted. Current library, recovery set and small JSON proofs remain. An incomplete archive, Git error or unverified catalog never authorizes cleanup.

An OS lock per DATA prevents concurrent runs. Deletion is restricted to owned directories; junctions/symlinks are refused. Keys are not imported or changed, and SSH trust is not weakened. No collector, model, publication, OS update or actual server restore is invoked.

The plan in the data folder

What to synchronise - MAIN VPS and its folders, the R2 bucket of snapshots, the Forgejo repositories, the parts of the recovery archive - is described by config\mirror.json of the data folder, version 2. That is the household's data and the program holds none of it: config\mirror.sample.json is a template without a single value, a <...> in place of each. SYNC ALL builds no plan from the template and does not finish an old version 1 plan: without a v2 plan, or with a <...> left in it, it stops changing nothing and says what to do. On a new machine the plan is copied from the data folder of a machine where SYNC ALL has run, or filled in from the template.

Commands and results

Requires Git, Windows OpenSSH, Python 3.10+ and the application's own rclone. Python is looked for as `py -3`, then `python.exe` on PATH; the Microsoft Store stub does not count and the version is

checked. Without Git or a fitting Python SYNC ALL says at once what to install. The wrapper runs under App/AudionPs.exe. No drive letter is hard-coded.

```
Runs\sync-backup.cmd -Root "<DATA>" -WhatIf -NoPause
Runs\sync-backup.cmd -Root "<DATA>" -NoPause
Runs\sync-backup.cmd -Root "<DATA>" -Only "Audion Portal,Forgejo Mirror" -NoPause
Runs\sync-backup.cmd -Root "<DATA>" -Snapshot "<archive.tar.gz>" -NoPause
```

DRY RUN reads the SSH catalog without writing DATA or downloading the archive. -Only restricts installation, not input completeness checks. -Snapshot reuses an existing archive and adjacent .sha256; live Git is still verified. MIR is the permanent mode; no separate conflict override is needed.

BACKUP-CATALOG.md/json describes the current copy, HEAD/refs, payload hashes and protected paths. The current Recovery/sync-all/sync-all- retains transaction.json, result.json or failure.json, plus verification inventory. Exit 0 means full success. The unpacked Recovery is for reading; restore Unix modes from the original tar. These checks do not perform an actual server restore.

Tests/Test-LibrarySync.ps1 covers MIR, protected data, corrupt archives, interruptions, rollback and safe cleanup on local fixtures without real network. After dialog changes, also run Tests/Stand/Run-Stand.ps1.

Deleting ._runtime

With the program closed and CLI operations finished, ._runtime can be deleted both in the program and in each DATA. It holds only rebuildable files, not key backups or unique synchronization state. The SYNC ALL journal and temporary rollback live in DATA/Recovery/sync-all and survive scratch deletion after an interruption. Successful MIR still removes previous SYNC ALL generations.

Cloud session.conf and its source clouds.conf checksum live in DATA/config/rclone. The next opening preserves refreshed tokens; a newer explicit reconnect is not overwritten by a stale session. Conflicting session bytes are retained in DATA/Recovery/cloud-sessions. BISYNC pairs and listings live in DATA/config/bisync; that operation requires a selected DATA folder.

Preparation of DATA migrates the old layout with hashes and empty-folder checks. An explicit offline migration is also available: App/AudionPs.exe -NoProfile -ExecutionPolicy Bypass -File Engine/Scripts/Migrate-RuntimeStorage.ps1 -Root "" -IncludeProgram Use IncludeProgram only for the DATA that owns the old program files. Name conflicts never overwrite different copies. Working machine and server configuration never returns to the public project.