

QUICK START EDITION

myOrganizer: CURATOR Quick Start Guide

A safe, concise path from setup to first organized collection

Collection Utility for Retro Archive Tracking, Organization, Repository

Updated: October 11, 2026 (v2.5.0-beta.1)

Created, architected, and managed by Miguel A. Ortiz

Development and coding assistance: OpenAI ChatGPT

Third-party tools, authority data, and trademarks remain credited under their respective notices.

Copyright © 2026 Miguel A. Ortiz. All rights reserved except as granted by the applicable license.

CURATOR organizes source files into a LaunchBox-compliant collection output. Use this guide to prepare the required folders, launch the gateway, run the first build, and view the final collection report.

Critical Safety Warnings

These warnings apply throughout this guide. Read them before configuring folders or starting a run.

PROTECT SOURCE FILES - Use copies, not irreplaceable originals, for the Source folder. Keep a verified backup and inspect CURATOR output before moving or replacing source material.

CLEAR REPLACES OUTPUT - Clear removes existing contents from the configured Organized output folder. Use it only when that folder is safe to replace. Use Append when existing output must remain and its receipt evidence is intact; invalid retained collection evidence stops safely.

PRESERVE INCOMPLETE SETS - Keep this folder and its contents in the collection output. Manually removing retained incomplete-set archives can prevent a later Append or Resume run from completing.

PRESERVE ACTIVE EVIDENCE - Do not move, edit, rename, or delete the following while CURATOR is running:

-  Active relational and manifest evidence
-  Run-state records `run_state`
-  Working data `workspace`
-  Stage logs and reports
-  Placement receipts
-  `database\CURATOR.db` authority evidence

 **Why this matters** — Cleanup actions can remove recovery and audit history.

AUTHORITY MATCHING CAN TAKE HOURS - Runtime scales with the volume and size of files processed. MAME support in myOrganizer: CURATOR is a VERY EARLY, experimental implementation. It must NOT be relied upon as a proven means of organizing MAME sets. Continue using clrmame as the definitive MAME ROM manager. Stage 5 attempts supported CDI, NRG, and MDF/MDS decoding through the bundled dependencies; failed images are logged, retained unchanged, and continue to review.

CDI DAT MATCHING REMAINS UNVERIFIED - No tested Dreamcast or Jaguar CD CDI has matched a loaded DAT, either as the original image or through its decoded tracks. Successful decoding or conversion does not confirm DAT identity or playback compatibility. Review unmatched CDI files and their generated outputs.

Before You Launch

Antivirus may interrupt processing

Antivirus and other security software may interrupt myOrganizer: CURATOR by temporarily blocking access to files it needs. This can cause delays or prevent an operation from completing. If this happens, review your security software's alerts and activity history. An access error alone does not confirm that antivirus software caused it.

DolphinTool warnings

If DolphinTool displays a blocking Warning dialog, read its message before dismissing it with OK. A warning alone does not establish its cause.

Install PowerShell 7.6.1 or newer.

Keep the CURATOR application folder intact. Do not remove files from these folders:

- 📁 config
- 📁 core
- 📁 dependencies
- 📁 scripts
- 📁 templates
- 📁 tools
- 📁 platforms
- 📁 DATs
 - 📁 database

Treat the complete application-provided 📁 dependencies folder as required.

MDF/MDS NOTICE - No separate Aaru setup is required. Some pairs are decoder-incompatible; CURATOR logs the failure, preserves both sources, and moves on normally.

Treat the complete application-provided 📁 tools folder as required.

Do not use 📁 workspace or 📁 _temp for permanent files. CURATOR uses those as working areas.

Prepare DAT Files

CURATOR needs DAT files before it can build useful authority.

Place DAT files here:

 DATs

DAT files may be placed directly in  DATs, or organized like this:

 DATs\<SourceAuthority>\<ParentCompany>

Common DAT sources:

- No-Intro / DAT-o-MATIC: <https://datomatic.no-intro.org/>
- Redump: <https://redump.org/downloads/>
- TOSEC: <https://www.tosecdev.org/downloads>
- MAME / Software Lists: <https://www.mamedev.org/release.html>

Prepare Official MAME Software Lists

MAME support in myOrganizer: CURATOR is a VERY EARLY, experimental implementation. It must NOT be relied upon as a proven means of organizing MAME sets. Continue using clrmame as the definitive MAME ROM manager.

For Atari 2600 cartridge and cassette authority, use the official per-list files shipped with the same MAME release:

-  a2600.xml
-  a2600_cass.xml

1. Download the intended official MAME release from <https://www.mamedev.org/release.html> and extract it. Use one release consistently; do not mix software-list files from different MAME releases.
2. Open the extracted MAME hash folder and locate the official per-list XML files.
3. Copy  a2600.xml and  a2600_cass.xml into the appropriate authority folder, for example  DATs\Atari\MAME\Atari 2600. CURATOR reads the per-list XML without loading an external DTD.
4. Run Prepare DAR, or, after database initialization, run Stages 1, 2, and 3 separately from the full menu. Review Stage 2 and Stage 3 evidence to confirm both software lists were accepted and published to Core.Database.
5. When updating MAME, replace both XML files from the same new release and rebuild Stages 1-3. Retain the MAME release number as provenance for the authority baseline.

Use the official per-list XML files from the MAME hash folder for these inputs. A separate DTD file is not required.

Prepare Atari 8-Bit Preservation Authority

Unsupported HyperList inputs

Do not use the Atari 2600 HyperList XML or its derived DAT as authoritative ROM definitions, including copies from older packages.

CURATOR recognizes Atari 8-bit Preservation exports by their contents:

-  Atari 8-bit Software Preservation Initiative export - physical-release provenance with machine-readable dump hashes from <https://www.a8preservation.com/v2>
1. Generate or download the current machine-readable export from the authority's official site. Keep the unmodified export and record its retrieval date and source URL.
 2. Place each export anywhere under CURATOR's  DATs folder. Use a clear Atari 8-Bit subfolder; filenames are descriptive only because Stage 2 recognizes the format from its content. CURATOR public packages include the eight individually allowlisted CURATOR-created authority DATs: the Atari 8-bit Preservation DAT, six Atarimania GameBase DATs for Atari 2600, 5200, 7800, Jaguar, Lynx, and ST, and the Atari Lynx Dragnet supplemental DAT; all other DAT payload remains user-supplied and excluded. Sets from GameBase-derived DAT sources are placed beneath `<Output>\<Company>\<Platform>\GameBase\`. Category, revision and alternate-dump subfolders follow that bucket. Existing output is not automatically reorganized into the new bucket.
 3. Run Prepare DAR, or, after database initialization, run Stages 1, 2, and 3 separately from the full menu. Stage 2 must identify the A8P profile, and Stage 3 must publish its sets and members into Core.Database.
 4. When the export changes, replace that authority file, rebuild Stages 1-3, and retain the source date as provenance for the new authority baseline.

Use the available authoritative sources when evaluating Atari 8-bit coverage. Absence from No-Intro, TOSEC, Libretro, or MAME alone does not establish that a file is unknown, invalid, or corrupt.

- progetto-SNAPS DATs: <https://www.progettosnaps.net/dats/>

DAT sites provide authority metadata. They are not instructions to download ROMs, disc images, or copyrighted content.

WARNING: VERY EARLY MAME Implementation

MAME support in myOrganizer: CURATOR is a VERY EARLY, experimental implementation. It must NOT be relied upon as a proven means of organizing MAME sets. Continue using clrmame as the definitive MAME ROM manager.

MAME output uses SPLIT sets. Use clrmame to manage MAME collections:
<https://mamedev.emulab.it/clrmamepro/>

CURATOR accepts a content-validated MAME -listxml export for coin-operated Arcade ROM and CHD authority. The XML structure identifies the export; the filename does not need an MAME prefix.

- Output uses split sets: machine-owned ROMs remain with each machine, while parent, BIOS, and device dependencies remain separate.
- MAME internal CHD SHA-1 authority takes precedence over Redump-derived optical-track evidence.
- nodump entries are excluded; incomplete authority sets go under Incomplete Sets.
- General merged, non-merged, standalone, and non-Arcade MAME collection management remains outside CURATOR's scope. Supported MAME software-list XML supplies exact ROM and CHD authority; it does not replace MAME's dependency-aware collection management.

Use clrmame to organize, audit, and rebuild MAME sets. Do not rely on myOrganizer: CURATOR to establish that a MAME collection is complete or playable.

Prepare LaunchBox Platform XML Files

CURATOR uses LaunchBox platform XML files because CURATOR builds a LaunchBox-compliant output collection.

Place LaunchBox platform XML files here:

 platforms

To generate the XML files:

5. Install LaunchBox from <https://www.launchbox-app.com/download>.
6. Open LaunchBox and create or use a LaunchBox library.
7. Import at least one game for each platform CURATOR should recognize.
8. Use the actual LaunchBox platform name you want CURATOR to recognize.
9. Close LaunchBox.
10. Copy the XML files from:

 <LaunchBox install folder>\Data\Platforms\

File pattern: *.xml

11. Paste them into CURATOR's  platforms folder.

Keep the CURATOR-supplied routing stub XML files in  platforms. Do not delete small XML files just because they look empty or simple.

Start CURATOR

12. Open the CURATOR application folder.
13. Launch Curator.ps1 with PowerShell 7.
14. Read the startup notice.
15. Type the required acknowledgement.

Expected result:

- The CURATOR gateway opens.
- The gateway shows Source files, Organized output, and Working area paths.
- On a clean installation, database\CURATOR.db does not yet exist. The Gateway shows only P — Prepare DAR. Other menu choices are hidden and their keys are not accepted until the database exists.

Run Prepare DAR

DAT discovery reports completed DATs and the sets and members read for the current platform. Reference preparation reports its reuse check and summary loading. Checking unchanged sources may still require reading their contents.

From the first-run Gateway, choose P and confirm the preparation run. Prepare DAR runs only Stages 1–3 without requesting Clear or Append or preparing collection output:

- Prepare DAR

Expected result:

- CURATOR prepares platform and DAT authority.
- Stage 1 creates and initializes database\CURATOR.db and reads LaunchBox platform XML files.
- Stage 2 reads DAT files.
- Stage 3 commits the validated DAR authority revision to database\CURATOR.db.

The full menu becomes available when the Gateway refreshes and finds the database. Its appearance does not prove that all three preparation stages succeeded. If Prepare DAR fails, correct the reported problem and run P again; choose O — Organize only after Prepare DAR completes successfully. Stage 2 hashes every original DAT/XML and ingests only new files or files whose content hash changed. Application or mapping changes alone do not re-ingest unchanged files. Stages 1 and 3 retain their own source and policy checks. RunIDs identify execution and provenance; a new RunID alone does not invalidate retained authority.

Set Folders

After Prepare DAR succeeds and the full menu appears, choose F — Manage folders to review or set:

- Source files
- Organized output

- Working area

Use three different folders.

Recommended first test:

- Source files: a small test collection folder.
- Organized output: an empty output folder.
- Working area: an empty scratch folder.

The working area must be outside the application folder. Do not store permanent files there.

Choose 9 — Check setup after reviewing the folders and before starting Organize.

Run Organize

After Prepare DAR succeeds, run Organize. CURATOR uses the saved reference library; there is no separate DAR-file selection step. Stages 1–3 can refresh reference data without discarding collection progress. Starting Stage 4 clean warns before replacing that progress and requires Stages 5–10 to run again.

Traversed archive containers are not duplicated into needs_attention; their descendants remain independently eligible for match or review, preventing .zip.zip outputs.

- Organize

Expected result:

- CURATOR scans the source folder.
- CURATOR matches files against authority.
- CURATOR prepares the output plan.

Stage 10 processes copy-only outputs first, then single-file ZIPs (including one-member sets), multi-file ZIPs, and remaining conversions. Each ZIP group shows one completed-file count bar for outputs requiring creation. ZIP output uses native 7-Zip Normal compression, level 5, with a default ceiling of 24 archive workers and one native thread per archive.

- CURATOR writes the organized collection to the Organized output folder. Complete Atari Jaguar CD authority sets publish as ZIP. Stage 10 normally builds output from the frozen Stage 9 plan and files prepared in the Workspace. There is one deliberate exception: when CURATOR has already proven that a source CHD is valid and satisfies the accepted authority set, Stage 10 copies that CHD directly from the original configured Input folder instead of generating it again. Before copying, CURATOR verifies that the recorded CHD path agrees with that configured Input folder. This saves processing time and does not modify the source CHD, so keep the Input folder available until Stage 10 finishes. Other complete optical authority sets publish as newly created CHD files, regardless of source container type.
- CURATOR commits the permanent Official and Unofficial generation to Core.Database and publishes 00_Manifest\CURATOR_Collection_Manifest.json. Organizer receipt evidence proves which existing files Append may safely reuse. Preserve the database,

current collection manifest, and receipts; deleting or changing them can break trusted continuity and require supported recovery or a deliberate rebuild.

- CURATOR tells you to run Reporting for collection statistics.

If the Organized output folder already contains files, CURATOR asks how to proceed:

- Clear: remove existing output folder contents first.
- Append: preserve output and create missing targets. Discovery and retained database receipts identify unchanged targets without Stage 10 content hashing or archive inspection. Keep output unchanged during processing. Ordinary Append misses same-size external changes and changes after discovery; use Full physical collection audit after manual changes. Eligible incomplete-set ZIPs can be replaced when verified new members are added.
- Keep needs_attention files you want to reconsider; the folder itself is disposable. An exact file can belong to more than one authoritative set: CURATOR may use it in one complete official set while also reporting a different set that shares the file but is still missing other members. The duplicate-looking review copy therefore does not mean the official set is damaged, and it may help complete the other set later. For future runs, include the preserved needs_attention contents with newly acquired source material so CURATOR can test whether the combined input completes additional sets. needs_attention remains outside authoritative collection truth and Reporting excludes it from collection evidence.

For a first test, use an empty output folder or choose Clear only when the folder is safe to replace.

Run Reporting

After Organize completes, use:

- Reporting / Collection coverage report
- Full physical collection audit (V): hash every collection artifact and review differences before any authoritative manifest update.

Expected result:

- Choose Reporting to compare the saved collection with current DAT reference data. Official and unofficial output count toward collection coverage; needs_attention is excluded. If Organizer was interrupted, choose Resume. To organize another input collection after a completed run, run Stages 4–9 to prepare a new output plan. Stage 10 retains workspace contents until output creation and collection catalogue publication finish, then clears the contents once while preserving the workspace folder. If publication or final cleanup fails, choose Resume to finish the remaining work without creating a duplicate collection generation. Final cleanup announces the total file and folder count before deletion and completion afterward, without a cleanup progress counter. Execution.Workspace.CleanOnFinish must remain true.
- Reporting writes Reporting.json in the configured reporting folder.
- Reporting shows available company reports from the configured reporting folder.

Use Reporting for collection statistics. Stage 10 completion totals are current-run output totals, not final collection statistics. When the latest Organizer generation added permanent items, Reporting announces the exact new-item total and a platform breakdown after startup; zero additions produce no new-item announcement.

If files are manually added, changed, or removed outside CURATOR, use Full physical collection audit (V). It hashes every collection artifact and updates authoritative manifests only for differences the operator approves.

Quick Checklist

Before the first real run:

- PowerShell 7.6.1 or newer is installed.
- CURATOR is launched from the application folder.

📁 dependencies is complete.

📁 tools is complete.

📁 DATs contains DAT files.

📁 platforms contains LaunchBox platform XML files.

📁 platforms still contains CURATOR routing stubs.

- Source files, Organized output, and Working area are different folders.
- The first source folder is small.
- The output folder is empty or safe to Clear.

Common Problems

.cdi Files Remain Unmatched

No tested Dreamcast or Jaguar CD CDI has matched a loaded DAT as an original image or through decoded tracks. Successful conversion does not confirm DAT identity or playback compatibility. Keep the original CDI and review unmatched files and generated outputs; changing a filename does not establish a match.

Problem	Fix
config.json is missing	Restore the complete CURATOR application package.
Curator.Core.Primary.psm1 is missing	Restore the complete 📁 core folder.
Missing required library	Restore the complete 📁 dependencies folder.
The 7-Zip archive provider is missing	Restore the complete 📁 tools folder, including 7z.dll. CURATOR does not require 7z.exe.
Stage 1 reports no platform XML collection	Copy LaunchBox platform XML files into 📁 platforms and keep CURATOR routing stubs.

Problem	Fix
Stage 2 finds zero DAT files	Add DAT files under  DATs, then run Prepare DAR again.
Output folder is not empty	Use an empty output folder for testing, or choose Clear only when safe.
Reporting shows stale or missing results	Run Organize first, then run Reporting again.

First Success

A successful first setup means:

- Prepare DAR completes.
- Organize completes against a small source folder.
- Output appears in the Organized output folder.
- Reporting opens and reconciles the output; then expand in controlled steps.

Current collection behavior

The CUE-authority resolver accepts a corrected CUE descriptor only when every extracted media member matches one unambiguous authoritative set. This check does not replace general-purpose CHD integrity verification.

Complete Redump Sega Naomi disc sets can produce standalone CHD output using authoritative filenames. DAT-declared publication files are retained even when an image-related folder name is present. XML game identifiers remain separate from authoritative filenames.

Start collection stages through the Gateway. Clear or Append is selected before processing; Clear replaces prior output once. Resume retains the interrupted run identity and completed output without clearing again. Blocking completeness findings stop final output. Receipts are saved before generated workspace sources are removed.

Recognized catalogue categories and format folders are resolved independently. USA-inclusive sets stay at the platform root regardless of language tags; recognized alternate editions use Alternates. Original member directories remain inside ZIPs. Only a file named exactly _ with zero bytes is excluded as a placeholder.

For a bound single-member set, a missing release classification can use recognized DAT set tags such as Beta or Unlicensed. An already recognized member classification keeps precedence. Matching, payloads and set memberships are preserved.

CDI DAT matching remains unverified. No tested Dreamcast or Jaguar CD CDI has matched a loaded DAT, either as the original image or through decoded BIN tracks. CURATOR can attempt CDI decoding and preserve the original image, but successful conversion does not identify the game or confirm playback compatibility. Review unmatched originals and generated outputs before using them in your collection.

Completeness follows the selected DAT set. A DAT may describe one disk as a complete set, even when the game needs several disks. When every required member of that DAT set is found and matched, the set follows its normal destination rules.

Whole-game estimates are informational and do not determine output placement. They do not establish that a game is playable or that all editions, disks or companion files are present.

When required DAT members are missing, the available matched members remain under Incomplete Sets, grouped by their authoritative DAT set. Similar names do not combine different DAT sets. Append may add verified members to an eligible existing incomplete-set ZIP, but even a completed accumulated ZIP is not automatically moved or converted into the normal collection.

Append can add verified new members to an eligible existing ZIP in Incomplete Sets. CURATOR validates the retained archive and freezes a combined member recipe before replacing it; repeated members are skipped without rewriting the archive. Earlier contents, source history and set memberships are preserved. Completed accumulated sets remain in Incomplete Sets; automatic completion assessment and final relocation or conversion are not available.

WARNING: Preserve Incomplete Sets. Do not manually delete or move the Incomplete Sets folder or its contents. Later Append or Resume processing that needs a previously recorded incomplete-set archive will stop if that archive is missing. Recreating an empty folder is insufficient. Keep the folder and its contents in place until this limitation is addressed.

Choose Resume from the Gateway to continue the saved stage without clearing successful output. Ordinary output failures are recorded while later outputs continue; Resume retries unfinished work and preserves successful outputs. Catastrophic failures can stop processing. ZIP groups show one completed-file count bar; optical preparation uses one shared display. Collection clearing uses simple messages. Recognized cracked variants use the Cracked folder label for newly planned output; existing folders are not renamed.

Incomplete-set save details are recorded in DETAIL.log. A log file that remains unavailable can stop processing. Review the logs if cleanup reports that working files remain.

v2.5.0-beta.1 accepts existing v2.5.xx and v2.0.0-beta.6 databases at schema 26 and preserves their creation identity. This compatibility does not establish that every unfinished run can resume. Let the Gateway check recovery eligibility. Keep input and working files unchanged until processing finishes.

Stage 10 streams ZIP/Copy receipts without rechecking source content after planning or rereading fresh output. Existing Append and recovery checks remain.

Authoritative member names determine ZIP entries; configured package naming also uses authoritative set names. Naming is separate from duplicate equality. Existing CHD, CDI and RVZ formats retain their physical extensions; CHDs remain standalone and never belong inside ZIPs.

Original DAT definitions and set memberships remain preserved. Unidentified ordinary files go into folders named after their source extensions, such as A26, BIN and JPG, within needs_attention. Files without an extension remain under Unknown File. Unidentified optical discs keep their existing whole-disc review handling. Filename collisions receive numbered suffixes before the extension. Missing MD5 alone does not make a file unresolved.

Duplicate sets are compared by their complete member contents, including repeated-member counts: Size, CRC and SHA1, with MD5 compared when both declarations supply

it. A missing MD5 does not split otherwise matching content and cannot bridge conflicting supplied values. Names, member order, DAT filename, format, source, platform and ranking are excluded from equality. Configured source hierarchy selects one whole-set winner; distinct sets that share only some members remain separate. Preserve every original declaration and membership. Filename comparison is case agnostic.

Unmatched review plans bind their hashes to the selected physical source, including whole CHDs and raw CDI tracks. These physical hashes are separate from normalized matching identity and from final output SHA-256 receipts.

Organizer can publish an empty permanent collection manifest. Standard Reporting and FullAudit accept empty collections and continue to report populated collections. Physical file counts and fulfilled-definition counts can differ because one physical file can retain several memberships.

RVZ creation and recovery of converted media have limited support. MAME support is very early and experimental; use clrmame to manage MAME collections. MAME output uses SPLIT sets.

Appendix A: Credits, Sources, and Third-Party Attributions

CURATOR was created, architected, and is managed by Miguel A. Ortiz. Copyright © 2026 Miguel A. Ortiz. Development, coding, and documentation assistance was provided by OpenAI ChatGPT; that assistance does not transfer project ownership or third-party rights.

Bundled components remain the work of their respective authors and contributors and remain governed by their own licenses. The concise credits below supplement—not replace—the controlling upstream notices and license texts.

Component	Creator / contributor credit	License and authoritative source
Aaru managed libraries	Natalia Portillo and Aaru contributors; CURATOR modifications: Miguel A. Ortiz, with OpenAI ChatGPT assistance	MIT (CommonTypes); LGPL-2.1-only (other bundled Aaru assemblies); https://www.aaru.app/
AlphaFS	Peter Palotas, Jeffrey Jangli, Alexandr Normuradov, and contributors	MIT; https://github.com/alphaleonis/AlphaFS
HashLib4CSharp	Ugochukwu Mmaduekwe and contributors	MIT; https://www.nuget.org/packages/HashLib4CSharp
Microsoft libraries	Microsoft and .NET Foundation contributors	MIT; Microsoft.Bcl.TimeProvider and Microsoft.Data.Sqlite
Newtonsoft libraries	James Newton-King and contributors	Json.NET: MIT; Json.NET Schema: AGPL-3.0-only or commercial license; https://www.newtonsoft.com/
SQLitePCLRaw / SQLite	Eric Sink, SourceGear, LLC, Microsoft Open Technologies, Inc., and contributors; SQLite authors	SQLitePCLRaw: Apache-2.0; SQLite: public domain; https://github.com/ericsink/SQLitePCL.raw ; https://sqlite.org/
Spectre.Console	Patrik Svensson, Phil Scott, Nils Andresen, Cédric Luthi, and contributors	MIT; https://github.com/spectreconsole/spectre.console
7-Zip	Igor Pavlov; retained notices also credit Apple Inc., Facebook, Inc., Yann Collet, and Alexander Roshal (unRAR)	LGPL-2.1-or-later plus unRAR restriction and BSD components; https://www.7-zip.org/license.txt
BinMerge	Chris Putnam and contributors	GPL-2.0-or-later; https://github.com/putnam/binmerge
Modified MAME chdman	MAMEDev and contributors; CURATOR modification copyright 2026 Miguel A. Ortiz	GPL-2.0-or-later; https://docs.mamedev.org/license.html
DolphinTool	Dolphin Emulator project contributors	GPL-2.0-or-later; https://dolphin-emu.org/
Redump cue/GDI metadata	Redump project contributors	Public-domain metadata; https://redump.org/
verifydump	j68k; embedded Python, PyInstaller, OpenSSL and Microsoft contributors	Version 1.0.0; MIT plus incorporated runtime licenses; see LICENSE

Complete packaged records:  **LICENSE** and  **config\dependency-manifest.json**. The dependency manifest records the exact packaged versions, hashes, provenance, and license identifiers.

Authority and data-source credit: No-Intro / DAT-o-MATIC, Redump, TOSEC, Libretro Database, MAME, progetto-SNAPS, LaunchBox, and other operator-selected authorities

remain credited to their respective projects and contributors. Their names and trademarks belong to their respective owners; mention does not imply affiliation, sponsorship, or endorsement.

Modified chdman and source availability

CURATOR bundles tools\chdman.exe as the unofficial modified MAME chdman build 0.289-CURATOR.8-poc. It retains standard chdman commands and adds logical-track hashing. Scan Preparation decodes and hashes logical tracks without writing BIN files and retains bounded header prefixes for Matching. Applicable normalization rules and pending output operations still extract tracks through the configured Core tools. Collection Assembly uses recorded relational source evidence without probing or rehashing collection media. Copying an unchanged whole CHD does not require track extraction. Acceptance is bounded to the documented collection; it does not establish universal compatibility or playback correctness. When recovery supplies an authoritative corrected CUE, Stage 9 reconciles it against the complete originating CHD membership and suppresses only obsolete recovered-CUE review records. Genuinely unresolved source evidence remains in NeedsAttention.

The build is GPL-2.0-or-later, credits MAMEDev and contributors, and is not produced, sponsored, approved, supported, or warranted by MAMEdev. MAME is a registered trademark of Gregory Ember.

This v2.5.0-beta.1 release provides matching application, modified chdman, modified Aaru.Images and third-party source ZIPs, a release inventory and SHA-256 sidecars. Provide all source companions beside the application download with equivalent access. Packaged Input, Output and Workspace defaults are exactly C:\; configure appropriate destinations before use. CleanOnFinish remains true and schema remains 26.

Release dependency records identify verifydump 1.0.0, unofficial Aaru.Images 5.4.1-curator-poc1 and unofficial chdman 0.289-CURATOR.8-poc. Json.NET Schema 4.0.1 is supplied under its published AGPL alternative (publisher limit: 1,000 validations per hour). See LICENSE for exact source assets and original terms. CDI DAT matching remains unverified. This beta does not establish universal acceptance of source formats, interruption recovery or ST/GameBase inputs. Review results and preserve originals.