USER GUIDE

CURTAIN CALL

Authoring, rehearsing and running theatre audio

v1.1.2 (build 20261117)

Developed by YZONK

https://curtaincall.yzonk.com

Latest online guide

https://curtaincall.yzonk.com/userguide/latest.php

Scan or click to download the most recent PDF.

Curtain Call user guide

Contents

1 About Curtain Call: read this first **9**
1.1 How it was made: entirely vibecoded 10
1.2 What's new in v1.1.2 11
1.3 What's new in v1.1.0 12
1.4 What's new in v1.0.0 13
1.5 Licence: free to use, share and modify 15
1.6 Tools used to build it 16
1.7 Where it came from: sources of inspiration 17
1.8 Send ideas, problems and questions 18
1.8.1 Persistent errors and unexpected shutdown 20
1.9 Author, version and credits 21
1.10 Checking for updates 22
1.11 Checking local models for updates 24
2 Projects, installation and platform reference **25**
2.1 Requirements and one-time setup 25
2.2 Machine requirements: with and without local models 26
2.3 Untested operating systems and feedback reminders 27
2.4 Install with pip instead 28
2.5 Portable mode and first start 29
2.6 Launch and associate project files 30
2.7 The project file and separate exports 31
2.8 User/computer data lives elsewhere 32
2.9 Rebuild the demo and documentation as a developer 33
3 Quick start: a doorbell cue **35**
3.1 Build a safe first test 35
3.2 Know which mode owns the change 36
3.3 The whole workflow at a glance 37
3.4 First ten minutes with The Lighthouse Keeper 38
3.5 Explore the demonstration examples 39
4 Using the Curtain Call User Manual **40**
4.1 Maximise, print and export the guide 41
4.2 Numbered sections and readable printed pages 43
4.3 Hover help on controls 43
4.4 Dialog actions and keyboard navigation 44
4.5 The boxes you type into 45

4.6 Tip of the day 47
4.7 The guided tour: first steps, pointed at 48
4.8 Read the visual language 50
4.9 What search does and does not do 51
5 Projects, saving and recovery 52
5.1 One production, venues, rehearsals, and iterations 54
5.2 Compare show revisions 55
5.3 Unavailable recent projects 56
5.4 Create and prepare a production 57
5.5 Identify projects at a glance with a thumbnail image 57
5.6 Save, back up and move a project 57
5.7 Carry fonts with the project 59
5.8 Approve a project version upgrade 60
5.9 Clean up a project 61
5.10 Recover after an interrupted session 64
5.11 Understand three different kinds of state 65
6 Audio description: character texts and narration 66
6.1 Keep a saved character register 68
6.2 Write and translate character narration 69
6.3 Generate, import and review narration audio 71
6.4 Navigate and export one character's lines 77
7 Script writing and cue links 78
7.1 Inspect and follow gutter links 79
7.2 Take the gutter markers with the script 79
7.3 Write a scene with meaningful elements 80
7.4 Continue a multi-paragraph speech 80
7.5 Import, check and revise 81
7.6 Place a cue precisely 81
7.7 Identify the Script screen 82
7.8 Start with a template, not blank formatting 83
7.9 Separate text location from cue execution 84
8 Script revisions: approve changed anchors 85
8.1 Review every ambiguous destination 86
9 Library: media files, not playback settings 87
9.1 Import and audition media files 87
9.2 Same-name import choices 88
9.3 Process audio and restore versions 89
9.4 Enlarge, scroll and audition the Library waveform 91
9.5 Named audio markers 93

9.6 Link markers to script passages 95
9.7 Repair missing audio without rebuilding cues 95
9.8 Identify the Library screen 96
9.9 Metadata is show documentation 98
9.10 Media and performance rights reports 99
9.11 Required metadata: make the library fill itself in 100
9.12 Metadata profiles: type it once, reuse it everywhere 101
10 Cue Creator: fades, ducking and waits 102
10.1 A wider track waveform 102
10.2 Reach cue properties in a short pane 103
10.3 Choose the instruction, then its target 104
10.4 Placeholders: reserve the number before the cue exists 105
10.5 Pre-wait, post-wait and continuation 106
10.5.1 Worked timing examples 106
10.6 Ducking: let a foreground recording lower the background 108
10.7 Identify the Cue Creator screen 109
10.8 Build, reorder and reuse instructions 110
10.9 See the actual fade shapes 111
10.10 Understand counted and seamless loops 112
10.11 Draw the timing before running a chain 113
10.12 Visualise foreground ducking 114
11 Show Player and the live script 115
11.1 Before the audience enters 116
11.2 Project health and audio checks 117
11.3 During the run 118
11.4 Pause and carry on 119
11.5 Identify the performance screen 120
11.6 Read state words, not just colours 121
11.7 End of show 121
11.8 The playhead follows its cue through edits 122
11.9 Use Now playing without rewriting the production 123
11.10 Prepare a new performance 123
11.11 Screen lock: cover the screen, keep the show playing 124
12 Rehearsals: Drop pin versus Drop note 126
12.1 Keep notes usable in a short window 126
12.2 A practical correction cycle 128
12.3 Identify the Rehearsal Notes screen 129
12.4 Name each rehearsal and its iteration label 131
12.5 Check reference context when dropping a note 132

13 Rehearsal change digest: compare snapshots **133**
13.1 Establish an honest baseline 134
13.2 Interpret the comparison as facts 135
14 Flight recorder and run timeline **136**
14.1 Separate event evidence from audio recording 137
14.2 Compare the same dramatic passage 138
15 Scene recipes: reusable cue patterns **139**
15.1 Choose a built-in by the operator sequence 140
15.2 Save a user pattern responsibly 141
16 What-if simulator: rehearse silently **142**
16.1 Ask one bounded question 143
16.2 Read uncertainty explicitly 144
17 Show Score: the whole evening on one page **145**
17.1 Choose the timing 146
17.2 Zoom and export 147
18 Sound captions and accessible operation **148**
18.1 Write useful sound intent 148
18.2 Pick colours and keep reusable palettes 149
18.2.1 Choose a shade visually 149
18.3 Arrange the audience-caption table 152
18.4 Choose the second screen panel size 153
18.5 Show translated subtitles to the audience 154
18.5.1 Subtitle spacing, boxes and readability 156
18.6 The audience screen and its narrow mirror 160
18.7 Adjust the booth view before the performance 160
18.8 Subtitles: turn the dialogue into a title track 161
18.8.1 Time titles against a reference video 163
18.8.2 Build, time and export titles 167
18.9 Time subtitles during rehearsal 169
18.9.1 Reading rules and saved title data 170
18.9.2 Translate and export a subtitle language 170
18.10 Caption and subtitle look: separate sections, three levels 176
18.10.1 Typography controls and the sample 176
18.10.2 Following and overriding the caption look 178
18.11 Hard of Hearing subtitles: enhance accessibility for deaf and hard-of-hearing audiences 179
18.11.1 Show speaker names and character colour coding 185
18.11.2 Describe feelings with expressive typography 186
18.11.3 Add sound descriptions and context 186
18.11.4 Customize one subtitle's Hard of Hearing look 186

18.11.5 Export subtitles with Hard of Hearing features 187
18.11.6 Hard of Hearing configuration fields 187
19 Venue profiles: take the show on tour 189
19.1 Libraries: create and edit physical venues 190
19.1.1 Bundled venues and your own source checks 190
19.1.2 Copy a venue into a project 192
19.1.3 Venue JSON structure 193
19.1.4 Migrating older local libraries 194
19.2 Libraries: video projectors and lighting fixtures 195
19.2.1 Venue projector installations 195
19.2.2 Factory video projector examples 196
19.2.3 Equipment fields, units and unknown values 197
19.2.4 Manufacturer-checked starter equipment 199
19.2.5 Local equipment JSON 199
19.3 Audio gain profiles are separate 200
19.4 Keep room setup separate from creative cue design 202
19.5 Gain is not automatic calibration 203
20 Show-day checklist and safe resumption 204
20.1 Before the house opens 205
20.2 During an interrupted rehearsal 206
21 Exports and printing 207
21.1 Export all formats and share a clean show 208
21.2 After every export: open, show or email 212
21.3 Common PDF page furniture 212
21.4 Complete export format reference 213
21.5 Choose a paper workflow 214
21.6 Distribute without losing the editable show 215
22 Output log and troubleshooting 216
22.1 Diagnose one layer at a time 217
22.2 Keep evidence, not just the visible filter 218
22.3 Understand the output controls 219
22.4 Respect the playback limitations 219
23 Frequently asked questions 220
23.1 Does opening another mode stop audio? 220
23.2 Why does GO skip a row? 220
23.3 Why did a follower play during a fade? 220
23.4 Can a script line trigger itself automatically? 220
23.5 Why is the background still ducked after muting the announcement? 220
23.6 Can I share only a .ccshow file or exports ZIP? 221

23.7 Why are my user recipes missing on another computer? 221
23.8 Does a matching venue device name route audio there? 221
23.9 Will PDF import read a scan? 221
23.10 Will exported paperwork change after I edit the show? 222
24 Keyboard reference **223**
24.1 Editing and navigation shortcuts 224
25 Cue field reference **226**
25.1 Every cue field 227
25.2 Read defaults without accidentally rewriting a cue 231
26 Library fields and project settings reference **232**
26.1 Every library field 233
26.2 Every project setting 234
26.3 Apply or save app defaults deliberately 237
27 Profiles **238**
28 Preferences reference **242**
28.1 Reset one preference 243
28.2 Defaults for new projects 244
28.3 Which settings have app defaults and project overrides? 245
28.4 Every app preference 246
28.5 Every New projects default 253
28.6 Keep preferences separate from the production 255
29 Menus and toolbar reference **256**
29.1 Every main menu item 257
29.2 Script toolbar and panels 261
29.3 Library toolbar and detail controls 262
29.4 Cue Creator toolbar and script panel 263
29.5 Show Player controls 264
29.6 Rehearsal Notes toolbar 265
29.7 Shared table, output and Help controls 266
30 Glossary **267**
30.1 A 268
30.2 C 269
30.3 D 271
30.4 E 273
30.5 F 274
30.6 G 275
30.7 H 276
30.8 I 276
30.9 J 276

30.10 L 277
30.11 M 278
30.12 N 278
30.13 O 280
30.14 P 281
30.15 Q 282
30.16 R 283
30.17 S 284
30.18 T 285
30.19 U 286
30.20 V 287
30.21 W 287
30.22 X 288
30.23 O 288
30.24 P 288
30.25 V 288
30.26 A work in progress: where Curtain Call may go next 289
30.26.1 Platform testing and local-model verification 289
30.26.2 Shared venue and equipment libraries 289
30.26.3 Further audience access ideas 290
30.26.4 A show rendered for rehearsal and review 290
30.26.5 Script-aware assistance with human approval 291
30.26.6 Reliability and show-day confidence through rehearsal tools 291
30.26.7 More expressive audio and projection 291
30.26.8 Shared context without surrendering control 292

A complete, Curtain Call User Manual for authoring, rehearsing and running theatre audio. Choose a chapter in the index or search the whole manual. Start with the About chapter and installation instructions before the quick start. The modes are explained in activity-bar order; rehearsal tools follow, then show-day procedures and reference tables. This is one Markdown document, bundled with its illustrations. The full PDF's cover includes a clickable URL and QR code for the latest online guide. The printed copy stays usable offline; opening or scanning the link needs internet access.

Worked examples use **The Lighthouse Keeper**, the supplied demonstration project. Use a copied project when experimenting. Sound-check examples play audio only when you deliberately use preview, Test fire, a hotkey or GO.

1 About Curtain Call: read this first

Curtain Call is a desktop application for designing, rehearsing and running theatre sound. It keeps the **script**, the **audio library**, the **cue list**, the locked **Show Player** and the **rehearsal notes** of one production in a single portable project file, so the person who writes the cues and the person who presses GO work from the same source.

> **Show preparation:** Rehearse with the computer, audio interface, speakers and audience displays you will use for the performance. Keep a backup playback method ready and run the project's preflight checks before the audience arrives.

1.1 How it was made: entirely vibecoded

This application was **entirely vibecoded**: its source code, tests, demo project, diagrams and screenshot generator were written by AI coding agents, directed in plain language by the author, who described what a theatre operator needs, tried the results and asked for corrections. **This manual was vibecoded too**: AI agents wrote it from the code, the completed-feature list and the author's requests, and generated its illustrations from the demo project.

For the author, this was also a personal challenge: to explore how AI-assisted creation works and how far one person can go in building an application from scratch by describing ideas, making decisions, testing the results and improving them. The desktop app was inspired by the earlier Curtain Call web version, which the author created by hand; that experience helped shape the needs and ideas behind this new version.

The hope is that AI can empower more people to create tools for their own needs, even when a project might otherwise be beyond their available time or resources. Sharing those tools, along with feedback and what we learn from using them, can help make productions better and strengthen the community around them. AI is an aid, not a substitute for human judgement, testing or the people who use the tools in a real theatre.

What this means for you:

Topic	What to expect
Workflow	Write, rehearse and run a production from one portable project file.
Show preparation	Check media and cues, rehearse with the intended equipment, and keep a backup ready.
Platforms	Linux and Windows 11 workflows and clean installation have been checked by the author. Platform testing details are in the WIP section.
Help	Use the searchable offline manual for instructions and Help > Send feedback or ideas for questions and suggestions.
Your data	Single-file projects are ZIP archives containing JSON and ordinary media files. Folder projects remain supported.

Automated release tests: The latest offscreen Linux source regression passed all **2,080 tests** with no skips (Python 3.15.2, PySide6 6.12.0). The compiled app, demo and PDF are checked separately. Automated checks do not replace testing on the intended show equipment.

Tip: Help > About Curtain Call shows the exact version and build you are running. Quote it in any report; every PDF export prints it in its footer too.

Example: For a first rehearsal, copy **The Lighthouse Keeper**, run the whole show on the theatre's computer and speakers, fire every cue once with GO, try Stop all and Panic, and capture the run with **Record** in the Show Player header. Review the recording and notes with your team.

1.2 What's new in v1.1.2

This patch, **v1.1.2 (build 20261117)**, adds visible draggable corner and edge boxes to subtitle regions, two-axis region movement and keyboard resizing. Standalone builds now include trusted HTTPS CA certificates for demo, font and optional model downloads, while retaining certificate and hostname verification. The demo remains an online-only download, not part of the app.

1.3 What's new in v1.1.0

This release, **v1.1.0 (build 20261116)**, makes installed copies self-updating:

- **One update path:** the app receives every update and patch from its Velopack feed. **Help > Check for updates** offers the patch, downloads only the changed files and restarts into the new version; a copy that was not installed from a packaged release does not contact a version service and simply says so.
- **Update schedule:** *Preferences* > *Interface* > *Updates* limits how often a startup check runs and how many days a fresh patch waits before it is offered.
- **Dialogs:** resizable dialogs remember their size and where you put them, small messages keep a fixed size, and long text such as **About Curtain Call** scrolls vertically instead of growing past the screen.
- **Safer project folders:** a new project suggests `Documents/Curtain Call Projects`, checks that the folder can really be written, and no longer fails with a "Bad file descriptor" error on Windows drives or shares that refuse hard links.
- **Icons and file type:** packaged copies register the app icon and open `.ccshow` files on double-click for the current user; portable copies stay untouched.

The v1.0.0 notes below still describe the feature set of this build.

1.4 What's new in v1.0.0

Version 1.0.0 is the first finalized desktop release after the alpha builds. This edition, **v1.0.0 (build 20261114)**, brings these workflows together:

- **Complete exports:** deliver applicable formats, authored languages, reports and reusable profiles in one operation, with a checksum manifest.
- **Portable working projects:** export an essentials-only native copy with its script, required audio and selected AD recordings, attached video, thumbnails, effective settings, profiles and permitted fonts. Media keeps its original filenames; unused assets, history, logs and caches stay out of the copy.
- **Portable fonts:** permitted licences need no confirmation. One permission checkbox covers all unconfirmed font rights for the export; restricted fonts are never copied.
- **Missing-file correction:** a failed export identifies required missing media and opens the appropriate editor so you can correct the assignment.
- **Audio Description mode (Ctrl+9):** prepare character descriptions, manage selected recordings and languages, and generate speech with optional local engines installed separately under `local-models/`.
- **HoH Accessibility mode (Ctrl+8):** edit project character colours, feeling styles and descriptions. Use **Ctrl+H** to toggle HoH styling; audience movement stays off until you deliberately enable it.
- **Shared HoH profiles:** separate Feelings and General settings for individual styles and profile-wide accessibility and animation choices.
- **Audience guide:** export the project's speaker and feeling legend for the team and audience.
- **Reference-video subtitle timing:** record title start times while watching the video, overwrite timings, and pause, seek and resume in the detached workspace.
- **Independent subtitle playback:** use **Subtitles GO (F9)** for the title clock, separately from audio GO. Audience output and the Show Player mirror share pause-aware timing.
- **Smoother desktop operation:** fixes for PDF-export and window-close freezes, safer media/window lifetimes, and consistent dialog actions.
- **Refreshed manual:** current HoH screenshots and a reading-first audience-access workflow.

The matching PDF for v1.0.0 is labelled **v1.0.0, build 20261114** and includes a cover URL and QR code

for the latest online guide. GPL-3.0-or-later licensing and source-attributed venue references are included in this release.

The Linux archive is compiled on **Fedora 45 with glibc 2.44**, Python 3.15.2 and PySide6 6.12.0. Older Linux systems may need the source installation instead. This compilation does not certify native Windows/macOS packages or every audio/display device. Check the version and build in **Help > About Curtain Call**, then rehearse on the intended show computer.

1.5 Licence: free to use, share and modify

Copyright (C) 2026 YZONK. Curtain Call's original desktop application source, documentation and original application assets are licensed under the **GNU General Public License, version 3 or (at your option) any later version** (`GPL-3.0-or-later`).

- You may use the app freely, including for commercial and professional shows.
- You may share and modify it. Preserve applicable copyright, licence and attribution notices, and mark modified versions and their changes as the GPL requires. When distributing the app, provide its corresponding source as required by the GPL; distributed modified versions must remain under the GPL.
- Private modifications do not have to be published. Contacting the author is not required; feedback and feature requests are welcome through **Help > Send feedback or ideas**.
- The app is supplied without warranty, to the extent permitted by law.
- Your own scripts, recordings, subtitles, projects and exports do not become GPL-licensed merely because you create or run them with Curtain Call.
- Third-party software, reference data and demo media retain their own terms. The app licence does not grant blanket permission to reuse those materials.

Read the full GPL text, also available from **Help > About Curtain Call**, or the GNU website. Source distributions include the application notice in `LICENSE` and separate third-party information in `THIRD_PARTY_NOTICES.md`.

1.6 Tools used to build it

Tool	Used for
Visual Studio Code	The editor in which the project was written; it also inspired the look (activity bar, dark theme, status bar).
GitHub Copilot agents (Copilot SDK) with large language models	Writing and reviewing all code, tests, documentation and this manual from the author's instructions.
OpenRouter with DeepSeek	AI-assisted development alongside Copilot.
Ollama with local models	Exploring architectural ideas and decisions.
Python 3	The programming language of the whole application.
PySide6 (Qt 6)	Windows, widgets, text documents, PDF writing and printing.
Qt Multimedia with its FFmpeg backend	Playing and decoding audio (MP3 and WAV are covered by automated tests; FLAC, OGG and M4A depend on the platform's FFmpeg build), and drawing waveforms.
Python standard library	Reading ODT and DOCX scripts, JSON project files, the unittest test suite.
pypdf (optional)	Importing selectable text from PDF scripts, when the `pdf-import` extra is installed.
Segno	Generating the latest-online-guide QR code locally for the PDF cover.
Poppler tools (`pdftoppm`)	Checking exported PDFs page by page during development.

1.7 Where it came from: sources of inspiration

Source	What it gave Curtain Call
Curtain Call for the web (curtaincall.yzonk.com)	The author's earlier browser version and the direct predecessor: the idea and demo show. Projects can still be exported to its format (**Project > Export > Web version**).
Stage and screenplay writing conventions	The script elements (scene, action, character, parenthetical, dialogue, transition) and the Enter / Tab rules of the Script mode.
Visual Studio Code	The dark, calm interface: activity bar for the modes, flat toolbars, small-caps section headers, coloured status bar.
Theatre practice	Rehearsal notes by department, show-day checklists, panic and stop-all, audience captions for deaf and hard-of-hearing people, and venue profiles for touring.
QLab	The reference for how a cue-based playback tool feels to an operator: a numbered cue list, a single GO, waits and fades on every cue. Curtain Call is an independent program; `.resources/qlab` in the source holds an exported QLab file used only as an import sample.

1.8 Send ideas, problems and questions

Choose **Help > Send feedback or ideas...** (also a button in **Help > About Curtain Call**; no project needs to be open). Pick **Bug**, **Idea**, **Help**, **Praise** or **Other**. Choose an optional subtype, enter your email, a summary and a message. For Bug, open **Reproduce**: describe the result you expected and list the exact starting state and actions under **Repeat steps** so YZONK can reproduce the issue. Review these details before sending: the report, contact email and visible diagnostics. Choose **Send** to submit the report and listed files to **YZONK**. Required fields are checked first; an error identifies and focuses the field. This source-version feature requires the server API to be deployed; older published application downloads retain their original feedback workflow.

Control	What it does
Type	Bug, Idea, Help, Praise or Other; the main grid never shifts when switching
Subtype	Optional detail, default General; listed below
Email	Required private address for replies; automated release emails are not yet implemented
Reproduce	Bug-only Expected and Repeat steps fields; switching types preserves the draft but excludes these fields from non-bug submissions
Include system details	Adds app version/build, OS, architecture, Python and Qt; excludes project names/paths and files
Include logs	Optional; checked for Bug, unchecked for other types. Unchecking removes logs from the files-to-send list; rechecking restores the prepared snapshots
Add logs...	Adds reviewed `.log`/`.txt` snapshots: at most 3 files, 1 MiB each and 2 MiB total, including prepared session/pending diagnostics; explicitly enables log inclusion
Add pictures...	Adds PNG/JPEG screenshots: 3 MiB each, up to 24 million pixels; pictures stay included when logs are unchecked
Remove	Removes the selected attachment from this report, never from disk
Preview	Opens the selected log or screenshot only on request; no inline log preview
Send	Validates the fields, checks possible duplicates and submits the report, private email and listed files; no separate consent checkbox
Copy tracking ID	Copies the receipt after the server confirms storage
Visit website	Opens curtaincall.yzonk.com

Example: In **The Lighthouse Keeper**, the storm fade in cue 6 sounds too abrupt on your speakers. Choose **Bug**, subtype **Playback**, summary "Storm fade feels abrupt", describe what you heard, and fill in Expected and Steps under Reproduce. Review the logs and system details under Diagnostics before approving.

The service returns a unique tracking ID after storing the report. Save it for correspondence. The receipt thanks you and confirms your tracking number. Feedback may help a future release; a fix, feature or reply is not guaranteed. Automatic email notifications are deferred to a future backoffice workflow; entering your email does not subscribe you to a mailing list. Possible duplicates show only titles approved for publication, never other users' private reports or email addresses. If submission fails, your text stays in the open dialog; retry unchanged text there to avoid creating a second record. Drafts are not saved when the dialog closes. There is no email fallback or automatic upload. If possible duplicates appear, nothing has been submitted yet. Review the matches and click **Send** again only if your report adds useful information.

Type, Subtype, Summary, Message and Email occupy a fixed grid. **Reproduce** and **Diagnostics** use fixed-height tabs below it; changing the type does not move the main inputs. Reproduce is available only for Bug.

Include technical report is a separate option in Diagnostics, initially off even for Bug. Selecting it collects a local snapshot of display, audio output and installation/runtime facts and opens a review window. Read it and press **Include report** to approve, or **Cancel** to exclude it. Nothing leaves the computer until you press **Send** in the feedback dialog.

The report lists screen count, logical resolution, scaling/DPI and the Qt platform; audio-output names, default status and Qt-reported sample-rate, channel and format capabilities; and packaged/source status plus runtime versions. No device is opened and no test sound is played. Reported capabilities do not establish working playback. Audio-device names may contain personal information: decline the report if you do not want those names shared. Project contents, file paths, usernames, hostnames, IP addresses and device identifiers are not collected.

The approved **curtaincall-technical-report.txt** appears in the files-to-send list independently of **Include logs**. Use **Preview** to read it again, uncheck **Include technical report** to exclude it, or **Remove** to discard the snapshot. It counts toward the normal text-log and attachment limits. Unchanged retries send the same approved bytes; close/reopen the feedback dialog to collect fresh facts. Collection errors leave the option off and are shown explicitly.

Type	Subtypes
Bug	General, Crash, Playback, Performance, Compatibility
Idea	General, Feature, Usability, Accessibility
Help	General, Setup, Workflow
Praise / Other	General

Logs and system details default on for Bug and off for other types. They are always optional: no idea, question or compliment requires a log. You can explicitly include diagnostics for any type if useful. Opening the panel automatically prepares **curtaincall-current-session.log**, a snapshot of this app session. Preparation does not imply inclusion: review it under Diagnostics, remove it or leave Include logs unchecked, then click Send. Older entries may be omitted with an explicit notice to fit 1 MiB. No unrelated system logs are collected.

Logs may contain project names, file paths, operator details and other private information even when system details are unticked. Remove a sensitive snapshot, or redact an exported log locally and attach that file instead. Adding files updates the files-to-send list. Review that list before clicking Send. Files are snapshots: editing a file after attaching it does not update its preview or upload. Remove and reattach it after redaction. The server stores reports and logs privately together; failure retains the open form and unchanged retry identity. Local session history, exported logs and manually selected files are never cleared or deleted after submission. App-managed diagnostic logs follow the selective cleanup policy below. The full list is limited to **5 files / 8 MiB total**; logs retain their smaller limits. Screenshots can also contain private data: crop/redact before attaching.

1.8.1 Persistent errors and unexpected shutdown

Curtain Call keeps warning/error messages and Python exception tracebacks in **errors.log** inside a **diagnostics** folder beside its preferences file. Native crashes can also leave a separate crash-stack log. These files survive restarts; they are not recordings or a complete previous session history. The feedback panel prepares **curtaincall-pending-diagnostics.log** when these logs contain data, alongside the current-session snapshot. Both count toward the existing log limits. Preview or remove either before sending. If pending diagnostics exceed 1 MiB, only a bounded prefix is prepared; the unsent remainder stays local for a later report.

After an **Unexpected shutdown**, the next primary app launch offers to review a Bug / Crash report. Forced quits and power loss can also trigger this notice; it is not proof of a software bug. Declining sends nothing. Accepting opens the normal feedback form, not an upload: enter your email, the

expected result and the actions before the shutdown (or say you do not know), then review and choose Send. The recovery review comes before other startup notices and tours.

Only after a confirmed server receipt, and only if the exact prepared pending snapshot was included, are its matching bytes removed from the app-managed diagnostic logs. New errors written during upload and any unsent remainder stay local. Unchecked/removed logs, failed requests and uncertain receipts are not cleared. Changed local logs are kept, and cleanup errors are shown alongside the successful tracking receipt. This never changes your session history, exports or manually selected files.

1.9 Author, version and credits

Curtain Call is made by **YZONK** (YZONK.com, curtaincall@yzonk.com). The demo music is by **Zolvi** via Free To Use; the phone announcement was created by **YZONK** with NaturalReader. See `THIRD_PARTY_NOTICES.md` for the track list and source details. The app icon and banner were made for the web version. **Help > About Curtain Call** also lists the supported audio formats and platforms.

1.10 Checking for updates

Choose **Help > Check for updates** to contact the Curtain Call website immediately. The website receives your network address; the app does not send project data. The check compares both version and build and opens the update dialog when a newer version is advertised. Results are deferred while Show Player is open or audio is playing or paused. A failed check is reported; try again when online.

On compiled Linux x86-64 installations that support it, the update dialog can download and install the update for you. Nothing is downloaded until you choose **Download and install**; the download shows its progress in megabytes and can be cancelled. Every update is checked against its publisher signature before and after downloading, and installation replaces only the app's own verified files: your projects, preferences, profiles, portable data and local models are never touched, and the previous version is kept in a backup folder beside the installation. Before installation, Curtain Call checks for playing audio and unsaved changes and closes the main window; cancelling keeps the current installation unchanged. If installation fails, the error is shown and the previous version is restored where possible. After installing, start Curtain Call again to use the new version.

Installations created by the official Windows, macOS and AppImage packages are updated by Velopack instead. Those builds download only the files that changed (a patch) rather than the whole application, show their progress as a percentage, and then close, apply the patch and start again on their own. When no patch is available for your version, the full package is downloaded instead.

On other installations, or when an update is not signed for automatic installation, the dialog offers the release page instead. Follow the release page's manual installation instructions and keep backups of your projects and profiles.

To check automatically, enable **Check for updates at startup** in **Edit > Preferences > Interface > Updates**. This option is off by default. When enabled, each startup contacts the website, which receives your network address; no project data is sent. An up-to-date result does not interrupt you. Errors and available updates are shown after any open modal dialog, Show Player or active/paused audio. The setting is stored as `check_updates_on_startup` in preferences.

Two numbers next to it decide when checking and installing happen. **Check at most once every (days)** limits how often a startup check may contact the feed: 0 checks at every start, and 7 means at most once a week. **Wait before offering a patch (days)** holds a new patch back after this computer first sees it, so an unknown bug can be reported before a show uses it: 0 offers the update as soon as it is published, while 5 waits five days. The wait is counted from the day the update was first seen

here, not from its publication date, and a patch release still asks before anything is downloaded or installed. A manual check while a patch is held back explains when it will be offered. Both values are stored as `update_check_interval_days` and `update_install_delay_days`. The schedule keeps three bookkeeping values in preferences: `last_update_check` records the day of the last completed check, and a held-back update is remembered by its version in `pending_update_key` together with the day this computer first saw it in `pending_update_first_seen`. They maintain themselves; there is nothing to edit.

Packaged copies also refresh the application icon and the `.ccshow` file association for the current user at startup, so double-clicking a project file opens it in Curtain Call. This is stored as `register_file_associations` (enabled by default) and never touches portable copies, source checkouts or other users' settings.

1.11 Checking local models for updates

Choose **Edit > Check for model updates...** to compare your installed local models with their trusted catalogues. Installed translation models are compared with the official index over the network; the index host receives your network address, and no project text or audio is ever sent. AD voices are checked locally against this app's reviewed, revision-pinned catalogue without any network contact. Engine versions (Argos Translate, Piper) are pinned by each app release and change with application updates.

The dialog lists each installed model's state. An update is marked as needed only when the installed files are damaged or no longer match the reviewed catalogue, so the feature could stop working; a newer published translation model is optional and the installed version keeps working. Nothing is downloaded or replaced until you tick the updates you want and choose **Update selected**. Downloads show measured progress and can be cancelled; each working model is preserved until its replacement has been downloaded, integrity-checked and validated, and it is restored unchanged if an update fails. Failed updates explain the error and can simply be tried again.

To check automatically, enable **Check for local model updates at startup** in **Edit > Preferences > Interface**. This option is off by default. The startup check is quiet: it only opens the dialog when updates actually exist, after any open modal dialog, Show Player or active/paused audio. The setting is stored as `check_model_updates_on_startup` in preferences.

Creator of the MAILART.pt cultural project and CurtainCall free show-control app. Your support helps keep these open resources free, active, and continuously growing for everyone. Thank you for empowering independent tools! Support the creator.

Curtain Call is a one-person project maintained by YZONK when time allows. Maintenance and support are best-effort, with no promised release schedule, response times or support guarantees.

2 Projects, installation and platform reference

Example: Install Curtain Call on the touring laptop, save and copy **The Lighthouse Keeper.ccshow**, then open the copy. For a folder project, copy the complete folder instead. Do not use an exports ZIP as your only editable backup; preferences and user recipes are stored separately on that computer.

2.1 Requirements and one-time setup

You need **Python 3.10 or newer** and **PySide6 (Qt 6.8 or newer)**. The setup scripts install PySide6 and Segno into a private `.venv` environment. Initial dependency installation needs internet access; the bundled manual works offline afterwards. Curtain Call is developed and tested on Linux, and the author has tested its functionality on Windows 11. Detailed platform coverage is listed in the WIP section. The author has also confirmed successful packaged-app installation on clean Linux and Windows 11 computers without the development environment. Optional local translation models are separate downloads, not bundled with the app.

2.2 Machine requirements: with and without local models

Use	What the computer needs
Standard app, without local models	For source installation, Python 3.10 or newer and PySide6 (Qt 6.8 or newer). A compiled distribution supplies its app runtime. No AI model or dedicated GPU is required for editing, cue playback, captions or manually authored subtitles.
Audio and audience output	A working audio output and support for the media formats used in the show. Connect the intended interface, speakers and any audience display, then rehearse on that setup. Reference-video playback also needs a compatible video decoder.
Storage	Space for the app and dependencies, project audio/video, exports and backups. Media and the number of copies determine project size; keep additional free space for installation and generated files.
Online subtitle translation	An internet connection and consent to send the selected text to the provider. No local language-model installation is needed. Provider limits and privacy details are explained in the translation chapter.
Optional local subtitle translation	A writable installation folder, compatible Python 3 and virtual-environment support, and a supported x86-64 or ARM64 system. The current translator runs on CPU; a dedicated GPU is not required. On x86-64, a dependency requires SSE 4.1. Linux also needs working unprivileged `unshare` network isolation.
Local-model storage estimates	Budget approximately **3 GiB of engine downloads**, **6 GiB installed**, and **3 GiB of download cache**, plus models and temporary space. Example language models download around **62–83 MiB each**; each translation direction is a separate model. These are planning estimates, not fixed sizes for every OS or dependency version.

The base app has no published numeric minimum for RAM or CPU speed. Project size, simultaneous audio, video and optional models change resource use. Before installing local models, review the installer's free-space, available memory and compatibility information. Choose only the language pairs you need, leave headroom for temporary files, and keep model installation and translation

separate from live show operation. Model downloads require internet access; installed local models can translate offline.

2.3 Untested operating systems and feedback reminders

At startup, Curtain Call checks the operating system locally. The platform-feedback rule shows a one-time notice on Windows versions other than Windows 11, macOS and other non-Linux systems. Linux and Windows 11 have been tested by the author, not on every hardware configuration. This is a feedback invitation, not a failed installation check. The warning asks you to consider sending feedback and logs at least monthly.

On an untested system, a reminder is enabled by default and appears at a subsequent startup at least **30 days** after the last notice. It does not run a background timer or appear while the app stays open. Turn it off under **Edit > Preferences > Interface > Platform feedback**; the first warning still appears. The warning or reminder is deferred to a later startup if the app opens in Show Player or audio is playing. The dates and preference are stored locally in app preferences, not in the project. In the preferences JSON, `untested_os_reminders` controls the reminders and `untested_os_notice_dates` records the most recent notice date for each system.

To share a log, open a project and use **Project > Export > Session log**. Review the file for private information first: it can include project names, paths, cue details and events. Use **Help > Send feedback or ideas** for a reviewed report and review/remove the automatically prepared current-session snapshot. Additional logs can be attached manually. Nothing is collected by the reminder itself. Nothing is sent automatically, and the reminder records only that it was shown, not whether anything was sent.

Open a terminal in the Curtain Call application folder. On Linux or macOS:

```
tools/setup.sh
```

```
./curtaincall.sh
```

On Windows, install Python with its **Add Python to PATH** option, then run:

```
tools\setup.bat
```

```
curtaincall.bat
```

Run setup once before starting the launcher. If Python cannot be found, install Python 3.10 or newer and reopen the terminal. If Linux reports that venv is unavailable, install your distribution's Python

virtual-environment package (often `python3-venv`) and rerun setup. If dependency installation fails, check the error and internet connection; do not continue as if setup succeeded.

2.4 Install with pip instead

From the application folder, create and activate a virtual environment. On Linux or macOS:

```
python3 -m venv .venv
```

```
. .venv/bin/activate
```

```
python -m pip install .
```

```
curtaincall
```

On Windows:

```
python -m venv .venv
```

```
.venv\Scripts\activate
```

```
python -m pip install .
```

```
curtaincall
```

The `curtaincall` command is available in the activated environment. Installing from project metadata includes the offline guide and images. For optional selectable-text PDF script import, use `python -m pip install ".[pdf-import]"` instead of `python -m pip install .`. This installs pypdf; no OCR is performed on scanned pages.

2.5 Portable mode and first start

For portable use, run `tools/setup.sh --portable` on Linux/macOS or `tools\setup.bat --portable` on Windows. This creates the `portable.txt` marker next to the app. Settings then live in `portable-data/`, with new projects suggested under `portable-data/projects/`. Keep your complete project files on the portable drive too.

A virtual environment is not a cross-platform package: create it on the target computer at the location where you will use it. Rerun setup if you move the app or switch operating systems; do not assume a copied `.venv` will work.

At first start, the Welcome page offers **New project**, **Open project**, **Open demo project** and recent projects. **Open demo project** (also under **File**) opens The Lighthouse Keeper. The app packages do not include it, so the first use asks whether to download it from curtaincall.yzonk.com; you choose where it is saved, the app shows the download progress and opens the project when it finishes. Later uses open the saved copy directly, so the demo downloads once. Open the demo copy before experimenting; opening a project does not fire its cues. Use **File > Save as...** (Ctrl+Shift+S) to create an independent named copy and switch to it. Current unsaved edits are included; the original project file is not changed. For a folder project, choose a parent folder. Existing folders are never overwritten and copies inside the original project are refused. All project media, saved AD takes, processing versions and other project files are copied; referenced external Library audio is brought into the copy's library. Linked project paths are refused rather than following them. Finish processing/imports and stop playback first. On Welcome, right-click a recent item and choose **Duplicate project...** to copy it without opening or changing the original; the copy appears in Recents. Normal click-to-open is unchanged. Press **F1** or choose **Help > User guide** for this offline manual, then continue to **Quick start: a doorbell cue**.

2.6 Launch and associate project files

Use `curtaincall.sh` on Linux/macOS or `curtaincall.bat` on Windows. With dependencies available in your active environment, you can also launch `python3 -m curtaincall "path/to/project.ccshow"` or pass its project folder (use `python` on Windows). Quote paths with spaces, including The Lighthouse Keeper. A second launch hands its requested project to the already running per-user window rather than opening a competing operator window. Unsaved-change handling still matters. If the running window cannot acknowledge the launch, the new copy reports the problem instead of silently exiting. An older running version cannot acknowledge this protocol: save its work, close it and launch again.

After source setup on Linux, register desktop integration:

```
tools/install_linux_integration.sh

# Remove integration later:

tools/install_linux_integration.sh --uninstall
```

Packaged copies do this on their own. A compiled build (Windows installer or portable zip, macOS app, Linux AppImage/deb/tar.gz) adds the app icon and the `.ccshow` handler for the current user the first time it starts, and checks again on later starts so a moved or updated copy stays correct. It writes only user-level data (the registry under `HKCU\Software\Classes` on Windows, `${XDG_DATA_HOME:-~/.local/share}` on Linux) and needs no administrator access. Portable copies (`portable.txt`) never change the computer, and **Register Curtain Call for .ccshow project files for this user** in *Preferences* > *Interface* turns the automatic registration off. The script above stays available for source checkouts and repairs.

This installs the application-menu entry, the `.ccshow` file association and transparent yellow Curtain Call icons for both the application and project files. All five icon sizes are installed for the current user under `${XDG_DATA_HOME:-~/.local/share}`; no administrator access is needed. Projects without a custom image also use this logo in the recent list and project settings. Run the installer again after moving the app folder or updating its icons. Reopen the file manager if it still displays a cached icon; if opening a project still asks for an application, choose **Open with > Curtain Call** and make it the default. Uninstall removes the registered launcher, MIME definition and installed icons, not the application, settings or projects. Desktop integration is optional and writes to the user's desktop-data folder even in portable mode. Other platforms can use the default-program steps in README; the

Linux file-icon registration does not apply to Windows or macOS. A desktop association launches the app; it does not make audio auto-fire on opening a project.

2.7 The project file and separate exports

New projects are single `.ccshow` files containing the project data and media. In this guide, `exports/` in an output path means the separate sibling `<project name> -- exports/` folder, not a folder inside the project.

Item	Contents	Copy for a complete handoff?
`<project name>.ccshow`	Editable show data, script, media, saved runs, thumbnail and permitted project fonts	Yes; save before copying
Private recovery workspace	Working files and latest autosave; not encrypted	Kept on this computer for recovery, not a replacement for a backup
Sibling `<project name> -- exports/`	Generated deliverables, including reports and logs	As needed for distributing outputs

Older folder projects remain supported: copy the entire project folder, including `project.ccshow` and its media, rather than only its JSON file. Do not assume an absent run means the editable show is damaged if no run has been recorded. Every saved project and autosave snapshot includes the current **software_version** and **software_build** alongside **format_version**, the project schema number. These are independent of **Project settings > Version**, your own production revision label.

A project saved by a **newer** Curtain Call (a higher software or format version) is **refused, not opened**: the warning "Project needs a newer Curtain Call" names the relevant versions. This is deliberate. Opening it would drop the settings this version does not know, and the next save would quietly write the project back in the older format. The file is left untouched and stays in the recent list, so it opens normally once Curtain Call is updated. A file with an unreadable software or format version is reported as damaged instead. Older software versions, known older builds, and unversioned files require approval and an automatic backup before migration; see **Approve a project version upgrade** in Projects, saving and recovery. Older projects with the current software version but no recorded build open normally; the build is recorded the next time they are saved.

2.8 User/computer data lives elsewhere

Item	Location / meaning
preferences.json	User configuration folder under curtaincall, or portable-data/ in portable mode
recipes.json	Beside preferences; user cue patterns, not embedded in projects
Venue-profile library	Inside New projects preferences; copies into shows are independent
venues-user.json	Beside preferences; personal physical venues and source checks
datashows-user.json	Beside preferences; personal video-projector specifications and source checks
projectors-user.json	Beside preferences; personal lighting-fixture specifications, not video projectors
projector-installations-user.json	Beside preferences; local venue/projector links and installation details
venues.json, datashows.json, projectors.json	Older local catalogues kept untouched as migration backups, when present
portable.txt	Empty marker next to the app selects portable configuration storage
Recent-project paths / table layouts	Preferences convenience data, not production contents

2.9 Rebuild the demo and documentation as a developer

In a development checkout containing the demo source and reference audio, `python3 tools/create_demo_project.py` recreates demo/The Lighthouse Keeper; do not run it on your only modified demo copy. Release app packages do not include the demo. `python3 tools/prepare_assets.py` rebuilds app imagery. The manual visual generator is `QT_QPA_PLATFORM=offscreen python3 tools/generate_help_images.py`; it uses a repository-local disposable copy, never starts playback, and writes only bundled illustrations. **--diagrams-only** rebuilds SVG/PNG diagrams without screenshots.

The test command is `QT_QPA_PLATFORM=offscreen python3 -m unittest discover -s tests -t .`. It uses standard-library unittest. Unattended runs are protected by `tests/support.py`:

Guard	What it does	Change with
Per-test time limit	A test running longer than 120 s prints the stack of every thread (naming the test and the line it hung on) and ends the run	`CURTAINCALL_TEST_TIMEOUT_SECONDS`
Unanswered dialog watchdog	A modal dialog nobody answers is closed after 5 s and the test fails with the dialog's title, instead of waiting forever	`CURTAINCALL_TEST_DIALOG_SECONDS`
Multimedia release	`release_engine_audio()` stops playback and lets Qt's FFmpeg threads wind down before the next test builds players	—
Test-window cleanup	Windows created during each test are deleted after its cleanup, stopping their media players and releasing hidden dialogs and timers; pre-existing shared fixtures are preserved	—

During development, run the affected test modules first. Before publishing, run the full supported-Python CI matrix and installed-package smoke check; repeat broader audits only when changes or

failures justify them. The main window owns its session, engine and audience-feed timers, so they cannot outlive its widgets.

Tests answer dialogs by patching the helper (`ask_yes_no`, `pick_open_file`…) or with a `QTimer.singleShot` that fills and closes the window. `.github/workflows/tests.yml` runs the suite on the oldest supported Python (3.10) and the newest (3.14), with FFmpeg installed so synthetic-video decoder tests run rather than skip, then builds the wheel, installs it with no package index and runs `tools/smoke_test_installed.py` outside the source tree: bundled guide, images, icons and launcher, and the Help window and its PDF export with network access blocked. `tests/test_python_compatibility.py` catches syntax and standard-library names newer than 3.10 locally. Release metadata comes from `curtaincall/version.py`; the version updater/build tooling keeps packaging in step. These are maintainer commands, not steps required to run a prepared performance.

Note: The demo is a teaching production, not a guarantee of safe levels in your room. Its varied cues deliberately demonstrate automatic chains, infinite loops and ducking. Review and sound-check your own copy.

3 Quick start: a doorbell cue

Example: Open a **copy** of **The Lighthouse Keeper**. Its **House Ambience (AMB)** and **Phones Announcement (ANN)** already demonstrate a background and foreground recording. Add a new Doorbell track and cue only to the copy; the supplied demo does not include a doorbell file.

1. Choose **File > New project** and give the production a name. The dialog suggests **Create in** `Documents/Curtain Call Projects`; keep it, create a subfolder, or **Browse** to any other folder. A folder that does not exist yet is created, and the app checks that it can really be written before the project is created, so a read-only or unavailable location is reported in the dialog instead of failing later.
2. In **Library** (`Ctrl+2`), use **Add audio** to import your doorbell recording. Audio is copied into the project's library folder.
3. In **Script** (`Ctrl+1`), type or import the script. Put the cursor on "The visitor presses the bell" and choose **Add cue here**.
4. In **Cue Creator** (`Ctrl+3`), select the new cue, choose the doorbell track, and set its volume. Keep the cue armed so it can fire.
5. Open **Show Player** (`Ctrl+4`), check preflight in the output panel, select the cue and press **Space** or **GO**.
6. Save with `Ctrl+S`.

3.1 Build a safe first test

Start with one short, familiar sound at a modest volume; keep the amplifier low while checking the output. Name the recording **Doorbell** and give it the short code **BELL**. Name the cue **10 — Visitor rings**, use Type **start**, Loops **1**, Pre-wait **0**, Post-wait **0** and Continue **none**. Link it to an action line such as "The visitor presses the bell".

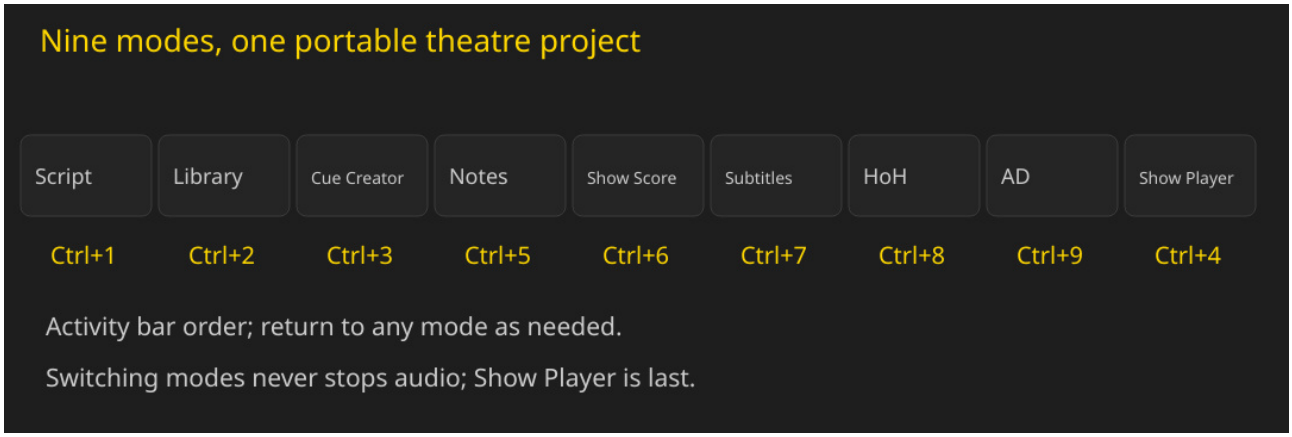
In Cue Creator, **Test fire** plays an armed cue immediately, ignoring its pre-wait and suppressing its automatic followers. It is an audition, not a simulation of the entire sequence. In Show Player, GO uses the programmed waits and continuation. Test both before relying on a timed chain.

3.2 Know which mode owns the change

Work	Mode
Write paragraphs and choose script elements	Script
Import media files, preview files, edit track metadata	Library
Set volumes, fades, loops, timing and cue links	Cue Creator
Fire cues and monitor sounding audio	Show Player
Record corrections under a dated run	Rehearsal Notes
See the whole evening drawn as one timeline	Show Score
Create, translate, time and preview dialogue titles	Subtitles
Prepare multilingual character narration texts and casting notes	AD — Audio Description

One library recording can be used by many cues. Changing a cue's volume does not change the recording or another cue using it. Opening this help does not pause playback; close it or return focus to the performance window before using performance keyboard shortcuts.

3.3 The whole workflow at a glance



The activity bar is the vertical strip on the left. You may revisit any editing mode; the diagram shows the bar order, not a rule that locks you into a workflow. Show Player is last in the bar even though its shortcut is `Ctrl+4`. A production is ready only when its creative script, copied media files, programmed cues and operator instructions agree.

Tip: Begin with a copied demo, not your only production project. The demo has real media files and automatic cue chains, so turning the amplifier down is still necessary before auditioning.

3.4 First ten minutes with The Lighthouse Keeper

About the demo play: The Lighthouse Keeper is a three-act play about Tomas, an aging lighthouse keeper, and Elsa, a young woman who arrives with urgent news: a ferry carrying forty people is lost in the fog outside his lighthouse. The play explores time, memory and responsibility as Tomas must decide whether to light the lamp that night. The story moves from the mundane morning harbour through an escalating crisis, to a distant dawn when the fog lifts. The dialogue is poetic and deliberate. Its soundtrack uses music by Zolvi selected to demonstrate library metadata, cueing, fades, loops and ducking; it is not original music written for the fictional scenes.

Quick-start walkthrough:

1. Download the separate demo, copy its `.ccshow` file and open the copy.
2. Read the Pre-show section in Script, including the two-paragraph opening announcement and the operator directions before the play begins.
3. In Library select **AMB House Ambience**. Inspect the filename and metadata; play its preview only if the output is connected and levels are safe. This recording is the harbour ambience heard at the start of Act 1.
4. In Cue Creator inspect cues **1** and **2**. Notice how cue 2 (Phones Announcement) ducks the AMB cue without replacing or editing the ambience recording itself—a key technique for layering dialogue over background sound.
5. In Show Player read the next-cue card, observe **Nothing playing**, then fire cue 1 deliberately. A loading delay is possible even with no programmed wait.
6. Listen to the opening ambience at the rehearsal volume and operator levels. Stop all before ending the test. Save project edits and keep a copy of the `.ccshow` file if you changed anything important.

You do not need to use every feature immediately. A script link and cue-in note make even a basic start cue much easier to call correctly than an anonymous row.

3.5 Explore the demonstration examples

The demo show contains all three acts with 20 cues that demonstrate core features:

- **Act 1:** Opening ambience and announcement, then the harbour scene (1.1), an argument at the gate (1.2), and the lantern moment (1.3) with a special fade curve.
- **Act 2:** A memory of summer, followed by the central storm (2.2) with heavy ducking to make dialogue intelligible, and the climb up the stairs (2.3).
- **Act 3:** Dawn breaking (3.1) and the finale (3.2) that plays automatically.

The demo also includes fictional technical and dress rehearsals with linked notes in different states. Open **Rehearsal Notes** and compare the earlier snapshot with the current show to see an example resolution and cue-volume change. In Library, the Phones Announcement track has an example marker; its cue links that marker to the opening dialogue line. **Project > Venue profiles…** contains two fictional profiles and setup checklists; neither is selected by default. Their trim values are examples, not recommendations for a real venue.

The three **Demo M**, **Demo D** and **Demo G** cues at the end illustrate memo, delay and group cues. They are disarmed, so GO skips them. If you arm and fire the group example, it starts the Pre-show start cues, including sound.

Subtitles and translations: The show includes 46 English subtitles generated from the spoken dialogue, plus 7 professional Portuguese translations for key moments: the opening welcome, the crisis announcement, and the resolution. The subtitle track demonstrates how dialogue-heavy plays can be made accessible to deaf and hard-of-hearing audiences. The translations are illustrative, not a complete verified translation. The project does not include a recorded performance or install an offline translation model; create a run yourself or install a model only when needed. Check audio rights and safe output levels independently before any public performance.

The Show Score and what-if tools can inspect the show silently to see pacing, loudest moments, and gaps between cues.

4 Using the Curtain Call User Manual

Example: Search for **Phones Announcement**, then press Next to move between the worked demo examples. Choose **Cue field reference** to compare the caption, ducking and timing fields before returning to **Show Player and the live script**.

The sidebar shows the active section; narrow windows replace it with a compact section picker. The reader displays only that section. Search still examines **all sections**, case-insensitively, including headings and tables. Previous, Next and Enter reveal the matched section and wrap at either end.

Choosing a section manually keeps the search and its total results, but clears the active match. Next/Enter starts again at the first global result; Previous starts at the last. Clearing search leaves the current section visible with no highlighted match. A query with no results also leaves that section visible and disables result navigation.

4.1 Maximise, print and export the guide

Use the buttons at the top right of the guide:

- **Maximise** (or F11 inside the guide, or the title-bar square, or a double-click on the title bar) fills the screen; press it again to restore.
- **Print** (`Ctrl`+P) and **Export** (`Ctrl`+E) both ask what to include: **Whole guide**, **This section** (the one on screen) or **Selected text** (drag over text in the reader first; the entry is greyed out with nothing selected).
- Print opens a paper preview with page navigation, **Print…** (choose a printer) and **Save PDF…**. Printouts use dark text on white paper with numbered pages, whatever the screen theme. Tip, Warning and Note callouts keep distinct, print-friendly background colors with dark text, and each chapter starts on a new page. The first content of a section or selection export stays on page one.
- A **Whole guide** PDF has a branded cover with the Curtain Call banner, guide title, release/build and author/site details. Its back cover repeats the version and contact details and reminds operators to rehearse with the actual show equipment and keep a backup playback method. Neither cover has the running header, footer or page count. The banner keeps its original proportions. Section and selection PDFs do not add covers.
- Export saves as **PDF**, **Markdown** (`.md`), **web archive** (`.mhtml`) or **plain text** (`.txt`). With a project open you only pick the format: the file is written into the sibling `<project name> -- exports/` folder (Save PDF… in the preview too). Without a project there is no exports folder, so a save window opens in your home folder; pick the format in its file type list. The whole-guide PDF's suggested filename includes the application version and build, for example `CurtainCall - user guide -- v0.1.1 build 20261109.pdf`. PDF exports run in a separate process with an activity indicator and Cancel button; cancelling or a rendering failure leaves any previous PDF untouched.
- The web archive replaces the old HTML export. It bundles the rendered text, styles and every included illustration into **one standalone file**; `.mht` is also accepted in the save window. Move or email just that file: no **images/** folder or internet connection is needed. Open it in a browser or viewer that supports MHTML (for example, Microsoft Edge); not every browser supports web archives. Use PDF instead if the recipient cannot open MHTML. A missing or unreadable illustration stops the export with an error rather than producing an incomplete archive.

- Afterwards the **Export complete** question offers Open file, Open folder or Send by email..., as for every other export.

Example: before a tech rehearsal, open **Show Player and the live script**, choose **Print > This section**, check the preview and press **Print...** to hand a paper copy to the substitute operator. To mail the whole manual, use **Export > Whole guide** and keep the suggested `.pdf` name. For an illustrated offline web copy, choose **Export > Whole guide > Web archive** and send the resulting `.mhtml` file on its own.

4.2 Numbered sections and readable printed pages

The reader, chapter picker and exports use the same heading numbers: **9** identifies a chapter, **9.3** a section inside it and **9.3.1** a subsection. The source manual keeps ordinary Markdown headings; Curtain Call adds the numbers automatically, so inserting a section does not leave stale numbers elsewhere. Exporting **This section** keeps its full-guide chapter number rather than restarting at 1. Selected headings keep their numbers too.

The PDF's **Contents** includes all three levels, indented by depth, with the page number where each heading actually appears. Cover pages remain free of running headers and footers; the contents uses physical PDF page numbers so they agree with your PDF viewer. In a PDF viewer that supports links, click an entry's title or page number to jump to that heading. Printed copies retain the page numbers for finding the same place by hand.

Chapters always start on fresh pages. A substantial section starts on a new page when it would otherwise begin partway down a page. Short FAQ answers, glossary entries and small subsections can share a page; headings near the bottom move forward when there is not enough room for their opening text. This adds useful breaks without giving every short entry a mostly empty page. The same layout is used by **Print**, **Save PDF...** and direct PDF export.

For example, export the Cue Creator chapter to share **9.5 Pre-wait, post-wait and continuation** and its **9.5.1 Worked timing examples** with a substitute operator. Those identifiers match the whole-guide export and the on-screen reader.

4.3 Hover help on controls

Holding the pointer over a control shows one sentence saying what it does, and the keyboard shortcut in brackets when it has one. Icon-only tool buttons, text boxes with a clear button, cue fields, preference options and the settings dialogs all explain themselves this way.

Example: Hover **Cue out (note)** in the Cue Creator's Notes tab and the tooltip reads "What follows this cue, so the operator knows what is expected next". Hover the search box of the output panel and the small × explains **Clear the log search**. Hover the **Pause** button in the Show Player for "Freeze the show where it is... (Ctrl+Space)".

A tooltip is a reminder, never the manual: it never hides a rule you need to run the show. When two controls look alike (pre-wait and post-wait, duck attack and release, fade out time and fade-out duration), read the tooltip and the chapter that owns the setting before relying on it in performance.

Tooltips appear on the dark theme and never take focus, so they cannot fire a cue.

4.4 Dialog actions and keyboard navigation

All file and folder pickers use the **native operating-system dialogs** by default. Under **Edit > Preferences > Interface > Window and controls**, untick **Use native OS file and folder dialogs** for Curtain Call's themed pickers. Saving preferences changes subsequent pickers immediately; no restart is required. This applies to opening, saving, importing, exporting and choosing folders. Native availability and appearance depend on the operating system and Qt platform plugin. Native pickers can remain light when the OS uses a light theme; Curtain Call does not change desktop settings to force them dark. Choose themed pickers when an always-dark appearance is more important than native OS integration.

The yellow button identifies a dialog's main action; **Cancel**, **Close** and secondary actions stay neutral. Destructive actions keep their danger styling and confirmation or backup requirements. Standard confirmation buttons are text-only; functional controls such as alignment and playback retain their icons. Read-only lists and dialogs offering several alternative import choices do not imply that closing or any particular choice is the recommended action.

Use **Tab** and **Shift+Tab** to move between controls. Focused buttons have a visible border. **Enter** from a single-line field invokes the main action where available; a deliberately safe Cancel default is preserved for destructive cleanup. Multiline text fields retain their normal Enter behaviour. Invalid project, marker and reference-library fields stay in the draft; validation reveals the relevant tab or scrolling field and focuses it for correction. Form rows can wrap and longer panels scroll in compact windows.

4.5 The boxes you type into

Text, numbers and notes start at the same distance from the left edge of every box: the Library's Details section, the cue editor, the settings dialogs and the Preferences window all put the first character in the same place, so a column of fields reads as one column and your eye lands in the same spot whether you are typing a name, a volume or a fade time.

- **Numbers** (volume, text size, how long a caption stays up) start where the text of a text box starts, so a spin box beside a text box no longer looks shifted to the left. For bounded settings, drag the slider for a quick adjustment or type in the number box for an exact value. Precise time positions and identifiers remain text or number fields.
- **Lists** (font, alignment, position, presets) start in the same place, with the arrow at the right clear of the words.
- **Notes** (multi-line boxes) start in the same place too, with a little extra air under the top edge.

Settings pages scroll instead of squeezing their boxes when the window is made short, so a field keeps its height and stays readable. Profile editors show a vertical scrollbar only when their content does not fit; it disappears again when enough space is available. The Library's Details section behaves the same way: make the window smaller and the boxes keep their size, with a scrollbar beside them instead of shorter fields.

Dialog windows can be resized by dragging the three-dot grip at the lower right. The custom window frame also supports dragging its edges and corners. Controls may limit how small a dialog can become. Each dialog remembers its last size and the place you moved it to on this computer, even if you cancel the dialog or restart the app; a remembered position is pulled back on screen if the monitor layout changed. A saved size is reduced if needed to fit the available screen, subject to the dialog's minimum size. Small messages such as a confirmation or a one-line notice have no resize grip and keep the size their text needs. Dialogs without editing tools but with a lot of text, such as **About Curtain Call**, scroll vertically instead of growing past the screen; horizontal scrolling is not used, because text wraps. Floating audio tools and the marker editor follow the same rule; each audio tool remembers its own dimensions. Compact sizes are only their first-opening defaults. Choose **Edit > Preferences > Interface > Reset saved dialog sizes**, then save preferences, to forget all recorded dialog and floating-panel sizes and positions. Cancel leaves saved sizes unchanged. Reopened windows use defaults; this does not reset section dividers, table columns, projects or other preferences. All dialogs and floating panels have a thin neutral outline to distinguish them from the

interface beneath them. Clicking a slider track selects the value at that position immediately, rather than moving by a page step. Dragging, keyboard adjustment and exact numeric entry remain available.

Drag a divider between resizable sections to share the available space. These section sizes are remembered in this computer's preferences across projects and app restarts. In Subtitles, the preview opens larger by default and keeps your chosen height when a reference video loads.

4.6 Tip of the day

When appropriate, the app offers **one** short tip at start-up, chosen from those this computer has not been shown yet: where a feature lives, or a habit that saves a rehearsal. Tips for newer applicable releases come first, preserving their order within each release. The dialog does not show tip counts; **Next tip** browses the list and wraps around.

- **Stop the automatic tips:** untick **Show a tip when the app starts** in that window, or in **Edit > Preferences > Interface > Tips**.
- **Turn them back on:** tick the same box in Preferences. **Show every tip again** forgets which tips this computer has read, so the series starts from the beginning.
- **Read one now: Help > Tip of the day** always opens the dialog, even with the automatic tips off or every tip already read.

Every tip displayed, including each one visited with **Next tip**, is recorded by its stable ID in this computer's `preferences.json`. History is saved as tips are shown, not just for the last tip when the dialog closes. It is never written into a project or into the static bundled `assets/tips.json`. If preferences cannot be saved, the app reports the error and remembers the IDs for this session. The bundled texts have an `id`, a `title` and a `text`, and may declare the versions they apply to.

Example: `applies_from_version: "0.2.0"` on a tip means the app holds the text already, but only offers it from version 0.2 on, so a later feature can ship with its tip. `applies_until_version` retires a tip that no longer applies.

A tip is never opened automatically when the app starts straight into the **Show Player** or while cues are sounding, so nothing covers the performance view on the way into a show; use **Help > Tip of the day** afterwards. Tips never play audio, never change a cue and never take a key away from the Show Player.

4.7 The guided tour: first steps, pointed at

The first time the app is used on a computer — and again after an update that adds stops — a short guided tour opens over the workspace. The window goes dark, one area stays lit with a gold outline, and a card beside it explains what that area is for. The five introductory stops are:

1. **Nine modes on the left** — the activity bar, and `Ctrl+1`..`Ctrl+9`.
2. **Write or import the script** — Script mode and the import formats.
3. **Import the audio once** — Library mode, waveforms and metadata.
4. **Every setting lives on the cue** — the cue list in the Cue Creator.
5. **The output panel tells you what happened** — `Ctrl+J` and the session log.

Release-specific lessons follow the introductory stops when they apply to the installed version. For example, a fresh installation of version 0.1.1 includes its applicable update lessons as well. If the output panel is hidden, the tour temporarily shows it for its lesson and restores your choice afterwards without changing the saved preference.

The tour changes mode as it goes, so the panel a stop describes is really on screen. **Next** and **Back** (or the right/left arrow keys) move through the stops; the card always says which stop you are on. The card fits the available window; if a lesson is too long to fit, scroll its text to read the rest while the navigation buttons stay visible.

Choice	What it means next time
Next / **Finish**	The tour is read; the stops you saw are not shown again.
Skip for now (or `Esc`)	The tour closes; stops already shown stay seen, and only the remaining applicable stops are offered at the next start.
Close forever	Every stop is marked as seen **and** the tour is switched off for this computer.

- **Reopen it by hand: Help > Guided tour** always starts the tour from the first stop, whatever the preferences say. Use it to show a colleague where things are.
- **Turn it off: Edit > Preferences > Interface > Guided tour**, the box *Show the guided tour on first use and after an update*. **Show the tour again** next to it forgets which stops this computer has seen, so the tour is offered again from the beginning.
- **Moving from one version to the next:** a stop may declare the versions it belongs to, so a release adds stops for what it brings. The automatic tour shows **only unseen applicable stops**,

including any older lessons not yet shown; it never repeats seen stops. Release lessons are marked "New in 0.3.0". Help > Guided tour replays all applicable stops, including ones already seen.

The stops themselves live in the app's `assets/tour.json`: each has an `id`, a `title`, a `text`, the `area` it points at (`activity_bar`, `script_editor`, `library`, `cue_list`, `output_panel`, `menu_bar`) and optionally `applies_from_version` / `applies_until_version`, inclusive like the tip rules. The bundled tour is static; it is not rewritten as you read it. Which stops have been seen is stored separately in `preferences.json`, on this computer only. A stop whose area a later release renames still shows its text, centred in the window, instead of failing.

The tour never plays audio, never changes a cue, never saves the project and never takes a key away from the Show Player: it is not offered when the app opens straight into performance or while cues are sounding, and it closes by itself when the window closes.

4.8 Read the visual language

Tip: Gold-tinted Tip boxes suggest a faster or clearer workflow; they are advice rather than required steps.

Warning: Red-tinted Warning boxes describe a safety risk, destructive action or limitation. Read them before executing the nearby procedure.

Note: Blue-tinted Note boxes clarify what is stored, what is assumed or what does not happen automatically.

Example: Green-tinted Example boxes walk through concrete values, usually from The Lighthouse Keeper. Copy the workflow, not an unchecked output level.

These labels are written as Markdown quotes beginning with **Tip:**, **Warning:**, **Note:** or **Example:**. The reader adds a dark background and inset margins. Search still sees the label and full text. Printed guides keep readable named labels even when the screen tint is not reproduced.

Illustrations shrink to the available reader width. For small screenshots, maximise the guide to read the detailed controls, or consult the numbered legend and nearby text instead. The chapter picker replaces the sidebar on narrow windows. The manual is not a web service: its text and images work without an internet connection.

4.9 What search does and does not do

Search matches the rendered guide, including table cells and callout text. It ignores Markdown punctuation such as the bold asterisks. Searching here does not search your current production or alter a cue. Use the Script find box, cue list or output-log filter for project-specific evidence. Help stays open as a separate, non-modal window; audio continues until you deliberately stop it in the app.

When exporting only selected text, drag across complete paragraphs or table cells first. A selection is a document fragment, not the original complete chapter. Markdown exports retain image references; when sharing an editable manual copy, distribute its **images/** folder too. MHTML web archives include the images even when exporting a selection; PDF also needs no accompanying illustration files.

5 Projects, saving and recovery

Example: Copy **The Lighthouse Keeper.ccshow** to **The Lighthouse Keeper — before dress.ccshow**. Open the copy and check that **House Ambience** resolves. Set version to **Dress run 2 — 2 November**, save, and retain the original file as your rollback.

A single-file `.ccshow` project contains the project and its media in a ZIP container. Open and share the file directly. Curtain Call works on a private per-user copy; **Save** rebuilds the file atomically. Autosave writes recovery data in that workspace and does not update the shared file until Save. The workspace is retained for recovery, including after closing; it is not encrypted. If the file changes elsewhere, use **Save as** to preserve the recovered copy. Exports go into a separate `<project file name> -- exports/` folder next to it. Existing generated exports are not included in the `.ccshow` file. Working saves use an uncompressed ZIP for faster rewriting. A clean sharing `.ccshow` uses maximum ZIP compression, which may take longer to create; neither compression nor the private workspace encrypts the project. New projects use the single-file format. Existing folder projects remain supported; **Save as** keeps the format of the open project. For a folder project, copy the entire folder, including `project.ccshow` and media. Older exports inside that folder are not moved or deleted.

Audio is copied into the project by default:

How the audio arrived	What Curtain Call does
Library > Import or drag and drop	Copies the file into `library/` and stores `library/<name>`
Relink / Relink missing	Copies the replacement into `library/`
Project JSON edited by hand or made by another tool, pointing outside the project (`/home/sound/cues/Storm.wav` or `../cues/Storm.wav`)	On opening, copies the file into `library/` and rewrites the path; the output panel lists the copied tracks
An absolute path that already points inside this project	Only rewrites it as a relative path; nothing is copied
A file that cannot be found	Left alone; it shows as missing and is fixed with Relink

Import and relink copy original files, never move or delete them. Explicit project cleanup can delete unneeded copies inside the project, not outside source files. If a file with the same name and identical

content is already in `library/`, it is reused, so opening the project twice never creates `Storm (2).wav` duplicates. A different file with the same name gets a numbered name. The copy counts as an edit: save the project to keep the new paths, or use Undo to go back to the outside paths.

Example: a colleague sends a project with a cue pointing at `/media/usb/Thunder.wav`. Open it with the USB stick plugged in; the log shows "Copied 1 audio file(s) from outside the project into library/: Thunder". Save, unplug the stick, and the show still plays.

To keep outside paths, turn off **Preferences > Saving & safety > Copy audio from outside the project folder into library/ when opening**. Preflight then lists every such track as "Audio outside the project folder", and **Project > Copy outside audio into project** gathers them on demand.

Project settings hold the venue, operator, notes, master volume and defaults. App preferences hold computer/user choices such as autosave, recent projects, appearance and table layouts, plus defaults copied into each newly created project. Existing projects do not inherit later preference changes.

5.1 One production, venues, rehearsals, and iterations

A **project** represents one production of a show, not one venue or one rehearsal. The production may travel to more than one place. A project stores one set of physical **Venue details** and can also contain several named venue profiles for room-specific setup notes and trim; only one profile can be active at a time. Venue details identify a place, while a venue profile holds an operator-tested room setup. They are separate and are not linked automatically.

Rehearsals belong directly to the project. Each dated rehearsal has its own notes and show-state snapshot. Its optional **Iteration** value is a free-text run/revision label, such as **Run 2** or **Dress**; it is not a separate object, folder, or parent that contains rehearsals. Several rehearsal sessions can use the same label. The change digest compares rehearsal snapshots, not venues or complete project-file revisions.

Rehearsals currently have no saved link to a physical venue or venue profile. If a production rehearses in different rooms, include the room in the rehearsal name or notes (for example **Dress — Run 2 — Civic Theatre**); changing the project's active venue profile later does not record where an earlier rehearsal took place.

Example: Keep one **The Lighthouse Keeper** project for the production. Add profiles for the rehearsal room and Civic Theatre, then activate the appropriate one for the current setup. Create a dated rehearsal for each session and use its optional iteration label to identify the run or revision. The sessions and snapshots stay with the project; the label does not group them under a venue.

5.2 Compare show revisions

Open **Project > Compare show revisions** to inspect two versions side by side without opening, changing or saving either one. Each side can use the current show, including unsaved edits, or a separately chosen `.ccshow` file. Use **Choose file...** for each side; **Refresh** reads the selected files again. Changes to the current show also update the comparison.

The table lists differing saved fields from the entire show, including settings, script, tracks, cues, subtitles, characters and rehearsals. Select a row to see complete left and right values below; `(not present)` is different from JSON `null`. Items with unique IDs are matched by ID, and position changes are shown separately. Other lists are compared by position. Inserting an item may therefore change the listed positions of other items too.

The comparison reads files as written, without upgrading them. Older formats, missing fields and software-version differences can produce differences that are not artistic changes. Files are not verified to belong to the same show; choose related copies yourself. This does not compare the contents of audio, script documents or other media files in the project. There is no automatic revision history or restore: keep whole-project backups separately.

Example: Compare the current **The Lighthouse Keeper** show against a saved dress-rehearsal `.ccshow` to find changed cue timing, then select a row to read both values. The comparison never fires audio.

5.3 Unavailable recent projects

Welcome and **File > Open recent** keep missing or unreadable project files listed, but **grey them out** and label them **Unavailable**. They cannot be opened from those entries while unavailable. Hover for the saved path and a reminder to reconnect the drive or find a moved file. A folder where the project file should be also counts as unavailable; readable files are not checked for valid project contents until you open them.

The Welcome list rechecks while visible (about every three seconds) and whenever you return to it; the Recent menu rechecks whenever it opens. Reconnect the drive and the entry becomes usable again. For a renamed or moved project, use **Open project** to select its new location; the new path is remembered normally. Unavailable entries are **not automatically removed from history**, and reopening the last project at startup skips an unavailable path without showing a file-error dialog.

To remove all remembered paths, choose **File > Open recent > Clear history...** and confirm. This also clears the Welcome list; it does not delete project files.

Example: Unplug the touring USB drive: its recent show is grey and marked Unavailable. Plug it back in at the same location and wait a few seconds on Welcome, or reopen the Recent menu, to use that entry again.

5.4 Create and prepare a production

1. From Welcome choose **New project**, or use **File > New project**.
2. Choose the project location and a production name. The app creates a new `.ccshow` file; an existing file with that name is not overwritten.
3. Open **Project settings** to record venue, operator and version. Set a conservative master volume and the default fade and track volume.
4. Import audio and the script, then save. Check that the project contains the expected media before copying it to a show computer.
5. On the show computer open the copied `.ccshow`, enter Show Player, inspect preflight, and audition through the actual sound system.

Project settings travel with the show. App preferences do not: projects folder, recent-project history, autosave interval, appearance and table layouts are user/computer choices. Defaults help construct new cues; inspect existing cues rather than assuming a default change rewrites all their values.

5.5 Identify projects at a glance with a thumbnail image

When you return to the Welcome page, **RECENT PROJECTS** are listed with a small thumbnail image to the left of each project name. The default image is the Curtain Call app icon; you can replace it with a photograph, artwork or logo that represents the show:

- Open **Project settings** and find **Project image** in the Details tab.
- Click **Choose image...** to select a PNG, JPG or other image file from your computer.
- The image is center-cropped to a square, converted to PNG and stored inside the project at `thumbnail/project.png`. The recent-project list uses a 64 × 64 icon; larger square copies are made for other project-image views.
- Click **Use default** to remove the custom image and return to the app icon.

Save the project to include its thumbnail before copying it to another computer.

5.6 Save, back up and move a project

`Ctrl+S` saves the open show. Undo/redo restores project edits, not the sound card or files already copied on disk. Saving does not make Undo into a disk-backup tool. Keep dated copies of the saved `.ccshow` file, especially before importing a replacement script or relinking audio.

For a folder project, copy the entire folder, including `project.ccshow` and `library/`. For a single-file project, save first and copy the `.ccshow` file. `script/` keeps the imported document; the editable script lines are saved in the project JSON. Exports are separate deliverables, not a complete editable backup.

Autosave writes a recovery snapshot; for a folder project it also saves the primary project. For a single-file project, autosave updates only the private workspace. Explicit **Save** updates the `.ccshow` file. It runs when there are unsaved edits and the configured interval elapses. It is not a continuously updated history and does not guarantee that the last keystroke survived a crash. A write error is reported; investigate it rather than assuming the project is safe.

5.7 Carry fonts with the project

Fonts are not part of a project by default, so a computer without your fonts shows a substitute. **Project > Project fonts…** lists every font chosen for captions, subtitles, HoH feelings and per-title or language overrides, and whether it is available here. Fonts whose licence permits it are also copied into the project automatically whenever a project is opened or saved, so a project keeps carrying its permitted fonts without extra steps; fonts that need confirmation or cannot be copied are reported in the log instead. When a project uses fonts missing on this computer, the app offers — after one approval — to download verified free copies of all of them from the free font catalogues, and reports any that could not be found.

- **Copy fonts into project** stores usable copies in `fonts/` (with a `fonts.json` list and checksums). Only fonts whose licence permits redistribution (for example SIL Open Font Licence, Apache, Ubuntu) are copied.
- A font with an unknown licence is copied only after you choose **Confirm permission…**. A font that forbids embedding or redistribution is never copied; install it on each computer instead.
- When a project opens, its `fonts/` copies are loaded for Curtain Call only. Missing fonts are reported in the log. **Locate font file…** finds a file yourself, **Install for me…** installs a copy for your user account (never automatically, no administrator rights) and **Use fallback…** substitutes a font for this session without changing the authored choice.
- **Download free font…** searches the free catalogues — Google Fonts, then Fontsource (which also hosts open fonts not on Google Fonts) — for a missing font after you approve the network contact; only the family name is sent, and files come from fonts.gstatic.com or the Fontsource content network. Each download is size-bounded and verified to really contain the requested family, saved for this app (also for future projects), and copied into the project only when its own licence metadata allows it. A family that is not published there is reported, never substituted.
- Profile exports that need fonts are written as a ZIP; others stay JSON. Importing a ZIP validates its contents and installs the fonts for the app.

The check cannot prove that a licence text is accurate. Glyph coverage is reported, and a bold style missing from a font is imitated by the system.

5.8 Approve a project version upgrade

Opening an **older or unversioned project** shows **Update this show?** before any migration. The question explains why the show needs updating and lists three steps: keep an exact backup, update and save the project file, then open it without starting playback. It gives the saved and current app versions (and file-format numbers when those change). Legacy data that still needs conversion is checked even if someone has already stamped the current software version into the file. Choose **Not now** to cancel: the file, current open project and recent-project entry stay unchanged. Opening through Recent, startup, the command line, a file association or an imported/copied `.ccshow` uses the same check.

Choose **Update and open** to create a **byte-for-byte backup before migration** of the project JSON: `project.ccshow.pre-upgrade.bak`. For a single-file project, this backup is in the private workspace, and **Save** updates the archive. Keep a separate copy of the original `.ccshow` before updating. For a folder project, the backup is beside the original JSON. Existing backups are never overwritten; later copies use `project.ccshow.pre-upgrade.2.bak`, then `.3.bak`, and so on. The backup contains the entire original file, including fields this app does not understand. After converting supported older data, Curtain Call saves the upgraded file atomically with the current **software_version**, **software_build** and **format_version**. After saving and reopening the upgraded project, it does not ask again unless another conversion is needed. The output panel lists the backup's full path after a successful upgrade.

If backup creation or saving fails, an error is shown and the project is not opened. The original stays intact; if saving failed, keep the completed backup too. If another program changes the file while you decide, the upgrade stops instead of replacing that change. Do not edit the same project in two programs at once. Opening an older `project.autosave.ccshow` upgrades and backs up **only the recovery file**; it does not overwrite a damaged primary project. Save normally after checking the recovered show to restore `project.ccshow`.

Example: Before upgrading an older single-file project, make a separate copy of its `.ccshow` file. Approve the upgrade, check the cues and save. For rollback, close the app and open the untouched copy with the older app. For a folder project, retain a copy of the complete folder instead.

A migration backup is **not a complete project backup**: it does not copy audio, source scripts or runs. Keep dated copies of the `.ccshow` file, or the whole folder for a folder project. Do not remove version metadata to force a newer file into an older app; update the app instead. Developer `Show.load` calls

without a confirmation callback automatically approve supported upgrades but still enforce version checks, backups and atomic saving.

5.9 Clean up a project

Use **Project > Clean Up Project...** to reduce a finished show to its current project data and cue-used audio. **Make an independent project backup first**: copy the saved `.ccshow` file, or the whole folder for a folder project. This is permanent file deletion, not an undoable Library edit.

The preview offers category checkboxes: **Unused media**, **Waveform caches**, **Exports, reports and logs**, **Recorded rehearsal runs**, **Archived script revisions**, **Original imported scripts**, **Autosave / recovery**, **Orphan AD recordings**, **Interrupted audio renders**, and **Unused subtitle reference videos**. Each shows the number and total size of its files for the current selection; Unused media also counts Library entries. The exact list below changes with the selection. Changing a checkbox clears the backup acknowledgement, so review and tick it again. Categories with no selected changes cannot be cleaned.

The selected category choices are remembered per computer in `cleanup_categories` after you confirm cleanup; **Cancel** does not change the remembered choices. The first time, all categories are selected. Untick **Unused media** to keep unused Library entries, their audio and their note links; retained tracks are protected even if their audio lives in a selected exports or script folder. The preview lists every file to delete, its size and category, plus the Library entries to remove. A track is used if **any cue** references it, including disarmed cues, placeholders and non-start cues. Unused Library entries are removed; shared audio stays if a retained track still points at it. Retained tracks also protect their saved audio versions. Removing an unused track can make its versions eligible for deletion; review the exact preview. All saved AD takes are protected, not just the preferred take. Their original WAVs, active processed audio and recoverable versions stay. **Orphan AD recordings** lists files in `audio-descriptions/` that have no saved take or metadata reference, including files left after undoing generation. AD waveform caches appear under **Waveform caches**. **Interrupted audio renders** lists leftover `.curtaincall-render-*` staging files inside generated media folders; normal successful or cancelled processing removes these automatically. Downloads/models, upgrade backups, project thumbnails and unknown root files are not treated as disposable trash. **Unused subtitle reference videos** removes old imported copies in `reference-video/` that are no longer assigned to subtitle recording. The current reference video stays, even if stored under another cleanup folder. Linked or outside files are never deleted.

Removed	Retained
Unused project audio and orphan files in `library/`	Audio referenced by retained tracks, wherever it lives
Unreferenced AD audio and interrupted render staging files	All saved AD takes and recoverable AD versions
Waveform caches, all files in `exports/` (reports and logs included)	Current `project.ccshow`, settings, cues, subtitles and authored script lines
All recorded files in `runs/`	Rehearsal groups, observations and historical snapshots stored in the project JSON
Originals in `script/` and its archived revisions	Current script-line IDs and cue/script links
`project.autosave.ccshow`	Unknown root files, other `.ccshow` files and empty folder scaffolding

The table describes what can be removed when those categories are selected, not what cleanup always removes. The original-script filename is cleared when its file is listed for removal; the editable script stays in the project. A note linked to a removed Library track loses **only that track link**, not its text, cue or script link. Deleting recorded runs also removes their recorded timing evidence; estimated Show Score timing remains available.

Review the list, tick **I have a backup and understand that cleanup cannot be undone**, then choose **Save project and delete listed files**. This saves current unsaved edits before removing files and clears undo/redo history so Undo cannot restore tracks whose files are gone. **Cancel** changes no project data and deletes nothing. If the project or listed files change during preview, cleanup refuses to proceed: open a fresh preview.

The preview shows **Before**, **Selected files** and an **Estimated after** size. After cleanup, the result reports measured **before**, **after** and **net change**; this may differ from the selected-file total because saving the show can change the project JSON and files can change during the operation. Sizes sum logical bytes of regular files in the working project, including kept files; links and their targets are excluded. They are not disk allocation or compressed archive sizes. Displayed sizes use decimal MB/GB (1 MB = 1,000,000 bytes), with KB/bytes for small files so nonzero sizes do not misleadingly round to zero. Exact byte counts are still used internally for validation and deletion plans. If a size cannot be measured, an error is reported rather than a guessed total. For a partial deletion, the measured result reflects what remains and lists failures.

Cleanup is blocked while audio (including paused audio), waits, Library or AD preview, run recording or audio-tag import is active. Stop those and let imports finish first. Links and files outside the project are never followed or deleted; the preview lists untouched linked paths. Used audio is protected even if a hand-edited project places it in `exports/` or `script/`.

Save failures leave the project and files untouched. If some files cannot be deleted, a warning and output-log entry list each failure; the cleaned project is already saved, and another cleanup can retry the remaining orphan files. Waveform caches, exports and recovery copies can be generated again by normal use; cleanup does not switch autosave off or erase the current in-memory log.

Example: In a backed-up copy of The Lighthouse Keeper, remove the cue for an abandoned **Spare storm** track. Clean Up Project lists its unused Library entry and copied audio along with old run recordings and reports. After cleanup, cues for **House Ambience** still find their audio; the authored script and rehearsal observations remain in the project.

5.10 Recover after an interrupted session

For a single-file project, back up the `.ccshow` file and reopen it. A recovery question appears only if the retained private workspace has changes not confirmed in the saved project file. Choose **Recover**, inspect the contents, and **Save**. If an interrupted workspace save left a newer autosave snapshot, recovery opens that snapshot. If the external file changed, use **Save as** to preserve the recovered copy separately. The workspace is not encrypted.

For a folder project:

1. Before changing anything, make an independent copy of the project folder.
2. Open `project.autosave.ccshow` **in its original project folder** using **File > Open project**. Relative media paths then still resolve.
3. Inspect cue order, recent script edits and rehearsal notes. Audition only after checking the output level.
4. Use **Save** to write the recovered contents to `project.ccshow`.
5. Re-open the primary project and verify it. Keep the independent copy until you are satisfied with the recovery.

If the primary and recovery files both fail to open, use an independent backup. Do not edit the only surviving JSON file experimentally. A missing recent-project entry does not delete a project: browse to its folder manually.

5.11 Understand three different kinds of state

State	Stored where	What to expect
Production content	`.ccshow` archive, or `project.ccshow` inside a folder project	Cues, tracks, script, project settings, rehearsals and snapshots travel together
Operator/computer choices	preferences.json	Opening mode, autosave, appearance, column layouts and app defaults are local
Runtime evidence	runs/ and exported session logs	Flight media files are separate files; an unexported session log is not an editable show backup

Project names, operator credits and the production version help colleagues tell copies apart. They do not change the app's release number. A saved project can be moved because its media paths are relative to the project folder (outside audio is copied in when the project opens, see above). If a new computer has different output routing or codecs, that portability does not replace a sound check.

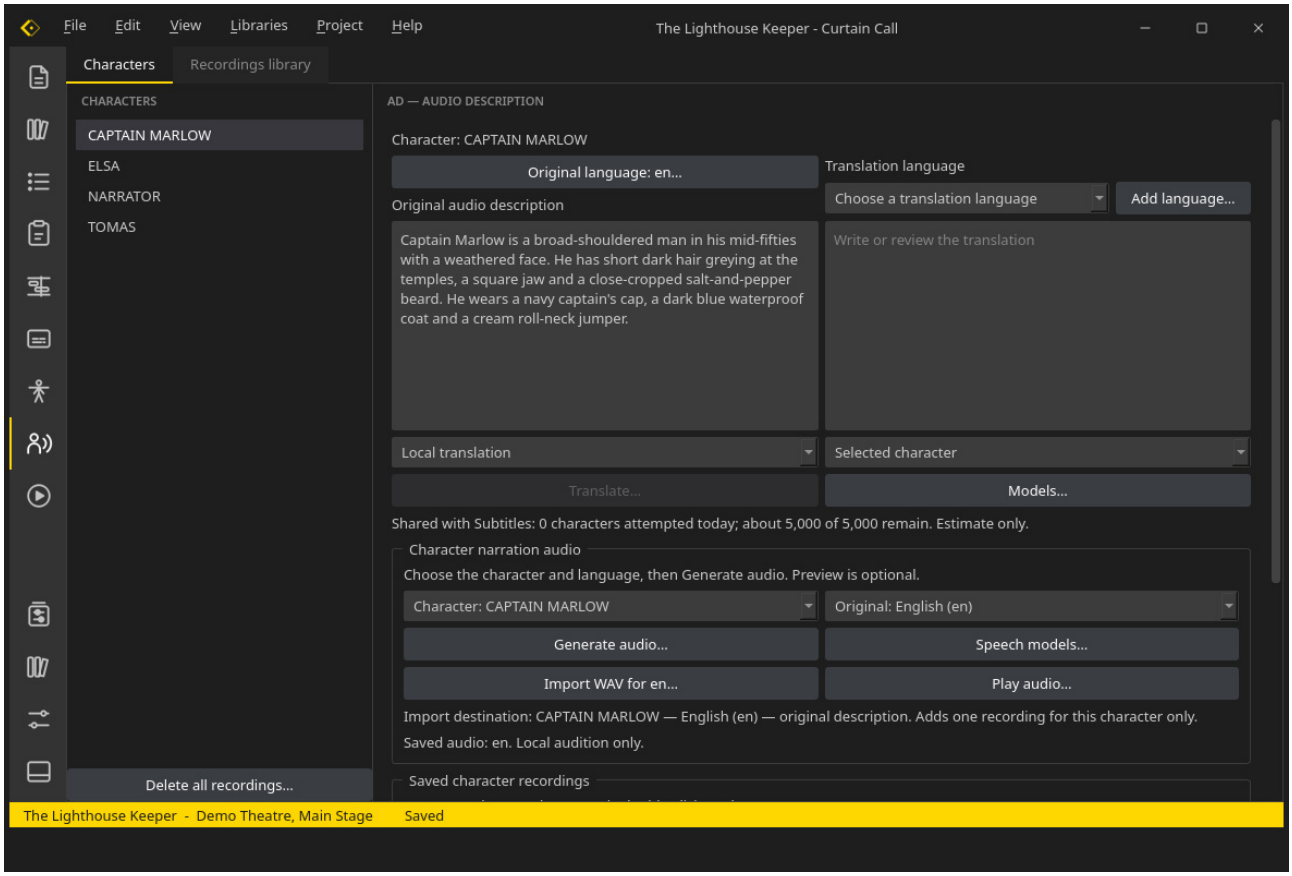
Warning: Undo restores project data, not external files. Removing a cue can be undone; distributing an outdated PDF or deleting an independent backup cannot be repaired by project Undo.

Before replacing a show or closing with unsaved edits, read the confirmation rather than assuming it is an autosave guarantee. A preference can save edits on entering Show Player. A separate preference can ask before quitting during playback. Neither adds confirmation to the emergency stop keys.

If a recent-project item is stale, use Open project to find the actual file. The recent list is a convenience and can be limited to 1–30 entries. It is not a catalogue of every project you own, and removing an entry does not erase its folder.

6 Audio description: character texts and narration

Open **AD — Audio Description** from the activity bar, **View** menu or **Ctrl+9**. AD provides character narration texts, translations, local speech generation and reviewed WAV imports. Saved narration can be auditioned locally; live AD cue timing and audience routing are not implemented. Switching to AD does not stop normal show audio.



The Lighthouse demo includes fictional character notes for all four characters and English **Original audio description** examples for Tomas, Elsa and Captain Marlow. These examples describe physical appearance and clothing, not stage actions. Captain Marlow's appearance is a fictional character-

design reference for this translation exercise; he remains voice-only in the demo, so it is not a claim about visible staging. Their AD translations start empty: open a copy of the demo, select one of these characters in AD, choose a target language and use the translation controls to test the workflow. The Narrator intentionally has no AD text, so you can also try entering your own. Character notes remain in Script and are not sent as part of AD translation.

6.1 Keep a saved character register

The **Characters** section in AD is a character-selection list only, sharing the same saved register as Script. Add/remove, casting notes, appearance navigation, dialogue and exports remain in Script's **Characters** sidebar tab. It is a saved project cast register, not an extra block inserted into the play's text. Script Character paragraphs automatically populate it during editing and imports. Repeated names are matched without regard to capitalization or extra spaces. Names in the register also feed **character auto-completion** in Script, including names you have added before writing their speeches.

In Script, select a character to edit **Character notes** (casting reminders, motivation or pronunciation), see how many **Character paragraphs** use the name, or choose **First appearance** to open Script at their first paragraph silently. Type a **New character name** and click **Add** to prepare another cast member. Notes are separate from the **Original audio description**: old casting notes are preserved and are not automatically copied into narration.

Editing an anchored Character paragraph follows the character's name while retaining notes, AD text and translations when the old name is no longer used elsewhere. If the old name remains in other paragraphs, the new name is a separate character. Names no longer used in the script stay in the register so their authored content is not lost. **Remove unused** deletes the selected unused character, notes, AD text and translations, including narration audio assignments, after confirmation; names still used in the script cannot be removed. Audio files are retained for Undo.

6.2 Write and translate character narration

Choose the **Original language** for all character descriptions, independently of the subtitle original language. Unset originals use the project/app default from **New projects**, initially the system language; explicit choices override it. Enter the selected character's **Original audio description**, describing what listeners need to know about them. Text is saved as you edit and can be undone; it does not change dialogue or subtitles. Changing the original language relabels the texts, not translates them, and requires confirmation. A language already used by AD translations cannot become the original.

Use **Add language…** to choose a target by language name or code. Each character has a separate editable translation in every target language. Merely choosing a new target creates no saved translation until text is entered or applied. Select a character and type the translation manually, or use:

Action	Effect
Local translation / Online translation	Choose the local engine or MyMemory; online requests still require explicit consent.
Selected character / All characters + Translate…	Translate this character or all nonempty originals to the displayed translation language.
Models…	Open the same local model installer as Subtitles; installed directional pairs are shared.
Translation language	Switch the editable translated text without changing originals.

Local translation uses the existing optional engine and models. Missing installations open the shared installer. The official catalogue loads directly; model downloads still require approval. Catalogue browsing sends no project text or audio. After installation, text processing uses the same offline safeguards as Subtitles. A general en/pt model is not proof of Portugal-specific quality: using it for **pt-PT** requires explicit approval and never silently substitutes **pt-BR**. Review language, pronunciation and meaning before preparing narration.

Before an online request, the consent preview names **MyMemory**, shows the exact outgoing text requests and explains retention/third-party risks and usage limits. Only original AD descriptions are sent, not character names, casting notes, subtitles or other project content. Only send text you have permission to share. Usage is shared with Subtitles and is an estimate, not a provider quota

guarantee. Long paragraphs split at word boundaries within **500 UTF-8 bytes** per request; paragraphs are reassembled afterward. Splitting can lose context, so review the result. A single word exceeding that limit must be edited or translated locally/manually.

Local and online results open **Review AD translations** before **Apply**. Apply explicitly replaces the shown existing translations in **one undoable change**. Cancel, service/engine errors, incomplete results or project/text/language changes during translation leave existing translations intact. You can edit applied text manually afterward; clearing a translated editor removes that character's translation for the selected language only.

Names, notes and texts travel with `project.ccshow` under `characters`, using stable character IDs, `audio_description` and `audio_description_translations`; the shared original language is `audio_description_source_language`. They participate in save, autosave and Undo/Redo. Older shows receive empty AD fields without changing notes. The register and AD texts are not inserted into script exports or past rehearsal snapshots and do not assign subtitle colours. Both panes scroll in short windows. Review descriptions with blind and visually impaired advisers before future audio use; machine translation is not a guarantee of useful or accurate audio description.

6.3 Generate, import and review narration audio

Create or open a project, select a character, and use **Character narration audio**:

AD starts with a **Characters** tab, leaving room for future tabs. Original and translated descriptions sit **side by side**, with translation tools first and voice generation below.

Optional character self-introductions: give different characters different voices and write their descriptions in the **first person**, as though each is presenting themselves: "I am the keeper. I wear a long coat and carry a lamp." Use the same voice for that character's presentation consistently, and review the wording with blind and visually impaired advisers. This is an artistic option, not a requirement or an automatic text rewrite. For distinct voices, generate each character separately; the all-character batch uses one chosen voice per language.

Choose the named **Character** or **All characters**, then the named **Original** or **Translation** language (with its language code), **Selected languages…** or **All languages**. For several clips, select a local voice for each checked language. **Generate preview** optionally synthesizes only the **first sentence** of the clip shown in the selector; use **Play preview** to pause or replay it; successful previews start playing automatically. You can skip preview completely and choose **Generate audio** to generate the full text and save the complete batch directly. Saving all assignments is one undoable edit; cancellation or failure changes no project audio. A language needs an installed matching voice; missing voices are not silently skipped or substituted.

Selected text only uses the highlighted passage from either text editor and the selected character. It exports a separate WAV after preview; it never replaces the character's complete narration. **Voice speed** pairs a horizontal slider with an exact numeric field, from **0.50 x to 2.00 x**; **1.00 x** is normal. Higher values speak faster. Speed changes clear any earlier preview; full generation uses the selected speed and records it in the saved audio's provenance. For batch generation, choose the voice in the **Voice speed** selector and use the same slider or exact numeric field to adjust it. Each voice remembers its own speed for this batch; switching voices restores that value. The character preview list follows the selected voice's language: choosing a Portuguese voice shows only its Portuguese descriptions. This preview filter does not exclude other checked languages from full batch generation. Preview and full generation use the speed belonging to the clip's voice. Clips sharing the same voice share its speed. Voice selectors show **installed voices only**. **More speech models…**, beside the selector, opens additional downloads **only for the selected language**, initially showing available-to-install packages. In multi-language generation, each language has its own More speech models

button. Closing the manager refreshes the installed voices while keeping the previous choice where possible. If no voice is installed, install one through this button; checking a missing language also offers installation. Declining or cancelling generates no audio. There is no substitute language or silent download. Licence/rights warnings appear only in the download confirmation, not again during generation.

1. Choose the named **Original** or **Translation** language in the narration selector. This combines the text source and language choice without a separate source selector. A translation must have a selected language; generating speech also requires nonempty text.
2. Choose **Generate audio…** to review the exact text and select a matching voice. No description text, character name or casting note is uploaded.
3. **More speech models…** opens the language-scoped optional voice manager; **Edit > AD speech models…** opens the full catalogue. Opening either performs local file checks only, not downloads. It distinguishes available-to-download, installed and unavailable voices. Search, status filters and checkboxes follow the same layout as the translation model manager. Engine compatibility, voice integrity and a fixed synthesis sample are checked automatically during installation; generation checks the engine and voice again. There is no manual Test ready button or temporary Ready to use status. Installed does not certify narration quality. **Uninstall selected…** removes only the checked voices' known files, preserving the engine, other voices and saved project audio. Unrelated files block removal.
4. Optionally choose **Generate preview**, then **Play preview** to hear the first sentence. This short preview is never assigned as the full narration. **Generate audio** generates the entire description directly, without requiring a preview or another save step, and stores it inside the project's `audio-descriptions/` folder as one undoable edit. Generation runs in a cancellable worker; playback starts only when requested. Replacing a previous assignment requires confirmation. Cancel, failure or a changed project/character/text/language/audio assignment during review cannot replace existing narration.
5. **Play audio…** auditions the saved file. A warning identifies descriptions edited since the audio was assigned; editing text does not delete older audio.

Available local voices

The manager includes **all 182 voices** in the pinned official Piper catalogue, across its languages and quality variants. Search and check as many voices as you want, then choose **Download and install…**. There is **no app-imposed voice count limit**; disk space, dependencies and download availability still

apply. Checked entries hidden by filters remain included. Each package is downloaded, verified and installed separately; a later failure reports how many completed and preserves those installations. Voices are not downloaded by opening the list.

The previously reviewed English voices below retain their documented credits. Additional catalogue voices display a warning because their full training-chain rights have not been cleared by Curtain Call; voices referring to Lessac have an explicit restriction warning. These are available after acknowledgement, not silently blocked or presented as unrestricted. Open a voice's model details for its pinned model-card link; its card is also downloaded with the package. The warning appears **once, in the download confirmation**; it is not repeated during checks or generation. Terms remain in model details and saved provenance. Multi-speaker packages currently use their default speaker, not a selectable speaker identity.

- **Piper Joe and Mike:** local neural English US voices, with documented CC0 datasets and a CC0 rdh training base. Download and compare them using the same reviewed preview workflow; no subscription or online speech request is required.
- **Piper Cori Samuel UK:** local neural English UK voice, with British phonemes and the documented CC BY 4.0 Hi-Fi TTS training chain. Keep the dataset credits.
- **Piper Cori Samuel / hifi_tts_92:** optional local neural English speech, with British vowels and en-US phonemes. Review pronunciation; it is not labelled a native UK voice. Requires Piper 1.8.0 and a compatible system Python/venv. **Download and install…** asks permission before fetching the isolated engine from PyPI and the pinned voice/configuration/model card from Hugging Face and its CDN. Files use reviewed sizes and SHA-256 hashes; the voice is staged and checked before publication. Existing voice folders are not overwritten.
- **Piper Portugal / tugao:** downloadable neural pt-PT voice for local use, with a **licence warning**. Its direct dataset is CC0, but its Lessac-derived training source restricts resulting synthesis. The manager displays the warning, and asks you to acknowledge it only when downloading the package. You can continue after acknowledging it; this is not a grant of commercial, public-performance or redistribution rights. Review the model card and source terms and use it only where permitted. Saved narration retains the provenance warning. Portugal and Brazil are distinct: pt-PT is never silently changed to pt-BR.

Synthetic eSpeak voices are no longer selectable. Existing synthetic audio, voice identities and system installations are preserved. Better local neural Portugal speech with a less restrictive training chain remains under investigation. Voice quality must be assessed by listening, not by assuming that a

neural engine or higher-quality label guarantees suitable narration. The WAV is uncompressed 16-bit mono PCM at the chosen model's native sample rate; the currently installed medium and high models use 22,050 Hz. Generation changes Piper's length scale for speed and otherwise keeps its model inference defaults. No lossy compression is introduced when the WAV is stored or tagged. Naturalness mainly depends on the trained voice, pronunciation and phrasing. Higher-tier models exist for some languages, but the pinned Portugal catalogue currently has only the tugao medium voice. Increasing file sample rate, converting to stereo or saving as 24-bit cannot recreate detail that the model did not generate. Audition different permitted models and punctuation/phrasing first. A permitted human recording imported as WAV remains an option for more expressive delivery. Normalization/compression help playback consistency, not pronunciation or acting.

Piper is installed alongside the optional translation engine in `local-models/piper/`, separately from the main app environment. Voice sizes vary by package; the original medium voices are approximately **63 MB** across three files. Installation confirmation and transfer progress show readable MB/GB totals, file count, speed and estimated remaining time. Engine installation and voice verification use a moving indeterminate bar when size/time are unknown. Engine/dependency download size, installed/cache/temporary space and RAM requirements are not fully measured; free disk space does not guarantee success. Inference uses CPU, not CUDA. Model update checks are still planned, not automatic.

The English voice's documented training dataset is **CC BY 4.0**, with attribution to **Cori Samuel**, the **Hi-Fi Multi-Speaker English TTS Dataset** and **NVIDIA Corporation**. The manager and saved audio record retain the attribution and link to the licence. Keep those credits with exported/shared narration and review terms for your intended use; the app does not grant rights in third-party voices. See **Third-Party Notices** for source and model-card links.

Import WAV… is independent of model availability. Select a permitted mono or stereo 16-bit PCM WAV, listen to its preview, verify its language and rights, and explicitly import it. The source file is not changed.

Audio is copied into unique files under the project's `audio-descriptions/` folder. Generation and import immediately save the character assignments and metadata to the project too; no separate Save is needed. This also saves other pending project edits. If saving fails, the audio remains on disk and the changes remain in memory with an explicit warning: retry Save before closing. New filenames include the character name, language code, AD label and a short unique ID, for example `Keeper_pt-pt_AD_a12b34c56d78.wav`. Earlier recordings keep their existing names and remain available for

Undo. WAV files include embedded RIFF INFO metadata: title, YZONK branding, software, UTC creation time, character name/ID, language, voice ID, speed, text hash and voice attribution. Full project text is not embedded. Metadata display depends on the player; the audio samples are unchanged. The outlined **Saved character recordings** group below generation lists all takes for the selected character with language, filename and a changed-text indicator. Select a recording to load the inline player; double-click to play it without changing the language for new narration. The player uses one line: **Delete recording · Play · seek slider · elapsed/total time**. **Delete recording…** permanently removes the selected WAV after confirmation and removes current character assignments to that file. Other recordings are untouched. Undo cannot recreate a deleted WAV; older history or saved copies of the project may still refer to the missing file. The recording list marks absent WAVs **Missing file** and inaccessible or unsafe paths **Unavailable**, so these entries are distinguishable before playback. If the WAV is already missing, **Delete recording…** instead offers to remove its stale recording entry and current assignments without deleting any file. Cancelling keeps the entry; permission errors are reported and leave it intact. Generate or import several takes of the same character and language to compare voices and speeds. Each new take is kept in the project; previous takes remain in the list, not just in Undo history. The newest take is initially **In use**. Choose an older take and click **Use this recording** to make it preferred for that language; this selection is undoable. Alternative takes and the preferred choice are saved with the project. Deleting an alternative leaves the preferred take unchanged; deleting the preferred take requires choosing another. At the bottom of the left Characters column, **Delete all recordings…** lets you choose languages or **All languages**. After a second confirmation it permanently deletes all takes and retained versions for those languages across every character, and removes missing-file entries. Other languages remain. Linked/outside files and files shared with another language or Library track are preserved with an explicit warning. Deletion errors are listed so remaining entries can be retried. Undo cannot restore physically deleted files. The **Import WAV for [language]…** button shows its destination immediately below: selected character, language name/code and original or translation. The file picker and listening confirmation repeat the same character and language. Import always adds one take for that character and language, even when generation is set to all characters or several languages.

The **Recordings library** tab gathers only the preferred **In use** recordings across every character and language. It uses the main Library's read-only table layout: recordings and a scrollable Details panel side by side above a full-width Waveform panel, with adjustable horizontal and vertical splitters. The table has Character, Language, Voice, File, Length and Status columns. All preferred recordings are

visible; alternative takes stay in the Characters tab. The first recording is selected automatically when no earlier selection exists. Select a row to load the inline player and seek slider, or double-click to play. The waveform uses the same waveform viewer and zoom area as Library mode: click to seek, Space to play/pause, Ctrl+wheel to zoom, and Zoom out, Zoom in or Fit to adjust the view. Details show the character, language, file, voice, recording's metadata. Character, language, filename, voice, text fingerprint and generation provenance are automatically filled into Notes for new recordings, not repeated in a separate information block below the editor. The Details panel uses the same editable metadata form as Library mode: code, name, artist, composer, publisher, rights source/status, album, year, genre, copyright and notes. Edits are saved per take in the project and support Undo; they do not rewrite the file's embedded tags or alter its character/language assignment or generation provenance. The waveform toolbar and **Audio markers** panel share the same construction as Library mode, including icons, order, spacing and button layout. The **Audio markers** tab and waveform allow adding, editing, moving, removing and previewing markers with the same marker dialog as Library mode. The waveform also offers **Trim Silence**, **Normalize**, **Compress**, **Convert** and **Restore**, using the main Library's floating processing dialogs. Applying processing changes that take's active audio and retains its old versions for Restore and Undo. New files remain under `audio-descriptions/`. Conversion can change the active file format; the original generated WAV remains a retained version. Review marker timing after processing. Permanently deleting a take removes its active file and project entry; retained version files remain on disk. AD recordings are not added to the main Library or cue tracks. Missing files and changed description text are explicitly marked. **Show character's takes** opens that character and selects the corresponding recording so you can compare alternatives, change the preferred take or delete it. Generation, import, preferred-take selection, deletion and Undo refresh the library. Switching AD tabs stops audition playback. This library does not add AD recordings to cue tracks or enable live AD output. Character records store per-language `audio_description_audio` assignments, voice identity, description-text fingerprint and attribution in `project.ccshow`. These assignments participate in save, autosave and Undo/Redo; old projects default to empty assignments. Previous files remain for history rather than being deleted when replaced, undone or a character is removed. Save and transfer the single `.ccshow` file, or the whole folder for a folder project, to carry the audio. A missing file is reported explicitly. This workflow is for local authoring/audition: it does not create performance cues, mix speech over dialogue or route AD to an audience device. Rehearse and review descriptions with blind and visually impaired advisers before performance use.

6.4 Navigate and export one character's lines

In Script, choose the **Characters** tab beside **Outline**. The **Character dialogue** list shows that character's nonempty spoken lines in script order. Double-click a line to select its exact Script paragraph without firing audio; **First appearance** still selects the first Character paragraph.

Choose **Export lines...**, then **PDF**, **ODT**, **Markdown** or **Plain text**, and choose a destination. The exported actor script contains only the selected character's speeches and their parentheticals, plus relevant act/scene headings with original scene numbers. It excludes other characters' dialogue, action text, cue annotations, casting notes, AD texts and translations. It does not modify the full script, live playhead or project. Characters without dialogue cannot be exported. Normal full-script exports remain unchanged.

Example: In a copy of **The Lighthouse Keeper**, add **CAPTAIN**, keep “Name is pronounced MAR-low” in Character notes, and write “The captain wears a weathered navy coat and carries a brass compass” in Original audio description. Set the original to English, add **pt-PT**, and write or request a translation. Review and Apply any machine result, then refine it manually. These saved texts can form the basis of later character audio files; no audio is made now.

7 Script writing and cue links

Example: Find **Pre-show** in The Lighthouse Keeper. The action says the house ambience starts when doors open, then the phones announcement plays over it. Select that line, check the linked cue markers **1, 2, 3**, and compare its element with the following **NARRATOR** character paragraph.

Script elements include heading, scene, action, character, parenthetical, dialogue and transition. Enter and Tab help switch between elements. Existing character names appear as completion suggestions.

Import ODT, DOCX, TXT or Markdown from the script toolbar. Selectable-text PDF import requires the optional `pdf-import` package extra; scanned PDFs need OCR elsewhere first. Encrypted and malformed PDFs are not supported.

The gutter shows scene numbers and linked cue numbers. Search finds text in the script. Double-click an outline heading to expand or collapse its entries.

7.1 Inspect and follow gutter links

Hover over a **scene or cue marker** in the script's left margin to see the scene title and linked cues' **numbers, names and actions**. These details follow the marker when you scroll or edit the script and refresh after undo/redo. Click a linked cue marker to **open that cue in Cue Creator without firing audio**. If several cues share the same line, a menu lets you choose the cue; cancelling the menu changes nothing. Clicking a scene marker with no cue selects that scene in the script editor. Hovering and following links never move the live playhead.

The left sidebar has **Outline** and **Characters** tabs. The saved Characters register is shared with **AD — Audio Description**; Script populates names from Character paragraphs and uses them for completion. Select a character to add names, keep notes, navigate their dialogue or **Export lines…**. AD's **First appearance** returns here without starting playback.

To link an existing cue, choose a script line in Cue Creator's script panel and use **Link**, or drag that line onto the cue row. Links use stable IDs so ordinary text edits preserve them. Script exports include ODT, Markdown and PDF; **Print layout** previews pages with cue markers and page numbers.

7.2 Take the gutter markers with the script

Every script export includes **scene numbers and linked cue markers**, in cue-list order. Cue labels include the number and name when both exist, for example **[Cue 12: Storm]**. Unlinked cues are not inserted into the script. ODT exports use a bundled formatting-only template derived from the supplied stage-script example: JetBrains Mono 11 pt body, centered bold character names, indented dialogue, italic parentheticals, scene headings and the example's page size/margins. Action uses the template's body styling; transitions are right-aligned. The template does not copy the example's play text, personal metadata or header/footer text. Fonts are referenced by name, not embedded; a word processor may substitute a font if it is unavailable. PDF/Print layout retain their own existing layout. PDF and **Print layout** show the labels beside the text. ODT keeps them as **margin comments** attached to the paragraph; show comments in your word processor to see them. Markdown shows them as **quoted annotation lines** beneath the element. Curtain Call ignores these generated annotations when re-importing ODT or Markdown, so they do not become spoken text or disturb unchanged cue links.

7.3 Write a scene with meaningful elements

Use the element selector for the current paragraph. The element controls layout and structure; it is not just a font choice.

Element	Use	Enter at the end creates
Heading	Play title or major section	Action
Scene	Scene title/location	Action
Action	Stage business and sound triggers	Action
Character	Speaker name	Dialogue
Parenthetical	Delivery instruction inside a speech	Dialogue
Dialogue	Spoken text	Break inside the same Dialogue element
Transition	End/change of scene	Scene

Tab cycles **Action → Character → Parenthetical → Dialogue → Scene → Transition → Action**; Shift+Tab reverses that cycle. Heading is chosen from the selector, not included in that cycle.

Example: add Scene "Scene 2 — Outside the lighthouse", Action "A bell rings", Character "KEEPER", Parenthetical "(calling upstairs)", Dialogue "Someone is at the door", then Transition "Blackout". After Character, Enter creates Dialogue; choose Parenthetical explicitly if you need an intervening direction. After Dialogue, **Enter continues the same speech**; **Alt+Enter** creates the next Character. In other elements Alt+Enter follows the same rules as Enter. Alt+Enter is reserved for script editing and cannot be assigned as a cue hotkey. Reuse completion suggestions for consistent character names.

7.4 Continue a multi-paragraph speech

Press **Enter** inside Dialogue to insert an internal break without creating another script element. Press it twice for a blank line between paragraphs. The whole speech keeps **one stable script ID and cue anchor**, including after save, reopen and undo/redo. Shift+Enter inserts an internal line break too. ODT exports use native line breaks, Markdown uses `<br />`, and PDF preserves the breaks; ODT and Markdown re-import keep the speech together.

Example: Type “The storm is coming.”, press Enter twice, then type “Keep the lamp burning.” Both paragraphs remain one Dialogue element. Press Alt+Enter and type the next speaker's name.

If a later paragraph needs its **own cue anchor**, use Alt+Enter to create a separate element and choose Dialogue with the element selector. Subtitles still split a long speech into readable chunks

sharing its script anchor.

7.5 Import, check and revise

1. Save or back up before importing a revised document.
2. Click **Import** and choose the ODT, DOCX, TXT, Markdown or supported PDF.
3. Inspect several scenes, speeches and directions. Styled ODT/DOCX paragraphs give better structure than visually formatted but unstyled text.
4. Correct element choices in Script mode; plain text cannot encode all the distinctions of a styled stage script reliably.
5. If any linked line moved, changed or disappeared, review the **Script revisions** dialog (below) before the import is applied.

ODT imports read paragraph styles, including inherited LibreOffice styles. Character, Dialogue, Parenthetical, Scene and Text body styles provide useful structure. PDF import reads selectable text, not page-layout semantics; proofread its order and element choices. Installing the optional `pdf-import` extra does not make scanned image pages searchable.

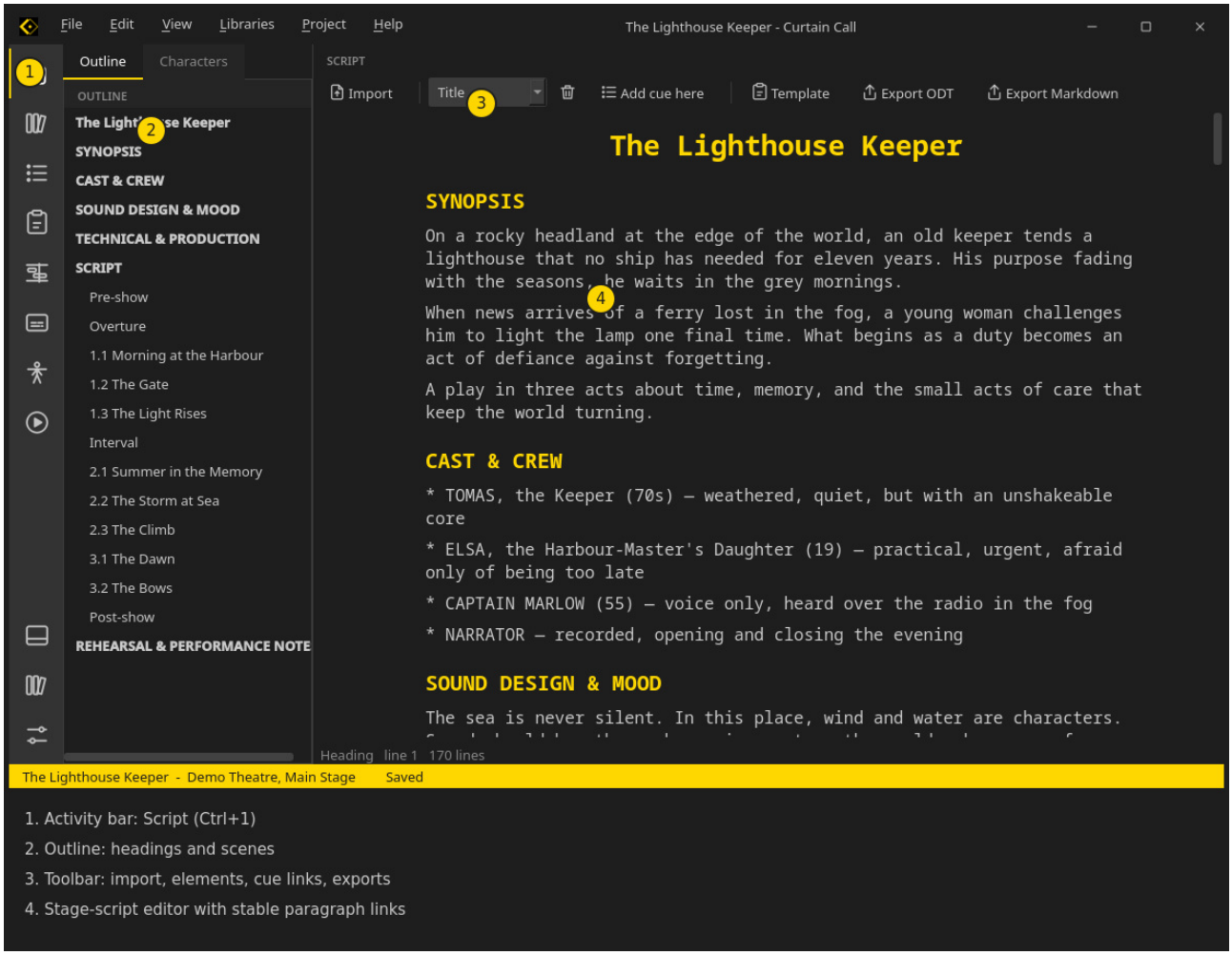
Re-import reuses old line IDs when the paragraph text is **identical**, matching repeated text in order. Editing the existing paragraph in Script mode preserves its ID too. Deleting a paragraph and creating a similar one is not the same as editing it.

7.6 Place a cue precisely

For a new cue, select the triggering action and choose **Add cue here**. For an existing cue, select its row in Cue Creator, select the appropriate script line and use **Link**. Dragging a line onto a cue row links the drop target, not every selected cue. Check the script snippet in that cue's row.

Example: link the doorbell to "A bell rings", not to the next spoken line. Later changing it to "The doorbell rings twice" in the editor keeps the cue link. Re-importing a document with that new wording may require relinking. Use the scene outline and script search to find the precise action; a link is an operator navigation aid, not an automatic text-triggered GO.

7.7 Identify the Script screen



1. Activity bar: Script (Ctrl+1)
2. Outline: headings and scenes
3. Toolbar: import, elements, cue links, exports
4. Stage-script editor with stable paragraph links

1. **Activity bar** selects Script (`Ctrl`+1).
2. **Outline** navigates headings and scenes; double-click a heading to expand or collapse descendants without deleting paragraphs.
3. **Toolbar** imports text, changes the current element, deletes a paragraph, creates a linked cue,

opens a template, exports and finds text.

4. **Editor** presents the stage-script layout, with scene numbers and linked cue markers in the margin. Paragraph identities survive normal text editing.

7.8 Start with a template, not blank formatting

Template creates a starter theatre script. If your project already contains script content or linked lines, the app asks before replacement. Treat that as a replacement operation, not a way to append another scene. It is undoable, but a backup before replacing an imported production remains sensible.

Character completion reuses names already in the script without case-sensitive matching. If **TOMAS** appears earlier, begin typing that name in another Character paragraph and accept the suggestion. This keeps spelling consistent while leaving the paragraph identity intact. Completion is not a cast database and does not create a separate character object.

7.9 Separate text location from cue execution

An **Add cue here** cue is linked to the current paragraph and displayed in Cue Creator. Set its recording and playback values there before relying on it. Cue links do not listen for actors speaking, detect a line automatically or make a script paragraph into a timed cue. They provide reliable visual navigation and operator context.

Several cues can share a paragraph. The demo Pre-show heading groups cues 1, 2 and 3; it is still your cue-in instruction that distinguishes doors open from five minutes before curtain. Place more precise links on action paragraphs when those distinctions improve the substitute operator's view.

Note: The editable script lives inside project.ccshow. The copied ODT/DOCX/text file in script/ is a source document; editing it externally does not update the open script until you import it again.

8 Script revisions: approve changed anchors

Example: In a copy of The Lighthouse Keeper, change the **2.2 The Storm at Sea** heading in the external script to **2.2 The Storm at the lighthouse**. Import the revision, inspect the suggested destination for cue **11**, and approve it only after reading neighbouring lines. Undo should restore the previous heading and cue link together.

When you import a revised script, every line that a cue or rehearsal note points at (an *anchor*) is looked up in the new text. If all of them are **unchanged** the import happens silently and the output log reports the result. Otherwise the **Review script revision** dialog lists each anchor side by side: the cues and notes that use it, the old line with the lines around it, a status and a choice of new line.

- **Unchanged**: same text in the same order.
- **Moved**: same text, but its scene or block was reordered.
- **Edited**: the text changed; the most similar line is suggested.
- **Ambiguous**: several lines fit almost equally (for example a repeated "Yes."); the neighbouring lines and scene heading rank them, but you decide.
- **Deleted**: nothing similar remains; the suggestion is **Unlink**.

Each choice shows a confidence, e.g. `(87%) Is the lamp lit tonight?`. Tick **Only items needing attention** to hide unchanged rows. **Accept all suggestions above** keeps every suggestion that reaches the chosen percentage and sets the rest to Unlink, so you can decide those yourself.

Example: BEN's "Yes." in Scene 1 carries cue 4 and Scene 2 has another "Yes." with cue 9. The playwright swaps the scenes. The scene that changed place shows **Moved** (the other stays **Unchanged**), and each "Yes." follows its own scene because its neighbours and heading still match; neither cue jumps to the first "Yes." in the file.

Apply revision replaces the script and updates every cue and note link as one change: a single **Undo** restores the old script and all links. **Cancel** imports nothing. Before replacing the script, the previous lines are saved in `script/revisions/<date-time>-lines.json`, and an older document with the same file name but different content is kept there as well.

8.1 Review every ambiguous destination

The revision assistant is not a proofreader. It ranks possible new locations for paragraphs used by cues and notes. Confidence measures the match evidence, not the artistic correctness of the destination. A very short repeated speech such as “Yes.” needs its scene and neighbouring dialogue checked even if a suggestion looks plausible.

Review status	Operator question
Unchanged	Does the retained location still describe the intended trigger?
Moved	Did the paragraph move with the correct scene or dramatic block?
Edited	Does the revised wording still represent the same sound moment?
Ambiguous	Which candidate has the correct surrounding dialogue and heading?
Deleted	Should this cue/note be unlinked or reassigned to another explicit trigger?

1. Compare the old line and its context with the candidate line.
2. Review all cues and notes shown as using the anchor, not just the first cue.
3. Change the destination when the suggestion is wrong; choose Unlink deliberately when the trigger has disappeared.
4. Use the attention filter for review efficiency, then check the resulting links in Script and Cue Creator after applying.
5. Save only after confirming the production remains callable in its new order.

Warning: “Accept all suggestions above” can leave low-confidence anchors unlinked. Review them yourself; a successful import does not prove that all cue-in instructions now make dramatic sense.

The revision archive preserves previous line data and conflicting source-document versions. It does not supply an automatic list of playable old show versions. Use project Undo immediately for the whole approved revision, or restore a dated whole-project backup when you need a complete prior production state.

9 Library: media files, not playback settings

Example: Select **House Ambience**, code **AMB**. Its file is **Zolvi - A Gentle Advance (freetouse.com).mp3**, album **The Lighthouse Keeper (demo score)**, artist **Zolvi**, and year **2026**. Its rights source is Free To Use. Cue 1 and Exit Music can reuse this file without sharing their playback settings.

The Library holds audio files, short codes, metadata and notes. Select a track to inspect its full waveform and preview it. Playback settings belong to cues, not tracks: multiple cues can call the same recording differently.

Missing files are flagged. Batch relinking searches a chosen folder for unique filename matches; ambiguous matches need your attention.

9.1 Import and audition media files

1. Open Library and choose **Add audio**; several files can be imported together.
2. Select each track. Review its name, code, artist/composer, licence and notes.
3. Use the preview player to check the recording and full-file waveform.
4. Choose **Add to cue list** to create a cue using the selected track, then set playback properties in Cue Creator.

Imported files are copied into `library/`, including files selected from that folder. Existing files are never overwritten. The waveform can take time to appear while analysis runs; preview the sound and inspect the log if analysis or playback fails.

9.2 Same-name import choices

When an incoming filename matches an existing track's filename, ignoring letter case, **Add audio** offers **Replace**, **Duplicate**, **Rename**, or **Skip**. Matching uses the actual filename, not the displayed track name: `bell (2).wav` is different from `bell.wav`. If several tracks share a filename, select the exact track to replace.

Replace keeps the track ID, cue links, metadata and audio markers, but points it at a new copy of the recording and reanalyses its duration. Review its markers, metadata and rights: these might no longer describe the new recording. Replacement is blocked while show audio or waits are active or paused. The old file remains on disk so **Undo** and **Redo** can restore either version; project cleanup can permanently remove unreferenced files and clears undo history, so back up before cleaning.

Duplicate creates a separate track with a unique filename, such as `bell (2).wav`. **Rename** creates a separate track using the entered filename and the original extension; an unsafe or already-used filename is rejected without overwriting anything. A matching file in `library/` without a track also offers these choices, but cannot be replaced. **Skip** leaves that item alone. **Cancel remaining** stops the batch while keeping earlier successful imports; file errors are reported and the remaining selections continue. Each successful item is a separate undo step.

Example: For The Lighthouse Keeper, replace `bell.wav` to keep its cue links, then inspect the old knock marker against the new recording.

9.3 Process audio and restore versions

Select a track in Library. Beside **Add marker**, a vertical divider separates the audio tools: **Trim Silence**, **Normalize**, **Compress**, **Convert** and **Restore**. Each opens a small floating, non-blocking panel with only its own controls; the main waveform remains usable. Close the current audio panel before opening another. Floating audio and marker editors share a compact default width and fit their initial height to their controls. User-resized dimensions are remembered independently and restored the next time each panel opens. The waveform has no extra preview caption; errors remain visible in the audio panel. These optional tools requires locally installed `ffmpeg` and `ffprobe` on the app's PATH, with the requested filters and encoder available. It does not download tools or send audio elsewhere. If the track points outside the project, first use **Relink** to copy it into the project. The project `library/` must be a real folder, and its filesystem must support hard links for safe, non-overwriting publication. Errors, unsupported encoders and cancellations are reported without changing the track.

Use **Convert** to choose WAV (16-bit PCM), FLAC or MP3 (192 kbit/s); MP3 is lossy, and a WAV conversion to 16-bit can reduce precision. Optional **Compress dynamic range** offers **Threshold** (-60 to 0 dBFS) and **Ratio** (1:1 to 20:1), each with a synchronized slider and exact numeric field. Defaults retain the original approximately -18.06 dBFS threshold and 4:1 ratio; attack remains 20 ms and release 250 ms. A 1:1 ratio leaves dynamics unchanged; higher ratios reduce levels above the threshold more strongly. Optional **Normalize sample peak** offers a **Target peak** from -30 to 0 dBFS, defaulting to -1 dBFS, with a slider and exact field. It measures after compression and adjusts gain; it is not LUFS or true-peak normalization, and lossy encoding can change peaks. Check the result by listening; neither option is a playback limiter or a substitute for mixing and headroom.

Trim Silence opens one floating, non-modal panel in Library and AD. **Remove edge silences (start and end)** is selected initially; **Remove inner silences (between audio)** is off. Select either or both to choose where silence will be removed. The **Silence threshold** defines the level below which every audio channel must be quiet. **Minimum silence** (0.01–10 seconds, slider plus exact field) determines which pauses qualify; shorter pauses are kept. Detection runs automatically on opening and after changing the level or duration; **Detect silences** lets you run it again explicitly. The summary shows the selected interval count and seconds to remove. Removing inner silence joins the remaining audio in source order; edge-only removal leaves interior pauses untouched. The original is retained as a recoverable version. Select at least one checkbox; no changes are applied by preview alone. The main Library waveform retains the original recording and automatically overlays the decoded processed

result in purple. Both use the same full-scale amplitude and source-time ruler. The purple result is mapped back to source time, with empty gaps where silence will be removed rather than shifting later speech to the wrong source position. Changes update after a short pause, without a refresh button. You can zoom, scroll and position the playhead while the panel remains open. Invalid trim settings and encoder/decoder/source-change errors are reported; outdated previews are removed rather than displayed as current. The preview summary identifies the settings and duration. It is a visual comparison, not a loudness or true-peak measurement. It does not play audio; Library preview playback still uses the original until changes are applied. Previewing never applies or retains an audio version; **Apply changes** remains a separate confirmed render using the current settings. Temporary preview audio and its waveform cache are removed on replacement, failure or panel close. Cancel removes the overlay and leaves the original unchanged. Selecting another track or opening another project cancels the staged processing; an obsolete result cannot be applied to the new selection. Rendering is cancellable, and encoder/decoder/source-change errors are shown. Review the new detection after changing the threshold. Quiet intentional entrances and reverbs can be mistaken for silence; all-silent files are not automatically trimmed or normalized. **Apply changes** asks for confirmation and renders in the background; **Cancel operation** stops the work. Processing itself does not start show playback.

Applying changes points the same Library track, and therefore its existing cues, at a new file without overwriting the original. The track ID, metadata and markers stay unchanged, but markers, cue timing, subtitles, captions and licence details are **not retimed or verified**: review them and rehearse the result. Stop show audio, paused playback and pending cues before applying or restoring a version. Undo and redo of a track-file change are blocked while these are active. In **Restore**, select a saved path under **Retained versions** and choose **Restore selected version** to use it again. Save the project to retain the version list across reopening. A retained version can also be restored when the current recording is missing.

Cleanup protects version files of retained tracks. Cleanup of unused tracks or orphan files is permanent and clears undo history, so keep an independent project backup. Undoing the edit that created a version can leave its generated file unreferenced; cleanup can remove it. If the project, source or playback state changes while rendering, the result is not applied and any completed output is identified for later cleanup.

Example: For The Lighthouse Keeper, detect the silence around the bell, review the proposed start so its quiet first strike remains, then apply a FLAC copy and check the knock marker before rehearsal.

Give related media files readable codes such as **AMB-SEA**, **BELL** and **RADIO**. Put source/licensing information on the track and performance instructions on cues. Library preview checks the file, not a cue's fades, ducking or timing. Use Test fire for those settings.

9.4 Enlarge, scroll and audition the Library waveform

The **Waveform** panel now spans the Library's width below the track table and Details. Drag the horizontal divider above it to give the waveform more height; Details still scrolls when the window is short. Stereo recordings show separate **L** (top) and **R** (bottom) waveforms in Library, Cue Creator and the Show Player's active-cue strips. Mono recordings use one lane; recordings with more than two decoded channels currently show one combined lane, not individual channel routing. The drawing represents decoded sample peaks, not a live output meter. Existing waveform caches are reanalysed once to provide separate stereo lanes.

The app does not currently offer per-channel mapping or output-device selection for show audio. Actual channel layout, downmixing and output routing depend on Qt, the operating system and the connected equipment; check both channels on the intended sound system before a show.

- **Zoom in** magnifies the time axis from **1x** to **16x** in doubling steps. **Zoom out** shows more time again. The visible centre stays at the same time when using these buttons or resizing, except where the beginning or end limits scrolling. Alternatively, hold **Ctrl** and scroll the mouse wheel over the waveform to zoom in or out around the point under the pointer.
- Use the **horizontal scrollbar** below the waveform to inspect earlier or later parts while zoomed. The time ruler and playhead refer to the complete recording, not just the visible section.
- **Fit** returns to the complete recording at **1x**, with no scroll offset.
- **Click the waveform to position the playhead without starting audio**. If audio is already playing, clicking seeks without changing playback state. With the waveform focused, **Space** plays or pauses from the current position. **Pause preview** preserves the position; **Play preview** resumes playback. Space in a text field remains a normal typing action.
- Selecting another track stops preview and resets its zoom, scroll and playhead. Leaving Library also stops preview. The separate preview player never fires cues or changes show-engine playback. Its player and audio output are released with the Library view; pending import-check callbacks cannot outlive their scan.

Example: Select **House Ambience**, choose **Zoom in** twice for **4x**, scroll to a quiet passage, click to

position the playhead, then press **Space** to listen. Choose **Fit** to see the whole recording again.

These are temporary viewing controls, not saved track or cue settings. Zoom does not trim audio, change its level, or increase the cached peak-analysis resolution. Library clicks do not edit Cue Creator's fade or ducking handles.

9.5 Named audio markers

The Library's **Used** column uses the same symbols as subtitle Text status: a circled check for tracks used by a cue, and a warning symbol for unused tracks. Hover for the status in words.

Markers are named positions in a Library track's complete recording, shared by every cue that uses that track. They are saved in the project as the track's `markers` list, with a stable ID, name and `position_seconds`; older projects start with no markers. They do not change the audio file, cue start time or script links.

Select a track and wait for its waveform. Choose **Add marker...** above the waveform to open the floating marker editor at the centre of the visible waveform. Enter its name and adjust the position with a slider or enter seconds to the nearest millisecond. The timecode below shows hours, minutes, seconds and milliseconds; the numeric field's minus/plus buttons adjust by one millisecond. The slider is coarser on long recordings, so use the field for exact placement. The editor does not block the Library: move it aside and drag its preview marker by the handle at the top of the main waveform. Dragging updates the exact time field while moving; slider and field edits move the preview marker and scroll it into view if needed. This is a staged preview: **Cancel** leaves the marker unchanged; **Save** records the position as one undoable edit and selects the new marker. Adding a marker does not start or seek preview audio. Selecting another track cancels the staged marker without saving it.

You can still right-click a point to **Add marker** or right-click near a marker to edit or remove it. In Library's **Audio markers** tab, Add uses the preview position; Edit, Remove and Preview act on the selected marker. Double-click also previews it. Removing asks for confirmation; edits support undo and redo. Markers appear as vertical lines in the Library, Cue Creator and active-cue waveforms. The list shows their names and exact times. A replaced or shorter audio file can leave a marker beyond its end; edit its time before using it.

Hover over a marker line to see its name and time. In Library, the **Markers** lane above the waveform has a handle for each marker: click one to select it in the Audio markers tab, drag it to move it, or double-click it to edit its name and exact time. Double-click an empty spot in the lane to add a marker at that position. These actions do not seek or start preview playback. Zoom and scroll for finer placement; moving a marker is one undoable edit. If markers overlap, select one in the list before dragging, or zoom in to separate them. Press **Esc** during a drag to cancel it.

The Lighthouse Keeper demo includes illustrative rehearsal markers for the Gate, Light Rises, Storm, Climb and Dawn cues. In Cue Creator, open a cue's **Marker links** tab to see its marker linked to the

corresponding script scene. These five-second positions are examples, not verified sound or lighting timings; listen and adjust them for an actual production. Links are navigation references and do not automatically trigger cues or seek audio.

In Show Player, a sounding cue with markers has its own marker selector and **Jump to marker** button. Select the marker and confirm; the jump affects only that cue and can change sound abruptly. The audio backend must report the file seekable and the marker must be before its end. Jumps are refused while the show is paused or waiting, while the cue is fading, looping, crossfading or has automatic continuation, or when it has timed audience captions. Stop a run recording before jumping: recorded runs do not represent seeks. The log records a jump request, not proof of sample-accurate playback. Backend seeking and codec timing can differ, so rehearse the action before relying on it. Clicking the active waveform itself never jumps.

Example: Add a **Bell** marker at **12.500 s** in Library. In rehearsal, preview it from the marker list, then select **Bell** on the active cue and confirm the jump if that cue is eligible.

9.6 Link markers to script passages

In Cue Creator select a start cue, then open its **Marker links** inspector tab. Choose a marker from that cue's track and a script paragraph, and choose **Add link**. Repeat to associate several passages and markers with the same cue. These links are separate from the cue's ordinary cue-in script link. Each cue owns its own associations, even when several cues use the same recording. Long link descriptions wrap to the available width; scroll vertically to read the complete marker name and script paragraph.

Select an association and choose **Show link** (or double-click it) to reveal the passage in the script panel and scroll the waveform to the marker. A heading normally hidden in that panel is temporarily revealed. The waveform must be available; zoom in to see the position more closely. This navigation never plays or seeks audio, fires a cue, or changes the show.

If a track, marker or paragraph is deleted, or the cue's track is changed, the affected links remain visible as missing or belonging to a different track; they do not follow a similarly named marker automatically. To repair one, select **Remove link** and add the correct association. Script revision review includes these references and applies the approved paragraph mapping; an unlinked passage leaves a link needing repair. Duplicating or copying a cue copies its links independently. Reusable scene recipes omit these show-specific links and report that omission.

Example: Two cues use the same recording. Link its **Door** marker to the first scene on cue 1 and to a different passage on cue 2; showing either link only navigates to its own passage and the shared marker.

9.7 Repair missing audio without rebuilding cues

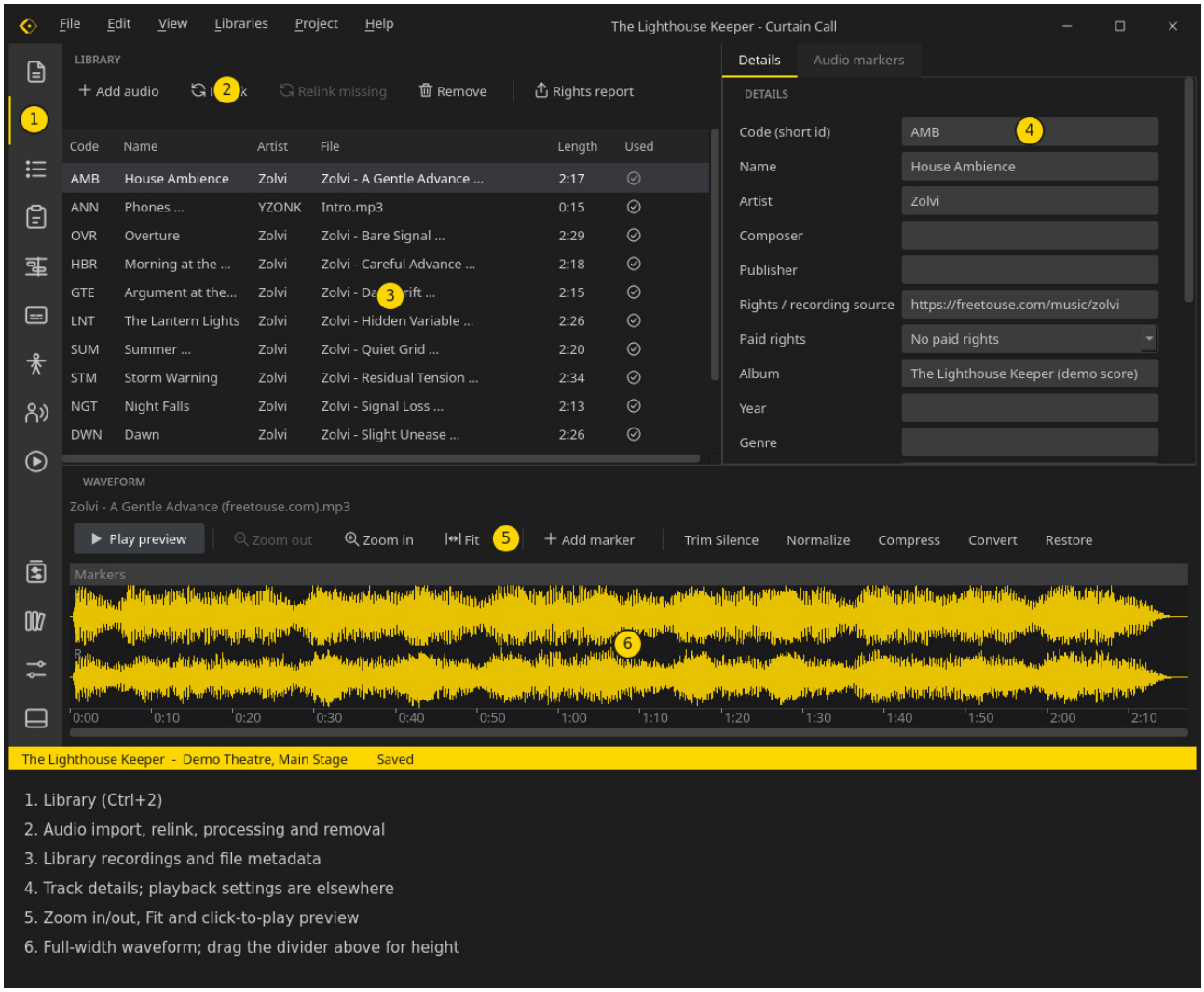
Select a missing track and choose **Relink** to select its replacement. The replacement is copied into the library if needed; the existing track identity and its cue references remain in place. Audition all affected cues after replacing a file: a different duration or edit can change the timing.

For many missing files, choose **Relink missing**, select a source folder and inspect the report. It searches recursively for **exact filename** matches. One match is used; zero or multiple matches stay unresolved. Copy failures are reported separately. It does not guess from audio content, similar names or track codes.

Example: a backup contains `Act1/bell.wav` and `Act2/bell.wav`. Batch relinking will not choose between them. Relink the correct recording manually and listen. If the stored name is `bell (2).wav`

but the backup has only `bell.wav`, use manual Relink rather than expecting a fuzzy match.

9.8 Identify the Library screen



1. **Library mode** is `Ctrl+2` in the activity bar.

2. **Toolbar** imports, relinks, processes and removes audio.
3. **Track table** lists the library, including missing-file warnings.
4. **Details** edits metadata in one form, not in playback tabs.
5. **Preview controls** play and zoom the selected recording.
6. **Waveform** displays the whole recording. The pictured preview is stopped.

9.9 Metadata is show documentation

Artist, composer, publisher, rights/recording source, paid-rights status, album, year, genre, copyright/licence and notes are stored in the project rather than written back into the audio file. **Use file info** fills empty metadata from the recording's tags; it does not overwrite fields you have already authored. Imported media files can populate tags automatically, but availability varies by file format and encoder. Tag values are not repeated in a second read-only summary below Details in either library.

A Copyright / licence entry is a reminder for the production team, not verification that you hold performance rights. The demo's written rights note is not a general licence for media files imported into your own show. Keep permission records outside the app as needed.

The waveform is peak analysis, not a calibrated loudness meter. It helps locate starts, quiet regions and tails; it does not establish the perceived output level. Analysis can run in the background across the library and is cached in `library/.waveforms/`. That cache may be rebuilt; it is not the only copy of the audio.

Tip: Keep track notes about the source recording (“warm low pad with distant water”). Put the performance call (“start when doors open”) in the cue-in note so different cues can reuse the same file with different instructions.

9.10 Media and performance rights reports

In Library Details, enter the **Composer**, **Publisher**, **Copyright / licence** and **Rights / recording source**, then set **Paid rights** to **Paid rights apply**, **No paid rights**, or **Unknown**. These are your statements, not legal determinations. Show Player preflight flags missing licence, composer, publisher, source and unknown paid-rights status for tracks referenced by start cues.

Use Library's **Rights report** button or **Project > Export > Media and rights (CSV)** to write `exports/media-rights.csv`. It lists each referenced track once, including disarmed cues, and flags missing facts and missing cue tracks. It is a planned-media inventory, **not proof that audio played**. The standard **Export all** also writes this inventory; performance-specific reports require selecting a recording.

For a presented show, use Show Player's **Record**, choose **Presented performance**, and start recording before any show audio starts. Stop recording after all show audio ends. Then choose Library's **Performance rights** button or **Project > Export > Performance rights (CSV)...** and select that performance. Each report belongs to one recording and is saved inside `exports/`; the app offers to open it, open its folder or start an email. It retains the track identity and metadata captured at playback start, even if the project is later edited.

The report counts **cue audio starts**, not individual loop passes. Engine playback seconds run from the engine's start request to its end, exclude pauses, and sum overlapping cue instances (so simultaneous uses can exceed elapsed wall time). Millisecond event timestamps and backend loading mean the values are not sample-accurate or proof of sound reaching an audience. Failures, recording started with audio already active, or stopped while audio was active are marked as incomplete. Editor auditions are excluded and flagged because they can affect show playback. Older run files lack playback boundaries and historical track snapshots; their reports explicitly say that attribution and duration are unavailable, rather than estimating them from cue fires. New run files use `format_version: 2`; older version 1 runs remain readable, but older Curtain Call releases may not understand the new event types.

Review the report and your licences before using it for a performing-rights return or programme. It is not a certified SPA form, does not submit anything, and cannot determine whether payment or permission is legally required.

Before **Remove**, check how the recording is used in cues. Replacing or removing library content can leave a cue without valid audio; preflight and the inspector are your next checks, not an assumption

that matching names preserve identity.

9.11 Required metadata: make the library fill itself in

A production can decide which metadata a recording must carry before it counts as library content. **Edit > Profiles > Metadata > Required metadata** holds the switch **Ask for required metadata when audio is imported** and one box per field; by default **Name** and **Copyright / licence** are required.

From then on:

- **On import** the inline Details panel selects the first incomplete file and names its missing fields. There is no separate metadata dialog.
- **In the library** a track that misses a required field gets an amber row and a tooltip naming what is missing, so an unfinished import cannot be forgotten.
- **Edit > Track metadata...** focuses the selected track's inline Details panel; there is no redundant Metadata toolbar button. The rules are not a lock: audio still plays, cues still fire, and nothing blocks a rehearsal.

Example: Import three storm media files for The Lighthouse Keeper with **Copyright / licence** required. One file has the licence in its tags, two do not. Details selects the first incomplete file and names its missing licence. Fill it there, then select the next amber row. The tracks you complete stop being amber in the list.

9.12 Metadata profiles: type it once, reuse it everywhere

A **metadata profile** is a named set of metadata values — the company as artist, the production's licence line, the genre you file effects under — that can be filled into any track. Profiles belong to this computer and this person, not to the project, because the same company metadata is reused across productions.

Open **Edit > Profiles > Metadata** to create them:

1. **From a track...** copies the metadata of a library track you already completed into a new profile (the quickest way to capture a licence line you typed once).
2. **New** starts an empty profile; type the values on the right.
3. **Duplicate** copies the selected profile; **Delete** forgets it. Tracks that already used a profile keep their values.
4. **Use this profile as the default for newly imported audio** fills its values into every new track automatically — only into empty fields, so tags already read from the file are never overwritten.

Both Library and AD library Details have the same inline profile controls. **Apply profile** applies the selected profile's nonempty values, with confirmation before replacing existing fields; fields not set by the profile remain unchanged. **Import profile...** reads Curtain Call JSON/ZIP metadata profiles into personal preferences; select the imported profile and Apply profile to use it. **Save metadata** saves the project, including metadata and other pending edits. Fields retain their existing edit-as-you-type and Undo behavior. **Save as profile...** stores the values you are looking at as a new reusable profile immediately, independently of whether you apply the track edits. If preferences cannot be saved, the app reports the error and the profile is available only for the current session.

Tip: Keep one profile per production company and one per venue's licence wording, rather than one giant "everything" profile. A profile only fills what it sets, so small profiles combine well: apply the company one, then the licence one.

Profiles and the required-field rules live in `preferences.json` on this computer. Removing a project, moving it to another computer or sending it to a colleague does not carry them: the colleague's own metadata rules apply.

Nothing here writes tags back into the audio files. The library metadata stays inside the project file, so a file you import is never modified by Curtain Call.

10 Cue Creator: fades, ducking and waits

Example: Inspect demo cue **1 House Ambience**: **70%**, **5 s Smooth** fade-in, endless loops, **5 s** loop crossfade and hotkey **a**. Then inspect cue **2 Phones Announcement**: selected ducking targets cue 1 at **0.25**, attack **0.2 s**, release **4 s**. These settings belong to the cues, not AMB or ANN.

Cues are ordered instructions. Start plays a track; fade changes another cue's level; stop stops its target; stop all stops everything. Memo cues hold operator text, and placeholder cues reserve a number until their instruction is written.

Each cue owns its volume, fade curves, loops, loop crossfade, ducking, pre-wait, post-wait, continue mode and optional hotkey. Disarmed cues remain in the show but GO, hotkeys, groups and automatic continuation skip them.

Example: start ambience with a slow fade in, then make a dialogue cue duck that ambience while it plays. Use the duck preview to check the result. **Test fire** auditions one cue; auditions do not automatically run followers.

Select several rows to edit shared settings. Copy, paste or duplicate cues to reuse them. Drag fade boundaries or the ducking band in the waveform to adjust their timing and levels. Undo and redo restore project edits.

10.1 A wider track waveform

The selected cue's **Track waveform** spans the width below the script panel, cue list and properties, like the wider Library waveform. Drag the horizontal divider above it to give the waveform more height or return space to the panes. The cue-property tabs remain independently scrollable.

Drag the fade boundaries to edit the cue's fade times, or drag an enabled ducking band to edit duck gain and attack. These are cue edits and can be undone. Use **Zoom in**, **Zoom out**, or **Ctrl+mouse wheel** over the waveform to magnify it up to **16x**, and use the horizontal scrollbar to move through the track. Wheel zoom keeps the time under the pointer in place where the track ends allow it; the buttons and resizing keep the visible centre. **Fit** returns to the whole track. Changing the selected audio file resets the view; editing a cue on the same file leaves the zoom in place.

Zooming, scrolling and resizing do not change the show or start playback. Unlike Library's audition waveform, clicking here does not start preview audio; use **Test fire** to audition a cue. Zoom does not increase the resolution of the cached waveform analysis or trim the audio.

10.2 Reach cue properties in a short pane

Every cue-property tab, including **Captions**, scrolls vertically when its pane is too short for the fields. The tab names remain above the scrolling content, so you can switch tabs without scrolling back up. Scroll inside the current tab to reach its lower fields; the waveform stays outside that scroll area. Fields keep their normal size rather than being squeezed into unreadable rows. In narrow panes, form labels move above their fields as needed. Ducking's **All**, **None**, **Copy to all cues** and **Preview** actions move into the toolbar's overflow menu when they do not fit side by side. Scrolling or changing the pane size does not edit the cue or start audio.

Example: in **The Lighthouse Keeper**, select **Phones Announcement**, open **Ducking** and scroll down to the target list and preview buttons on a short laptop screen. An intentional **Preview** or **Test fire** still plays audio; scrolling to the control does not.

10.3 Choose the instruction, then its target

Type	What happens when it runs
start	Starts the selected library track using this cue's settings
fade	Changes a currently playing target cue to Volume % / fade-to over Fade out duration
stop	Fades a currently playing target to zero, then removes it from playback
stop_all	Cancels pending waits and stops sounding cues using this cue's fade-out duration
group	Starts all armed start cues whose Group equals Group name to fire
delay	Waits Delay seconds, then schedules the next armed cue
memo	Displays operator text without starting audio
placeholder	A numbered slot whose instruction is not written yet: plays nothing, keeps the numbering, and is listed by the preflight

A fade or stop target is the **start cue**, not its library track. If that cue is not currently playing, the target action has no effect. Fading to zero is not the same as stopping: a silent faded cue can still be running and looping. Use stop to finish it. Firing a start cue already playing restarts that cue; distinct start cues using one track can overlap.

A group starts matching members directly: each member uses its own audio, volume, fades and loops, but its pre-wait and continuation are not run as independent GO operations. The group cue's own timing governs the sequence. Use a unique, consistent group name and test the group as a unit.

10.4 Placeholders: reserve the number before the cue exists

A **placeholder** is a cue whose instruction is not written yet. Add one from **Add cue > Placeholder** whenever a cue number must exist before its sound does: while the cue list is being planned, while a recording is still being made, or while you wait for a designer's decision.

- It plays nothing. GO fires it, logs "placeholder cue 4 Storm: nothing written yet, no sound" and moves the playhead on, so automatic continuation and groups behave exactly as before.
- Its row is drawn on a cool slate tint with diagonal hatching and the **State** column reads **PLACEHOLDER**, so it never looks like a cue that will play.
- It keeps its number, name, group and script link, so renumbering, the printed cue sheet and the Show Score need no scribbled gaps.
- The printed cue sheet and the next-cue card show **"Placeholder — nothing written yet"**, and the Settings column reads **to write**.
- The **preflight** names every placeholder each time you enter the Show Player: "Cue 4 'Storm' is a placeholder: its instruction is not written yet". That is your to-do list before the performance — replace the placeholder with the real cue (change its Type in Identity) or disarm it deliberately.

Use the **Memo** field to say what is missing ("needs a thunder recording"), and use the Cue in / Cue out notes for what the operator must know meanwhile.

- Cue **10 Sea**: start AMB-SEA, volume 60%, fade in 4 s, Loops 0 (infinite), loop crossfade 2 s, Continue none.
- Cue **20 Bell**: start BELL, volume 75%, Loops 1, Continue none.
- Cue **30 Blackout**: stop, target **10 Sea**, fade out 3 s, Continue none.

Fire 10 to establish the scene. GO 20 on the scripted bell action; the sea continues underneath. GO 30 on blackout to fade the sea away. No follower fires automatically because Continue is none. The bell ends naturally.

Loops **1** plays once; **3** means three total passes, not three extra repeats; **0** loops indefinitely. Loop crossfade overlaps adjacent passes and is limited to half the file duration by the engine. With one pass it does nothing. Listen for seams: a waveform does not prove the loop is smooth.

A start cue's fade-out setting is not an automatic fade at the file's end. For a planned exit, add a stop cue with the required target and duration. Fade curves shape the transition, not the track itself.

10.5 Pre-wait, post-wait and continuation

Pre-wait delays **this cue's action after it is fired**. Post-wait delays **firing the next armed cue after continuation is scheduled**. Post-wait is not a pause applied to every manual GO.

Continue	When the next cue is scheduled
none	No automatic follower, except for delay cues
auto_continue	As soon as this cue's action starts
auto_follow on start	When its audio naturally reaches EndOfMedia
auto_follow on other types	Immediately, like auto_continue; not after their fades finish

The next cue's own pre-wait then applies. The next armed cue is resolved from the current list after the post-wait; disarmed rows are skipped.

Situation	What happens
No following armed cue exists when the continuation would be scheduled (last cue, or every later cue disarmed)	Nothing is scheduled and nothing is logged; this is the normal end of a sequence
A follower existed when scheduling, but it was deleted or disarmed before the post-wait ended	The continuation is cancelled with the warning "Automatic continuation cancelled: no following armed cue remains." in the output panel

An infinite loop has no natural end for auto_follow. Stopping or fading out and removing its playback does not count as EndOfMedia and does not trigger that start cue's auto_follow.

10.5.1 Worked timing examples

Timed example: cue A is a 5 s recording with Pre-wait 2 s, Post-wait 1 s, Continue auto_follow. Cue B has Pre-wait 3 s. GO A at time zero: A starts near 2 s, finishes near 7 s, B is fired near 8 s and starts near 11 s. With auto_continue instead, B is fired near 3 s and starts near 6 s, while A is still playing. These are nominal timings; media loading and device scheduling can add latency.

Fade example: a 4 s fade cue with auto_follow and Post-wait 1 s schedules its follower after 1 s, not after 5 s. To wait for the fade before the next action, set Post-wait 4 s (or longer as needed) or insert a delay.

A delay cue schedules continuation after its Delay seconds even if Continue is none. Its pre-wait runs first and its post-wait runs after that delay. For a delay with Pre-wait 1 s, Delay 3 s and Post-wait 2 s, its follower is fired about 6 s after GO, before that follower's pre-wait.

Stop all and Panic cancel tracked pre-wait, delay and post-wait timers. A stop_all cue can itself schedule a new follower if its Continue is set; use Continue none for a terminating stop_all cue.

While a cue counts down its pre-wait, **another GO or hotkey for that same cue is ignored** (the log says "extra GO ignored"); it never starts a second copy and never cancels the first, so a nervous double press is harmless. Use Stop all or Panic to cancel a waiting cue deliberately. When the wait ends, Curtain Call looks the cue up again by its identity and runs it **as it is now**: edits or an undo made during the wait apply, a cue disarmed during the wait is skipped, and a cue deleted during the wait is cancelled with a warning in the log.

10.6 Ducking: let a foreground recording lower the background

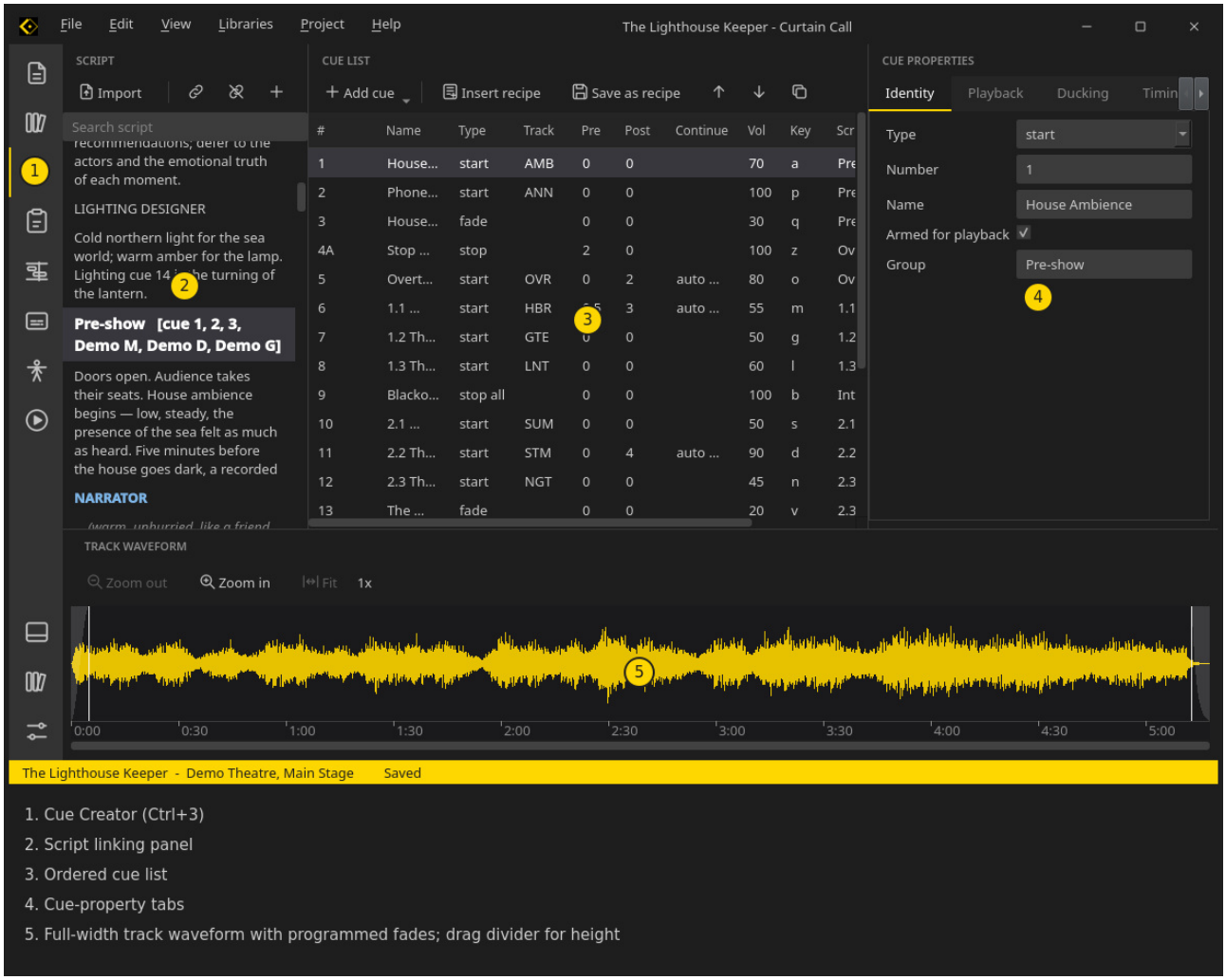
Configure ducking on the **foreground start cue**. Choose selected targets or all other playing cues. A cue never ducks itself. Project ducking defaults can supply gain, attack and release; turn off that option for cue-specific values.

Example: cue 10 Sea is playing at 60%. On cue 40 Radio choose selected ducking targets, select cue 10, set target gain **0.25**, attack **0.5 s** and release **2 s**. When Radio starts, Sea moves toward one quarter of its programmed level (60% × 0.25 = 15%, before master/live trim). When Radio ends or is stopped, Sea returns toward its unducked level over the release. This is triggered by active playback, not microphone input or speech detection.

Multiple active ducking sources do not multiply their reductions: the **lowest gain wins**, with the quickest active attack and longest active release. Muting a foreground cue does not remove it from active playback; stop it if you intend to release its ducking. A fade to zero likewise leaves its ducking active until playback is removed.

Use **duck preview** to hear a target alone and then with the source. Then test the actual sequence in Show Player. Target gain is a multiplier, not a decibel value; perceived loudness also depends on the squared output curve, master level, fades, live trim and the recording.

10.7 Identify the Cue Creator screen



1. **Cue Creator** is `Ctrl+3`.
2. **Script panel** selects paragraphs and provides Import, Link/Unlink and Add cue.
3. **Cue list** is ordered by performance sequence, not alphabetical cue name.

4. **Property tabs** are Identity, Playback, Ducking, Timing and Notes.
5. **Waveform** draws the track and the selected cue's fade/ducking envelope.

10.8 Build, reorder and reuse instructions

Use **Add cue** to choose Start, Fade, Stop, Stop all, Group, Delay, Memo or Placeholder. Give each new instruction a readable number, name and cue-in note. A number is free text, so the demo's **4A** can sit between 3 and 5 without renumbering the script. The list order determines “next cue”; numeric-looking names do not change that order. Every inspector field explains itself on hover, which is the quickest reminder of what **Pre-wait** and **Post-wait**, or **Duck attack** and **Release**, will do.

Drag rows or use Move up/Move down to rearrange cues. Duplicate reuses a selected instruction; Copy/Paste allows selected rows to be reused together. Inspect target and ducking references after cloning complex sequences. **Renumber** gives the list sequential numbers and is a production edit, not a cosmetic table setting. **Remove** deletes selected instructions and can be undone.

When selecting several rows, shared fields can be edited together. A mixed-value indication means that the selected cues do not all have the same value; changing it deliberately applies the new value to the selection. Confirm which rows are selected before changing a volume, target or continuation shared by a group.

An armed cue can execute; a disarmed cue is retained as documented material but is skipped by normal GO, individual hotkeys, group firing and automatic followers. Disarming is useful for an alternative effect you do not want accidentally called. It is not a live mute of an already playing recording.

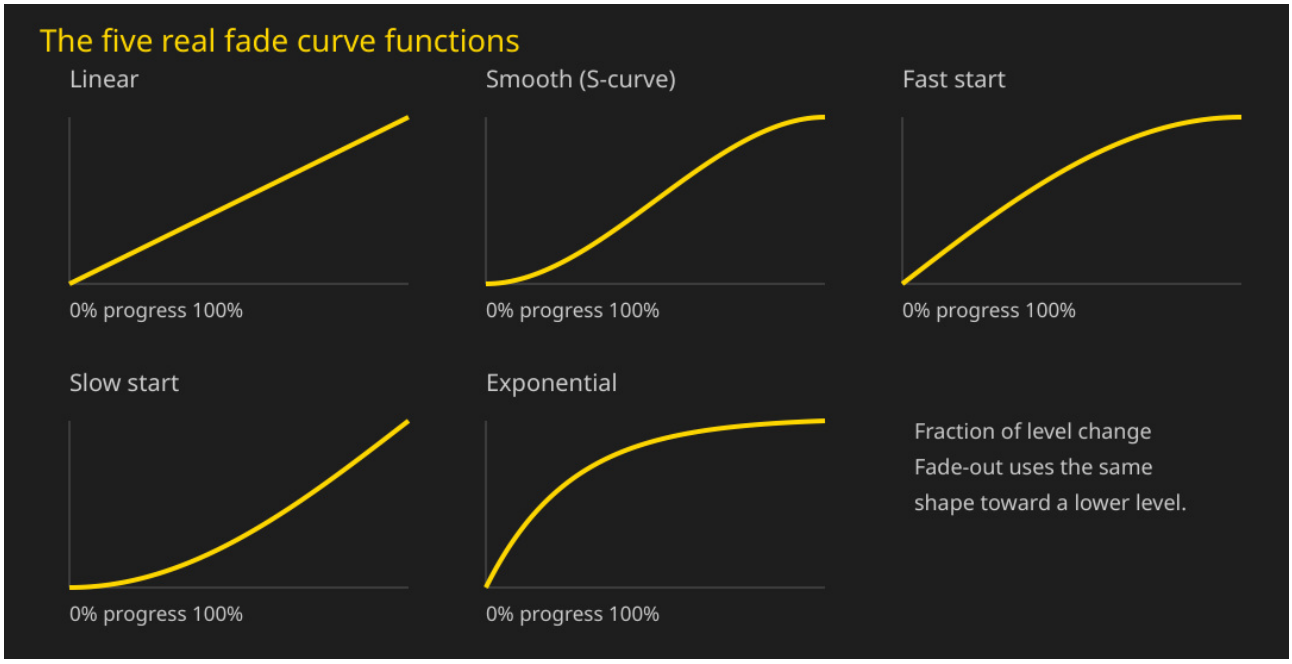
A disarmed cue looks switched off, in the Cue Creator and in the Show Player alike, so a silent GO is never a surprise:

- its row sits on a flat neutral tint instead of the editor background;
- its text is dimmed and drawn in italics;
- the **State** column reads **DISARMED** instead of READY;
- hovering that state cell explains what is skipped.

If the playhead rests on a disarmed cue, the next-cue card adds **"switched off, GO skips it"** to the action line. A cue that already fired keeps its **DONE** mark even if it was disarmed afterwards. Ticking **Armed for playback** again brings back the normal look immediately — the appearance is only ever

drawn from the cue, never stored separately.

10.9 See the actual fade shapes



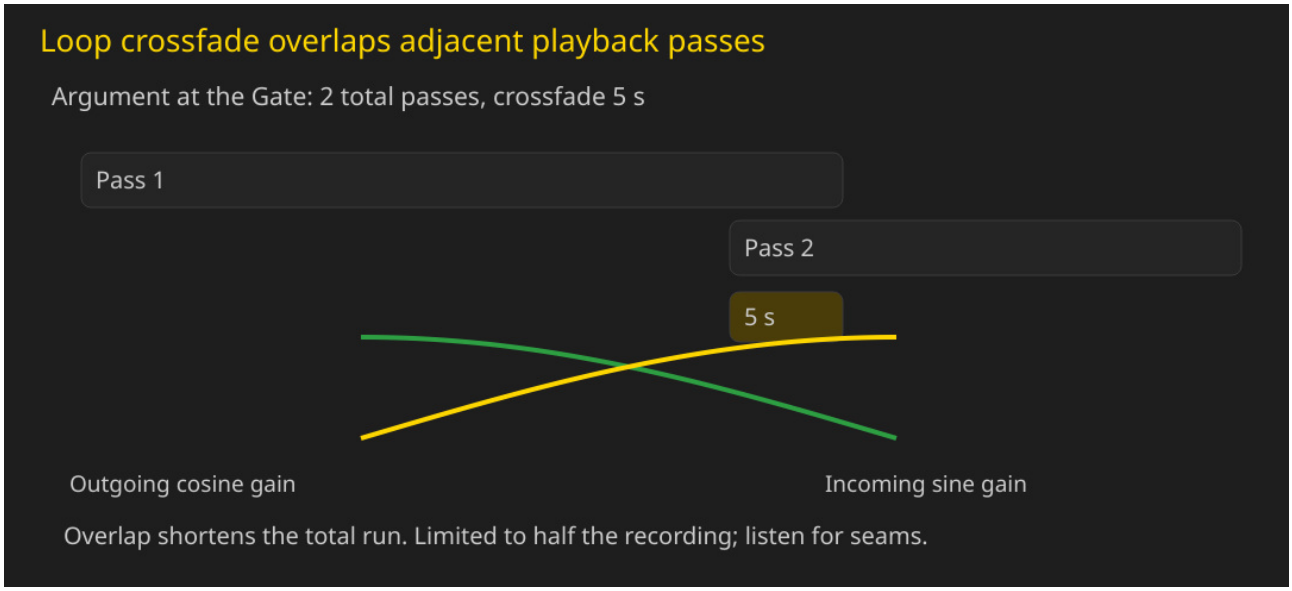
Curve	How the fraction of the level change moves
Linear	Equal progress through the change over equal time
Smooth (S-curve)	Gentle start and finish, quicker through the middle
Fast start	More change early, easing toward the end
Slow start	Less change early, increasing toward the end
Exponential	Quick initial change with a long approach to the target

The same curve can rise or fall. This drawing is progress through a level change, not a recording's waveform or a prediction of perceived loudness. The engine combines the fader with master, ducking and live trim before the output curve. In the demo compare House Ambience's Smooth entry with its Exponential exit, and the Fast start entry of **The Light Rises**.

Dragging the fade boundary edits the cue's seconds, not the underlying file. Fade in applies to

starting playback. For a dependable planned exit, target that start cue with a Fade or Stop instruction. A Fade to 30% is not a Stop at 30%.

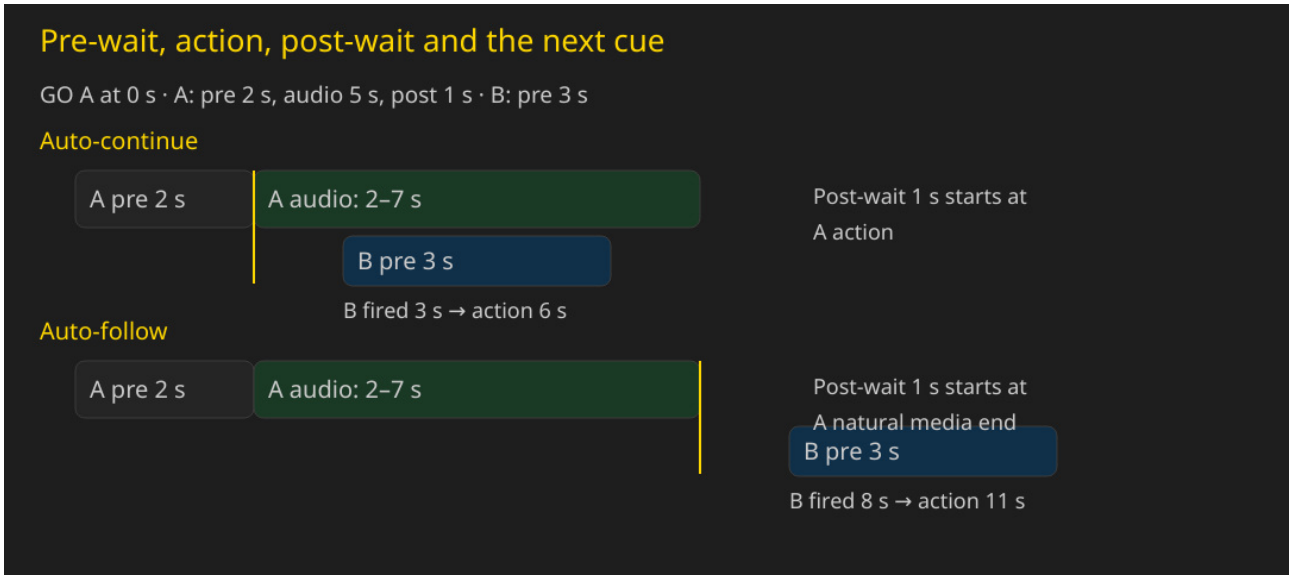
10.10 Understand counted and seamless loops



For a recording of duration D and overlap C, two passes run for approximately **2 × D − C** rather than 2 × D. This matters if an auto-follow depends on the last natural media end. More total passes add further overlaps. The engine limits C to half the recording duration even if a larger field value was entered. The final counted pass does not create an extra repeat merely to crossfade.

The demo **1.2 The Gate** requests two total passes with 5 s overlap. House Ambience loops without a natural final end. A clean-looking waveform does not guarantee a musically smooth seam; listen to the loop on the actual system.

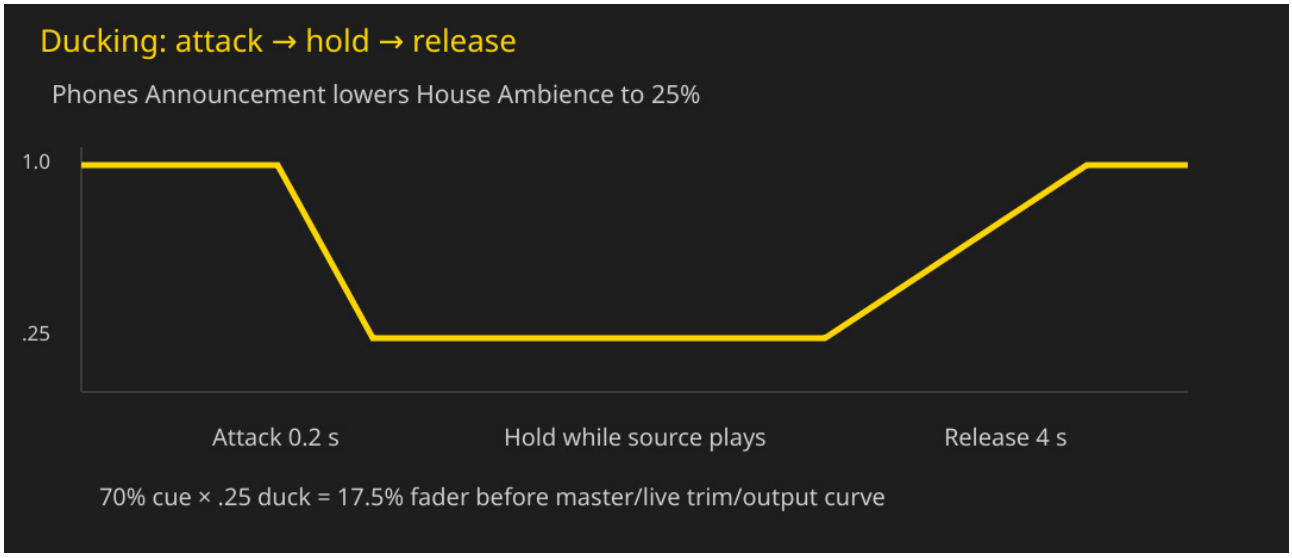
10.11 Draw the timing before running a chain



The chart reproduces the earlier nominal **11 s** auto-follow example. Gold boundaries indicate when a follower is scheduled; B still waits its own 3 s before its action. Auto-continue overlaps B with A; auto-follow on a finite start cue waits for natural media completion. Non-start auto-follow is different: it does not wait until a Fade or Stop envelope finishes.

Warning: An endless start cue has no natural EndOfMedia for auto-follow. Use an explicit Stop or a separately rehearsed operator cue; do not wait for an infinite ambience to end on its own.

10.12 Visualise foreground ducking



On the demo **Phones Announcement**, 0.25 means that AMB is reduced toward one quarter of its current unducked fader level. Attack is **0.2 s** and release **4 s**. The source holds the reduction for as long as it remains active, not for a fixed voice-detection interval. You choose selected targets or all others.

The ducking target list offers All/None selection and Copy to all cues controls; review their scope before applying. Defaults can be inherited from the project, but waveform edits resolve those defaults into explicit cue values when needed. The duck preview lets the target begin alone for 2 s, then brings in the source for an 8 s comparison before fading both. It plays real audio and suppresses automatic followers; it is not the silent What-if simulator.

Note: “lowest gain wins” describes several concurrent ducking sources. It is not an audio limiter, microphone sidechain or guarantee that overlapping tracks will not clip.

11 Show Player and the live script

Example: Before The Lighthouse Keeper begins, select cue **1 House Ambience** and read its caption and doors-open cue-in note. GO starts the programmed 5 s fade. Later press **p** for cue **2 Phones Announcement** only at the agreed five-minutes-to-curtain call; ambience remains active underneath.

Show Player is the locked performance view. The selected row is the next cue: Space or GO fires it, and Up/Down moves the playhead without firing.

The Now playing strips show sounding cues. Live trim temporarily changes a cue's gain; 100% uses its programmed level, and Reset returns to that level. The master volume belongs to project settings.

Use the right-edge icons to hide/show the script, switch script following on or off, or collapse fired cues. Memo rows use a subtle diagonal hatch, even when selected, so they stand apart from audio cues. The script follows fired cues' linked lines. Manual scrolling suspends following; press the **follow cues** button in the right-hand rail (the page with a downward arrow, directly under the script tab) to return to the latest linked line. Its pressed state shows whether following is on, and the choice is remembered for the next show. Browsing the script does not move the playhead or fire a cue.

Esc stops all audio. **Shift+Esc** is Panic. Use F11 for full screen.

In **Edit > Preferences > Interface**, enable **Hide the top bar in full screen** to hide the top File/Edit/View menu while any main-window mode is full screen. With the custom frame, the entire title bar also disappears, so the space is available to the show. Tap **Alt** by itself to reveal the bar temporarily; choose a menu command, dismiss a menu, click elsewhere or tap Alt again to hide it. The left-hand activity bar stays visible. Leaving full screen restores the menu; the preference is off by default. Menu shortcuts, GO and emergency shortcuts remain available, though an open menu can receive keys until dismissed.

The Show Player header is one line. It includes **Venue…**, the active trim, headroom warnings, subtitle languages and the audio check. It also shows the show time, wall clock, recording and **More**. The **Drop pin**, **Drop note**, **Lock** and **Full screen** buttons are always visible on this row. The controls have more space between them when the window is wide. On narrower windows these four buttons use icons; hover over an icon to read its name and purpose. A narrow, busy header may also use icons for Venue, subtitles and Record to make room for safety warnings. **More** also offers these actions, New performance, and timer and venue-trim resets. Choose **More > Keyboard shortcuts…** for the keyboard reminder; Preferences > Show Player can hide that menu item. **Ctrl+Space** freezes the show

in place and carries on from there. Check missing-file and playback-error messages in the output panel before a show.

11.1 Before the audience enters

1. Save and keep an independent copy of the project.
2. Enter Show Player to run preflight. Open output with `Ctrl+J`; resolve missing audio, absent targets and conflicting cue hotkeys.
3. Listen to the opening cue through the correct output device and amplifier. Preflight checks project references; it does not verify speakers or routing.
4. Check infinite ambience has a stop cue and timed chains finish safely.
5. Put the playhead on the first armed cue, confirm its script link and read the "what this cue does" summary.
6. Stop the test audio, reset any temporary trim and begin from the agreed cue.

11.2 Project health and audio checks

Open **Project > Project health** for the same project checks that Show Player logs as preflight. The dashboard lists missing and outside audio, missing cue, script, marker, rehearsal-note and subtitle references, unsupported cue values, rights information to review, hotkey conflicts and endless loops whose auto-follow cannot normally finish. Select an issue and choose **Go to fix**, or double-click it, to open the associated editor. Refresh checks the current project; navigation does not fire cues or change the Show Player playhead.

Library silently decodes each newly imported or replaced file in the background and reports failures above its track list. On entering Show Player with no active audio, waits or pause, the header offers **Run scan**, **Not now** and **Project health** in the same row. On a narrower window, choose **Check?** in that row to open the audio check menu and find those actions. The compact button changes to show progress or a result; open it for the full message and Project health. Run scan decodes Library audio without playing it; **Cancel scan** stops the check, not the show. GO, Stop all and Panic remain available throughout. For the least disruption, finish the scan and audition the show before the audience enters. A scan already running can continue if GO is pressed.

The dashboard reports decoder failures and counts files without a current result. Results are held for this app session and invalidated when file size or modification time changes; they are not saved to the project. Cancellation, missing files, changed files or a decoder error do not count as a successful check. Decoder success means only that this backend produced audio samples; it does not prove the entire recording is intact, speakers and routing work, or that a performance will be safe. Listen to important cues on the actual equipment.

Example: A start cue loops forever with auto-follow. Project health points to its Timing tab; choose a different continuation or plan an explicit stop, then rehearse the sequence.

11.3 During the run

Read the next-cue row before each GO. Up/Down changes the playhead without playing audio; selecting a later cue does not stop earlier sounding cues. Automatic continuation fires followers even while you are viewing other parts of the show. Do not add a manual GO into a rehearsed chain unless that overlap is intentional.

The Now playing strips expose active state, fades and loop progress. Live trim is a temporary multiplier: **100%** means the programmed level, not maximum output. Reset returns the multiplier to 100%. It does not rewrite the cue's saved volume or cancel its programmed fade.

Stop all (`Esc`) uses the project's default fade and cancels pending waits. Panic (`Shift+Esc`) cancels waits and cuts playback immediately without a fade. Neither is Undo. After either action, assess where to resume and deliberately select the correct next cue. Starting the same start cue again restarts it.

Manual script scrolling suspends following so you can inspect upcoming lines. The follow cues button in the right-hand rail returns to the latest linked line; unlinked cues cannot reveal a script location. Close help and focus the Show Player before relying on Space.

11.4 Pause and carry on

Pause (the bottom bar button or `Ctrl`+`Space`) freezes the show where it is:

- every sounding cue stops at its position, without fading out;
- running fades, ducking and loop crossfades stop counting and keep their level;
- a pre-wait or post-wait that was counting down holds its remaining time.

The header shows **PAUSED**, the affected rows read PAUSED and the button becomes **Resume**. The status bar at the bottom of the window always shows **PAUSED** (and how many cues are held), even from another mode, so a forgotten pause is hard to miss. Pause again (or `Ctrl`+`Space`) and the same cues, at the same positions, carry on: the fade finishes the seconds it still owed, the held wait runs out, the loop crossfade resumes.

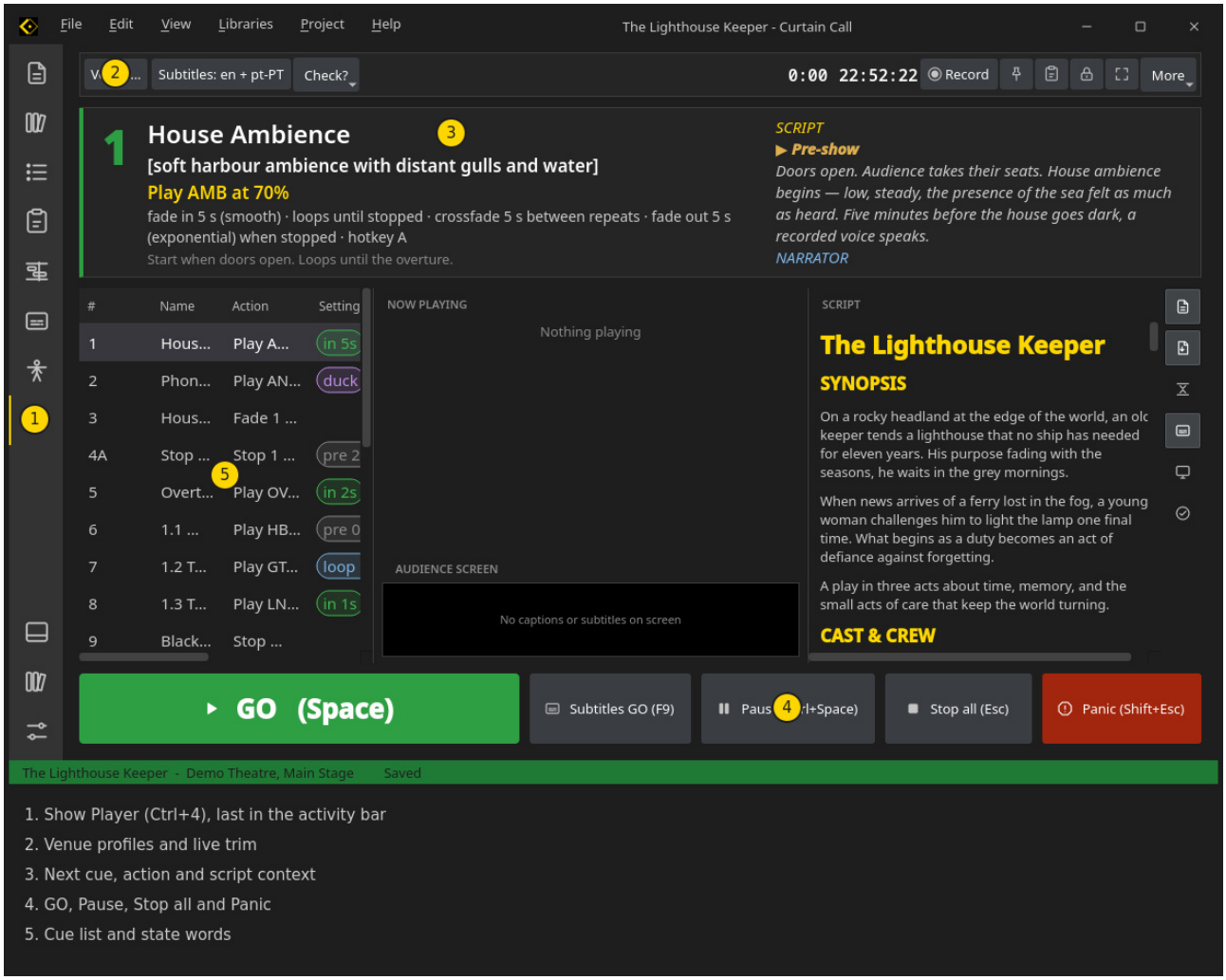
Example: In the middle of cue **1 House Ambience**, the stage manager asks for a two-minute break. Press **Ctrl+Space**: the ambience and its 5 s fade hold exactly there, and the next cue (with its 3 s pre-wait) does not start. After the break press **Ctrl+Space** again and the show continues as if nothing had happened.

Pause is not a second Stop all: nothing is stopped or lost, and nothing is restarted from the top. While paused:

- GO and the cue hotkeys are refused (the log says so) and the playhead stays where you left it, so a stray key cannot start a cue nobody hears;
- editor previews (audition) are refused too;
- **Stop all**, **Panic** and a stop-all cue still work, and they end the pause: the fade out is heard normally. After that, choose the next cue deliberately as after any Stop all.

To lock the screen itself while the sound keeps playing, use **Ctrl+Shift+L** (see *Lock the screen*).

11.5 Identify the performance screen



1. **Show Player** is `Ctrl+4`, the last icon in the activity bar.
2. **Venue…** opens venue profiles; the single-line header shows active trim, timers, recorder, bookmarks, screen lock and full screen.

3. **Next-cue card** shows the number/name, authored sound caption, action, settings, cue-in instruction and linked script context.
4. **Bottom controls** provide GO, Pause, Stop all and Panic; do not confuse their scopes.
5. **Cue list** and **Now playing** occupy the lower panels; the read-only script and its vertical icon rail sit at the right.

11.6 Read state words, not just colours

Pre-wait means the cue is fired but its action is scheduled for later. Playing means a voice exists, even if its gain is currently zero. Fading includes a countdown; muted and ducked describe output modifiers. Paused means the show is frozen on purpose and the cue holds its position; the header also shows **PAUSED**. A file failure removes that voice from active playback and leaves an error reason on its cue row until retried. Fired rows dim and can be collapsed without deleting cues.

The next-cue action explains whether GO plays a recording, changes a target, stops everything, starts a group or presents a memo. Settings chips summarise waits, fades, loops, crossfade and ducking; row tooltips give a longer description. Read the cue-in note and marked linked paragraph, not just the cue number.

11.7 End of show

After GO fires the **last armed cue**, the playhead is cleared and the next-cue card reads **End of show**. This is the same whether fired cues are collapsed or not, and trailing disarmed cues do not count. Further GO presses do nothing (the log notes "end of show"), so the last sound can never restart by accident. Returning to the Show Player from another mode keeps the end state.

To play a cue again on purpose, click its row or press **Up** (which selects the last visible cue), then GO. Down does nothing at the end of the show.

11.8 The playhead follows its cue through edits

The selected row is the playhead, and it is tied to the **cue itself**, not to a row number. Reordering, renumbering, inserting cues above it, and Undo/Redo leave it on the same cue. If the cue at the playhead is deleted, the playhead moves **forward** to the first armed cue that followed it (skipping disarmed cues and collapsed fired cues), and the log says so. It never moves back to an earlier cue that may already have been played. If no armed cue followed it, the player goes to End of show and the log shows a warning: select a cue before pressing GO. Opening another show always starts at its first cue.

Situation	What GO does
A cue is selected	Fires it, moves to the next armed cue
The last armed cue was just fired	Shows End of show; nothing is selected
End of show, GO pressed again	Nothing (logged)
End of show, a row clicked or Up pressed	That cue is selected and GO fires it

11.9 Use Now playing without rewriting the production

Each active strip shows name, time, loop pass, state, mini waveform, level and Mute/Fade/Stop. Newer voices appear above older ones. Its live-gain control is a temporary multiplier on programmed output; the Reset control returns it to 100%. The shown level includes fades, ducking, master and mute, not an acoustic meter at the audience's seats. Muting keeps a voice running; Stop removes it.

When a targeted fade reaches its programmed target, the cue is still active. That is useful for lowering a music bed, but a zero-level looping voice still uses playback and can keep its ducking source active. Stop it when finished.

Tip: Collapse the script with the right-edge Script icon only when you need more space. The icon remains accessible so you can reopen it without navigating to another mode or changing the live playhead.

Clock is the current wall-clock time. Show timer starts on the first GO and has its own **More > Reset show timer**. Neither is a cue's elapsed time, and resetting the timer does not rewind audio or reset the cue list. The recorder has a separate run-start origin.

11.10 Prepare a new performance

After sound check or a performance, stop all audio and pending cues, wait for fades to finish, and stop any run recording. In the Show Player header, choose **More** > **New performance…** and confirm. It clears fired-cue marks and playback-error indicators, returns the playhead to the first armed cue, resets the show and subtitle clocks and timing-capture progress, clears the audience screen, and unlocks audience subtitle language selection. The next armed GO starts a fresh subtitle pass. If no cues are armed, nothing is selected.

The action is blocked while audio, scheduled waits, pause or a recording are active; it does not stop or save them for you. It does not change cue settings, subtitle timings, the active venue or its trim, saved runs, or the project undo history. **More > Reset show timer** still resets only the timer and subtitle pass, not the whole performance.

11.11 Screen lock: cover the screen, keep the show playing

Lock the screen when the laptop must stay where it is but nobody should touch it. Press **Ctrl+Shift+L**, click the lock button in the Show Player header, or use **View > Lock screen**. The show does not stop: audio, waits, fades, loops, crossfades, ducking and automatic follow cues all keep running exactly as programmed, because the cover never talks to the audio engine.

While the cover is up, the cue list, menus, title bar, script panel and output log are hidden and every key and shortcut is swallowed: GO, Stop all, the mode switches, F11, Ctrl+S and the cue hotkeys do nothing. Only two things still work on purpose:

Key / button on the cover	What happens
Esc or **Unlock**	Asks for the operator password (when one is set) and uncovers the window
Shift+Esc or **Panic**	Stops all audio immediately, without a fade, and stays locked

Panic is deliberately outside the password: silencing a runaway cue must never depend on remembering a password.

Choose whether unlocking asks for a password in **Edit > Preferences > Saving & safety**: tick **Ask for a password to unlock the locked screen**, then press **Set password...** and type the password twice (at least four characters). With no password, the locked screen opens with Esc alone, which is fine for a private rehearsal room and dangerous in a busy foyer. Use **Change password...** to replace it and **Remove password** to go back to password-free unlocking; removing asks for confirmation. The password is stored as a salted PBKDF2 hash (`screen_lock_password_hash`) for this computer only, never in the `.ccshow` project, and it is never shown again, logged or exported. Preferences refuses to save with the box ticked but no password set, so a half-configured lock cannot happen. If the stored hash is ever damaged by hand-editing the preferences file, the lock still opens with Esc instead of locking you out.

A wrong password keeps the cover up, counts the attempt in the session log ("Screen stays locked: 2 wrong password attempts"), and leaves the show untouched. Cancelling the unlock dialog also leaves the cover up. Closing the window while locked asks for the unlock first; it is not a way around the lock.

Example: The Lighthouse Keeper runs on the stage-manager's laptop in the wings. During the interval the operator presses **Ctrl+Shift+L** and walks away with the preshow music still looping; the front-of-house manager cannot skip a cue or change the mode. When the operator returns, **Esc** asks

for the password and the same playhead is waiting where it was left.

Warning: The screen lock protects the app against accidental touches, not against a determined person. Anyone who can edit `preferences.json` on that computer can remove the stored hash, and the operating system can still switch apps, sleep or shut down. Use the operating-system lock or a locked room for real security.

12 Rehearsals: Drop pin versus Drop note

Example: Create **The Lighthouse Keeper — Dress**, iteration **Run 2**. Select cue **11 The Storm at Sea**, Drop note **Storm covers the radio lines**, and record **Listen from the back row; verify 20% ducking** in Details. Leave it Open until checked. A separate Drop pin **Paused for lighting reset** belongs in the session log, not the correction backlog.

Drop pin in the Show Player header opens an optional text prompt and adds a timestamped bookmark with cue/script context to the session log. It is not a project note. Export the log to preserve bookmarks outside the app session.

Drop note creates a persistent rehearsal note. It pre-fills the selected cue, its track and the selected script line in the expanded panel, falling back to the cue's linked line when needed. It asks you to create a rehearsal first if none exists; cancelling does not add a note.

Example: during a dress rehearsal, click Drop note, write "Bell enters too early", and add details such as "Wait until the visitor reaches the door". Use the status Open, In Progress or Resolved to track the correction.

Notes are grouped under dated rehearsals with an optional iteration/run. References can point to a cue, script line or library track. The **Go to** button beside a reference opens that exact location without firing audio or moving the Show Player playhead. Deleted references remain marked as missing.

Use **Export report** for UTF-8 text or **Export PDF** for a printable, numbered A4 report in `exports/rehearsals/`. Both include details, timestamps, statuses and references, including missing-reference warnings. **Send by email** exports the text report and opens an email draft with it (attached where your system allows, see *After every export*); nothing is sent automatically. Ctrl+4 returns to Show Player without stopping active cues.

12.1 Keep notes usable in a short window

The rehearsal picker and toolbar stay at the top. Below them the **note list** sits on the left and the **selected note** on the right. Drag the divider to share the width between the two; neither side can be collapsed completely.

The right-hand panel scrolls vertically: **Text**, **Extra details**, **Status** and all three reference selectors and **Go to** buttons remain reachable even when the window is short. The two text editors keep a usable minimum height; their own scrollbars handle long text, while the surrounding panel's scrollbar

reaches the status and references. The note list remains visible while you do this, so you can move to another note without scrolling the form back up.

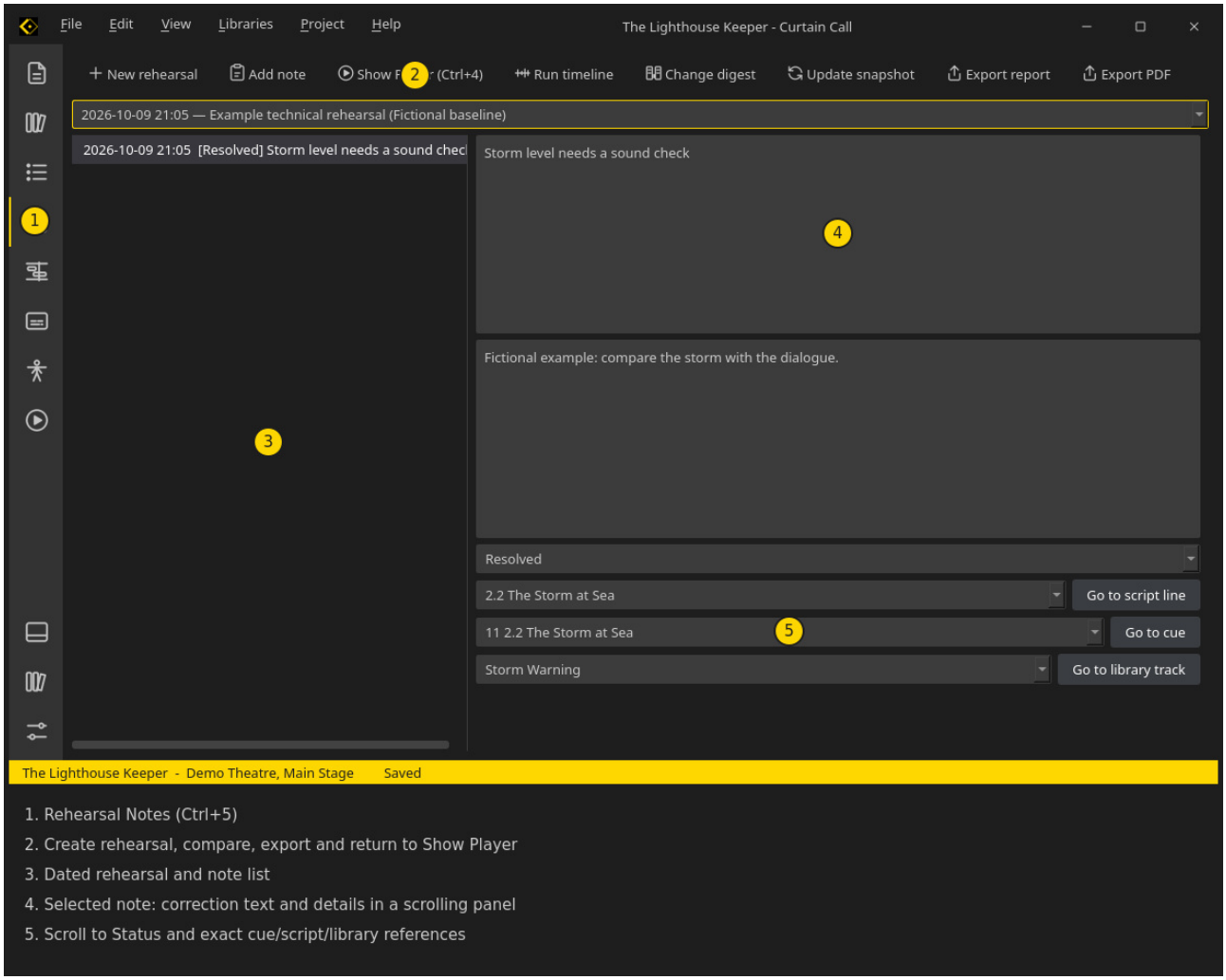
Example: open **The Lighthouse Keeper — Dress** on a small laptop, select **Storm covers the radio lines**, add details on the right, then scroll that panel down to **Go to cue**. Changing the layout does not edit a note or affect audio; typing still uses project Undo/Redo and `Ctrl`+4 returns to Show Player.

12.2 A practical correction cycle

1. Create a dated rehearsal such as **Dress — run 2**.
2. When the bell is early, use Drop note: title "Bell enters too early", details "Wait until the visitor reaches the door", status Open.
3. Check the references before saving the note. They describe the selected cue/track/script context, which may differ from the audio currently sounding.
4. Use Go to on the cue reference, adjust its triggering instruction or timing in Cue Creator, then mark the note In Progress.
5. Run the scene again. Mark Resolved only after checking the correction.
6. Save the project and export the rehearsal report for the director.

Drop pin is quicker for an event you only need in the run log, for example "Paused for lighting reset". Drop note is for a correction that should survive closing the app and be tracked through later rehearsals. A pin is not converted automatically into a rehearsal note.

12.3 Identify the Rehearsal Notes screen



1. **Notes mode** is `Ctrl+5`, before Show Score and the Show Player in the activity bar.
2. **Toolbar** creates rehearsals/notes, returns to Show Player, opens comparisons and exports reports.

3. **Dated rehearsal picker and note list** keep corrections in a rehearsal.
4. **Text and Details** separate a short issue from its explanation.
5. **Status and references** preserve exact locations and provide Go to navigation.

12.4 Name each rehearsal and its iteration label

New rehearsal records a recognisable name, optional iteration/run label and creation time. For example, use **Tech — Act 1** as the rehearsal name and **Run 2** as its label. The label is for searching and comparison context; each rehearsal remains its own project-level session. The app captures a baseline snapshot for later comparisons when the rehearsal is created.

Adding a note requires a rehearsal first. Drop note may ask you to create one; Cancel should mean no new note rather than an anonymous correction inserted into some older run. While writing, audio continues; return with `Ctrl+4` and restore focus before the next performance GO.

Status	Suggested team use
Open	Reported issue not yet addressed or verified
In Progress	An operator/designer is working on the change
Resolved	The team has tested and accepted the correction

Statuses are chosen by you. Changing cue volume does not automatically resolve a note, and deleting a note is not the same as resolving it. Keep useful details such as seat position, actor phrase and whether the problem occurred in the real system or a Library preview.

12.5 Check reference context when dropping a note

The selected cue may differ from the voice currently sounding. The visible, selected script line may also differ from that cue's saved anchor. Read the prefilled references so “Storm covers the radio lines” does not accidentally point at the following Night Falls cue. You can choose independent cue, track and script references; removed references remain explicit in reports.

Project Undo/Redo includes notes and rehearsals. The text editors use project history rather than their own unrelated local undo stack. Saving or exporting a report is not itself a status decision; make the status decision first.

13 Rehearsal change digest: compare snapshots

Example: Save a baseline for **The Lighthouse Keeper — Tech** (**Run 1**) with cue 11 at **90%**. In the later **Dress** rehearsal (**Run 2**) reduce it to **80%**, keep the note **Storm covers the radio lines**, and mark that note Resolved after listening. The digest should show the exact 90→80 change and note transition, not claim that the room now sounds correct.

In Rehearsal Notes, **Change digest…** opens a non-modal, read-only comparison. Choose **From** and **To**; the default is the previous rehearsal's snapshot to the **Current show / latest rehearsal**. Selecting the latest rehearsal as To also uses today's show. Older To rehearsals use their saved snapshots.

New rehearsal captures a lightweight baseline of cue fields, script text and scene structure, library identities and the note backlog through that rehearsal. Snapshots travel inside `project.ccshow`. At the end of a rehearsal, select it and use **Update snapshot…** to freeze the state you want to compare later. The dialog also offers **Update From snapshot…**. Updating requires confirmation, replaces that baseline with today's state and can be undone. It never changes cues or notes and cannot reconstruct an earlier state.

Notes are cumulative through each rehearsal and matched by stable ID, not similar wording. Existing notes can become Resolved or reopen; open notes stay in the digest. New and removed notes are explicit; deletion is not guessed to mean resolution. Cue additions/removals, field values (including effective ducking defaults), anchor changes and script paragraph changes are compared locally. Missing references remain explicit. An older project without a snapshot displays a warning: historical cue/script and status changes are unavailable, although new backlog notes can still be listed.

Group by **Scene** or **Department**. Scenes come from script scene headings; unanchored or missing links use **No scene**. Departments are derived, not invented: a cue link means Sound, otherwise a script link means Script, otherwise a track link means Library, and unlinked notes mean General. Counts are report entries (one changed cue may have several field differences). Click an entry to use the usual silent Go to navigation; removed references show a warning. **Export text** writes `exports/rehearsal-digest-<from-id>-<to-id>.txt`.

Example: finish Tech run 1 with "Sea too loud" Open and update its snapshot. Create Dress run 2, reduce that cue from 100% to 70%, move its script anchor and mark the original note Resolved. Compare Tech to Current: the digest lists the exact volume/anchor changes and resolved note, grouped under their scene or Sound department. No audio is fired and no live playhead is moved.

13.1 Establish an honest baseline

A snapshot is the show state at the moment you create or explicitly update it. It is not a continuous journal of every edit, and updating yesterday's snapshot today cannot reconstruct yesterday's cue levels. The **Update snapshot…** action requires confirmation and is undoable because it replaces project history data.

A good rehearsal routine is: create a rehearsal, collect notes, test changes, update its snapshot when that state should become the next comparison baseline, then create the next rehearsal with the appropriate run/revision label. If an older project has no snapshot, the dialog warns about unavailable history instead of inventing field changes.

13.2 Interpret the comparison as facts

Report entry	Meaning
New/removed cue	Stable cue identity exists on only one side
Changed cue field	Saved or effective value differs, including relevant ducking defaults
Changed script/anchor	Paragraph text, scene structure or referenced destination changed
Resolved/reopened note	The same note's explicit status changed
Open backlog	A note still needs attention through the selected rehearsal
Missing reference/history	Required object or baseline is absent; the report says so

Department grouping derives Sound from a cue reference, Script from a script-only reference, Library from a track-only reference and General otherwise. It is not a configurable staffing chart. Scene grouping uses actual scene headings; **No scene** is meaningful when a note has no reliable scene anchor.

Clicking an entry performs silent navigation. It never verifies audio, changes a field or moves the performance playhead. Counts describe report entries, so three changed fields on one cue can produce more than one entry. Exported text freezes the comparison at export time and belongs with the rehearsal communication.

14 Flight recorder and run timeline

Example: Record **The Lighthouse Keeper — Tech Act 1** and **Dress Act 1**. Compare the fired occurrences of **cue 8 The Light Rises**, then scrub to the nearby lighting-reset pin. A **+8 s** gap change means slower pacing in the second run; it does not mean the recording or lighting was eight seconds late.

Click **Record** in the Show Player header before a rehearsal run. Give the run an optional name, choose the rehearsal it belongs to (or none) and press **Start recording**. A red **REC** readout shows the elapsed time. While recording, Curtain Call timestamps every GO press, every fired cue (including hotkeys and auto-continue), Stop all, Panic, Drop pin, Drop note and playback failures, together with the cues sounding or waiting at that moment. No audio is recorded and nothing asks a question during the run, so GO is never blocked.

Click **Stop** (stop recording) to save a readable `.ccrun.json` file in the project's working `runs/` directory. Runs do not add undo steps. For a single-file project, recording marks the project as modified: save the project to include the run in the `.ccshow` file. Closing the project or quitting finalises a run that is still recording.

Open **Project > Rehearsal run timeline...** or **Run timeline** in Rehearsal Notes to review runs. The **Timeline** tab lists the selected run's events; drag the **Scrub** slider to see silently which cues had fired, the last cue, what was sounding and the notes so far. Recorded actions are never replayed through the speakers and the live playhead is never moved. The **Compare** tab matches cues of Run A and Run B and shows each cue's time in both runs, the delta, the gap since the previous cue and the pacing change. Changes of 2 s or more are tinted (amber = slower in B, blue = faster); cues fired in only one run are marked. **Export CSV** writes the comparison to `exports/`, and **Delete run** removes a run file after confirmation.

Example: record **Tech — Act 1** on Monday and **Dress — Act 1** on Thursday. Compare them: cue 12 "Storm build" shows a pacing change of +8.0 s, so the scene before the storm ran slower in the dress. Scrub to that moment to read the pin you dropped there ("Actors waited for the door"), then open a rehearsal note for the director.

14.1 Separate event evidence from audio recording

Record run stores event offsets and snapshots of sounding/waiting cues. It does not record a WAV file, microphone, continuous fader movement or actual acoustic arrival time. A fire event says the engine accepted the cue action; playback failure events are separate evidence. Recorded actions are never replayed through speakers when you scrub.

Event	What it documents
GO	Operator pressed the performance GO control
Fire	Cue fired by GO, a hotkey, group/automatic execution as reported by the engine
Stop all / Panic	Operator emergency/stop action and its run time
Pin / Note	Operator bookmark or correction context
Failure	Playback failure reported during the recording

14.2 Compare the same dramatic passage

Use named media files and start them at the same agreed rehearsal origin. If the first run begins before doors open and the second begins at the overture, raw absolute offsets are not a like-for-like timing judgement. Compare gaps between corresponding cue occurrences as well as their run offsets.

Only-A or Only-B rows are genuine missing execution evidence, not silently matched against an unrelated cue name. Damaged run files are skipped and identified. A comparison CSV is a table of timing facts; keep the source run files with the project for later inspection.

Warning: Delete run removes a separate file after confirmation. Project Undo is not a recycle bin for runs because recorder files deliberately live outside the editable show's undo history.

Stopping recording saves the run file. Closing or replacing the project also finalises a recording still in progress. For a single-file project, save the project to include the run in its archive. For a folder project, copy the whole folder; `project.ccshow` JSON alone does not include run evidence.

15 Scene recipes: reusable cue patterns

Example: In a copy of The Lighthouse Keeper choose **Pre-show announcement with ducking**. Assign **House Ambience (AMB)** to Pre-show music and **Phones Announcement (ANN)** to Announcement. Review the new start/memo cues and internal ducking target before Insert cues. Use fresh blank hotkeys rather than colliding with demo **a** and **p**.

In **Cue Creator**, select the cue to insert after (or clear selection to append), then choose **Insert recipe...**. Choose a built-in or saved user pattern and fill its required track slots from the library. The live preview lists every new cue, including its numbers, action, continuation and internal targets. Nothing plays while you choose tracks or preview. Optionally link only the first new cue to the selected cue's script line, then approve **Insert cues**.

Example: choose **Scene opening: ambience + fade in**, assign **AMB-SEA** to **Ambience**, and review the looping start cue (5 s fade in) followed by a **Scene begins** memo. Insertion is one undo step: **Ctrl+Z** removes exactly the bundle; redo restores it. Existing cues, hotkeys and the Show Player playhead stay unchanged. New numbers skip occupied numbers rather than renumbering the show; new cue IDs are fresh and hotkeys are always empty.

Other built-ins crossfade A to B, duck pre-show music during an announcement, or fade all audio for a blackout. The crossfade bundle first establishes its own new A cue with GO; the next GO starts B and fades/stops that A, not an existing playing cue. Announcement music returns automatically when the announcement ends. **Blackout** really stops all audio when fired. Fade durations and start volumes marked as project defaults resolve at insertion; the scene-opening fade is explicitly 5 seconds. Check your media files and timings in rehearsal.

Select several existing cue rows and choose **Save as recipe...** (also available inside the picker) to name a user pattern. Tracks become placeholders; internal fade/stop and ducking links follow the new IDs. External links are dropped with visible warnings; missing fade/stop or group targets produce disarmed cues that must be repaired before playback. Hotkeys, cue numbers and script links are not saved. User group names are isolated from existing show groups. Required empty slots prevent insertion; optional empty track cues are warned about and disarmed.

User recipes live in plain UTF-8 **recipes.json** beside the app preferences, including in portable mode. Saving or deleting a recipe changes that file, not the show or its undo history. Delete requires confirmation; built-ins cannot be deleted. A damaged file shows an error without hiding the built-ins;

replacement requires explicit approval when saving a new pattern.

15.1 Choose a built-in by the operator sequence

Built-in pattern	What you assign	How you operate it
Scene opening: ambience + fade in	Ambience	GO starts an endless bed with 5 s entry, then an automatic Scene begins memo
Transition: crossfade A→B	Outgoing A and incoming B	First GO establishes new A; next GO starts B and automatically fades/ stops that A
Pre-show announcement with ducking	Music and announcement	First GO establishes music; second plays foreground announcement and ducks that music
Blackout: stop all with fade	No track slots	GO fades all playback with the default fade, then reaches a memo after the wait

A recipe is a starting pattern, not a hidden playback macro. Once inserted, its cues are ordinary editable cues with their own identities, defaults and links. The preview lists them before approval. If its timings do not fit your scene, edit the inserted bundle and test it in the same way as hand-authored instructions.

15.2 Save a user pattern responsibly

Select the whole self-contained sequence you want to reuse, including target start cues and related stop/fade instructions. Save as recipe and use descriptive track slots rather than baking an external recording path into the pattern. Expect fresh cue IDs, blank hotkeys and no saved script anchors on insertion.

External target/ducking links cannot be assumed to exist in another production. Warnings explain dropped links or disarmed incomplete instructions. Required track slots must be filled before insertion; optional empty slots still need review. A disarmed generated cue is a repair task, not successful playback.

User recipes are stored in **recipes.json** beside preferences, not inside a project folder. Move that file separately if a touring laptop needs your pattern library. Venue-profile library copies and recipes solve different problems: venue profiles adjust room gain; recipes create creative cue sequences.

16 What-if simulator: rehearse silently

Example: In **The Lighthouse Keeper**, start the snapshot at **cue 1 House Ambience**, enter `2 @ 12` and `4A @ 30`, then simulate. Skip **2 Phones Announcement** to see missing foreground audio and ducking; skip **4A Stop House Ambience** to see the endless background survive its baseline exit. These times are deliberate operator assumptions, not script-derived predictions.

Open **What if…** in the Cue Creator toolbar or **Project > What if…**. This non-modal window is **silent and read only**: it uses a detached snapshot, never calls the audio engine, moves the live playhead or changes the project. Reopen it after editing to take a fresh snapshot.

Choose the start cue (one operator GO at 0 s), the observation window and optional skip cue. Enter additional operator presses as **cue number @ seconds**, one per line, then choose **Simulate silently**. These offsets are your assumptions: manual GO times and actor pacing are never invented. Automatic continue/follow chains use programmed waits; groups start members directly. Skipping omits that cue from GO, group and automatic paths; it does not stop an already running voice or invent a replacement operator press.

Cue lanes draw fader curves, hatched waits, loop boundaries and duck attack/hold/release bands. The explanation list says why each action happens; the skip summary compares starts, stops, fades, ducking and fire offsets with the baseline. Scrub **Sounding at** for linear fader × duck × master gains. A script-line query uses its first executed linked cue as the time anchor; unlinked/unexecuted lines are unresolved, not estimated from dialogue length.

Example: cue **10 Sea** loops endlessly; cue **18 Announcement** ducks Sea; cue **30 Blackout** stops Sea. Start at 10, enter `18 @ 12` and `30 @ 30`, then skip 18 to see the missing announcement and ducking. Skip 30 instead to see Sea keep sounding after its baseline stop. Query the announcement's linked script line in the baseline to inspect the gains at that execution point.

Unknown library durations leave natural ends, loop boundaries and dependent auto-follows **unresolved**. Endless loops have no natural end. Drawings end at the chosen horizon (loop boundary drawing is limited to 2000 passes). This is an ideal timing model, not an acoustic prediction or file/playback preflight: it excludes device latency, media errors, timer jitter and live operator trims. A run from `runs/*.ccrun.json` can be selected for an occurrence-by-occurrence **fire-offset comparison** (before pre-wait; recorded minus simulated time). Use the same run-start origin. Missing cues are listed; no hidden alignment is performed. The recorder does not capture continuous gain or prove

audio accuracy.

16.1 Ask one bounded question

A useful simulation has a known starting cue, explicit operator press offsets, an observation horizon and one hypothesis such as “skip the announcement”. The simulator takes a detached show snapshot; reopen after edits instead of assuming an old window is watching live production changes.

1. Select the start cue and observation time.
2. Enter extra presses as cue number @ seconds, one per line. The demo's **4A** is a valid displayed number; the UI resolves it to the relevant cue identity.
3. Simulate silently and read explanations for starts, waits, fades and stops.
4. Inspect Sounding at a chosen offset; compare the baseline with a skipped cue.
5. Query a linked script paragraph only when an executed cue provides its time anchor.

16.2 Read uncertainty explicitly

Unknown recording duration makes a natural end unresolved. An endless loop has no final natural end even when its loop boundary can be drawn. Unlinked text has no operator time anchor, so the tool does not estimate its time by counting words. A horizon limits what you see, not how long a real cue can continue.

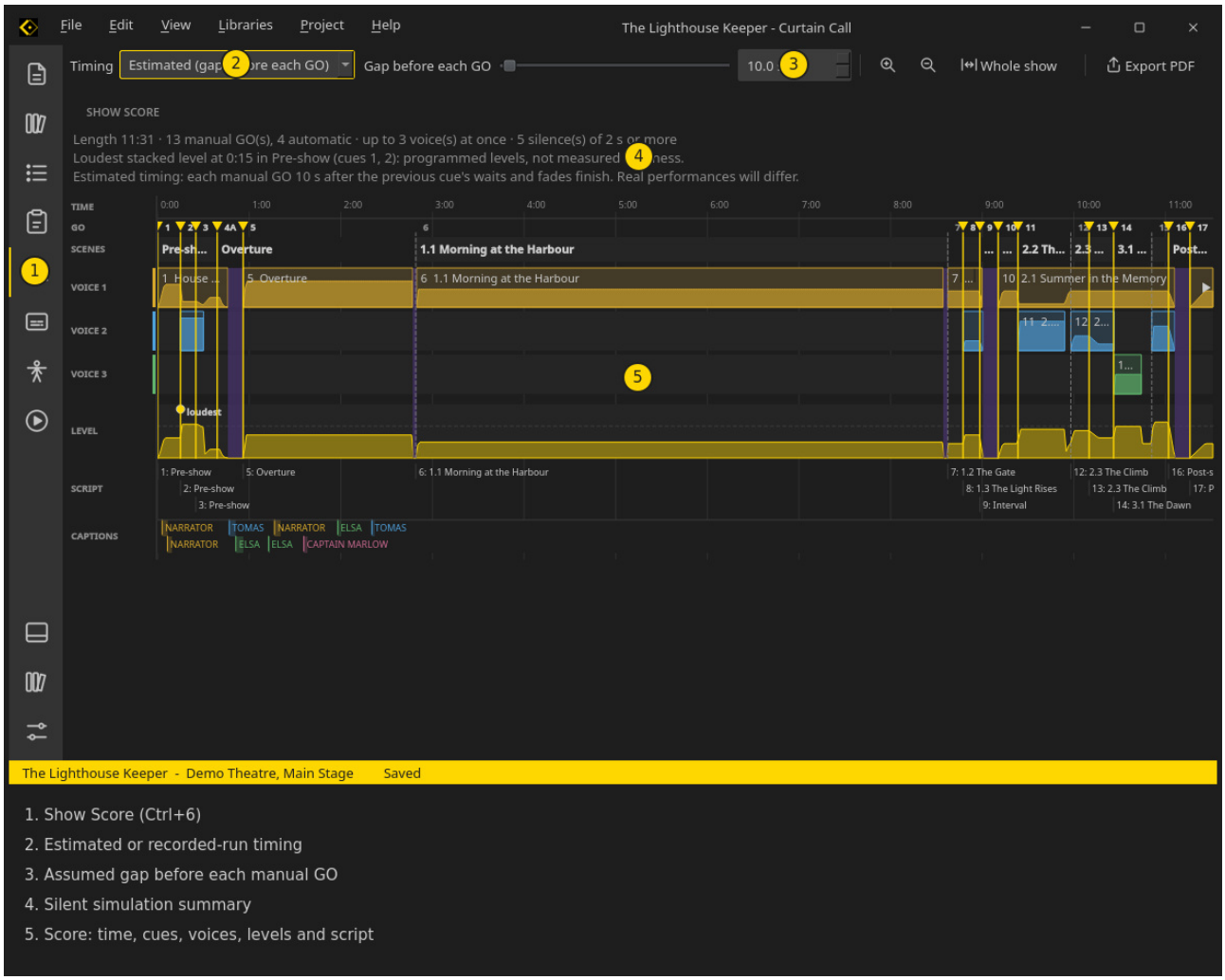
The display combines linear fader, duck and master values; it excludes live trim, physical routing and the room response. Ideal scheduling excludes media loading, Qt timer jitter and device latency. Flight-recorder comparison checks fire offsets at corresponding occurrences; it does not recover a continuous gain envelope.

Tip: Use What-if to explain sequence logic before playing audio. Then use actual preflight and a controlled rehearsal to verify files, levels and timing. A silent simulated voice is not proof that a codec will decode on the show laptop.

17 Show Score: the whole evening on one page

Example: In **The Lighthouse Keeper**, press `Ctrl`+6. With the estimated timing and a 10 s gap the score runs about 11 minutes: three voice lanes, 13 manual GO lines, four automatic follows, and the loudest moment in scene **3.2 The Bows**. Choose **Whole show**, then hold `Ctrl` and turn the wheel over the ending to zoom into that transition.

Show Score (`Ctrl`+6) draws the show like a musical score, read from left to right. It is **silent and read only**: it is built with the What-if simulator, never plays audio and never changes the project.



1. Show Score (Ctrl+6)
2. Estimated or recorded-run timing
3. Assumed gap before each manual GO
4. Silent simulation summary
5. Score: time, cues, voices, levels and script

The illustration identifies the mode, timing source, assumed GO gap, summary and score canvas. The demo's estimates depend on known file durations and do not predict an actor's pace.

From top to bottom the bands are:

Band	**What it shows**
Time ruler	Minutes and seconds from the first GO
GO row	A gold triangle and solid line for each operator GO; a dashed grey line for automatic continue/follow starts
Scenes	The scene each linked cue belongs to, taken from the script's Scene paragraphs
Voice lanes	One lane per sounding voice. Each bar is a cue's playing recording: fades are ramps, loops repeat, ducking shows as dips
Level	All voices stacked, with the loudest moment marked
Script	Text of the linked script lines at their cue's time

Purple columns mark **silences** of 2 s or more. A **hatched** bar means the recording's length is not known yet: the score asks for a waveform analysis and fills the length in when it is done. An **arrow** at a bar's right edge means the sound carries on past the end of the drawing, for example an endless loop. Small ticks at the top of a bar mark loop restarts. Hover a bar for its cue, track, start, end and level; double-click it to open that cue in the Cue Creator.

17.1 Choose the timing

- **Estimated** (default): each manual GO comes the chosen **gap** after the previous chain settles (waits, fades and automatic follows finished). The gap is your assumption about pacing, not a prediction of the actors.
- **A recorded run** from the flight recorder (`runs/*.ccrun.json`): the operator GOs are placed at the times they really happened. The gap box is then disabled.

The score refreshes shortly after each edit while the mode is visible.

17.2 Zoom and export

Use the zoom buttons or `Ctrl` + mouse wheel; **Whole show** fits the evening to the window width. **Export PDF** (or **Project > Export > Show score (PDF)**) writes `exports/show-score.pdf`: one A3 landscape page with a summary, a legend, the whole score and the usual header and footer with export date and app version.

Tip: The level band shows **programmed** level (cue volume × fades × ducking), not measured loudness. Two quiet media files can still be loud in the room, so confirm "the loud bit" by listening in rehearsal.

18 Sound captions and accessible operation

Audience captions make important sounds accessible in text for deaf and hard-of-hearing people and others with hearing difficulties. They can describe sound effects, music and other audio information that matters to the performance. They may also help people who cannot hear the sound clearly in a particular venue. The operator-facing **Sound caption** and the timed **Audience captions** are separate fields; writing one does not automatically create the other.

Example: Cue **11 The Storm at Sea** in The Lighthouse Keeper carries **[music for the storm scene]**. Compare that authored intent with the next-cue action and state word. Enlarge Show Player text to **20 px** for a substitute operator; do not rely only on a green row tint.

In Cue Creator, add a **Sound caption** to describe the intended sound in human terms. Captions appear prominently on the selected next cue and on each matching Now playing strip, and are included in the printable cue sheet and its CSV export. They are optional and can be hidden with **Edit > Preferences > Show Player > Show sound captions**. Cue state is also written as words such as **PLAYING**, **FADING** and **PRE-WAIT**, not just shown by row or strip colour.

Example: caption a storm cue as **[distant thunder rolls in, growing]** so a substitute operator can recognize the intended moment without decoding playback settings or hearing the recording. The caption is authored guidance, not a transcript or a guarantee that every recording matches it.

18.1 Write useful sound intent

A caption should tell a collaborator what the audience should experience, not repeat a filename or numerical fade setting. **[soft harbour ambience with distant gulls and water]** conveys a specific sound intention; a filename and level alone do not explain its dramatic function. Keep descriptions concise enough for the next-cue card while recording further operator instructions in cue notes.

Sound captions are authored manually and describe intended sound for the operator; they do not appear on the audience screen. The separate **Audience captions** cue field contains timed text for that screen, while Subtitles mode builds timed titles from script dialogue. Curtain Call does not recognise speech, generate subtitles automatically, transcribe media files or detect audience hearing needs. The Show Player preference controls sound-caption presentation on operator cards, and exports keep those captions as paperwork context.

18.2 Pick colours and keep reusable palettes

Every **Text colour** and **Background** row uses the same colour chooser, whether you are editing the audience caption board, the app's subtitle look, project overrides or **This title's look** in Subtitles mode. Type a hex value such as **#FFD700**, or select **Choose colour...** to open the full picker. For project or title overrides, the red reset arrow returns the colour to the value inherited from the level above. It appears only when a colour override is set.

The picker offers factory shades chosen for the dark interface (neutrals, gold and readable accents), your default palette and **Recently used** colours. Factory shades remain available even when your default palette is a custom one. Type six RGB hex digits or eight alpha-first ARGB digits in **Hex**, or use the **Red**, **Green** and **Blue** sliders and their 0–255 number boxes; the hex value and live sample stay in sync. An incomplete hex value disables OK until corrected.

18.2.1 Choose a shade visually

The picker also has a **saturation/brightness field** beside a large live colour sample and a **rainbow hue strip** underneath, like a graphics editor. Drag along the rainbow to pick the colour family, then drag inside the field: left is less saturated (towards white), right is richer colour, top is brighter, and bottom is darker (towards black). The circular markers show your selection; hex, RGB controls and swatches stay synchronized. **Opacity** adds an alpha channel from 0 (transparent) to 255 (opaque), with a synchronized slider and number field. Eight-digit colours use Qt's **#AARRGGBB** order, for example `#80FFD700` for half-transparent gold. **Transparent** is available in every picker's built-in palette and colour dropdown, including with a custom palette. Swatches and the large colour sample show partially transparent colours over black; only fully transparent colours use a checkerboard. Alpha values survive palette, profile and project saves; choosing Transparent is distinct from inheriting a colour. Press **OK** to accept either an opaque or alpha colour and add it to recently used colours.

Tab to the field and use **Left/Right** to change saturation or **Up/Down** to change brightness; hold **Shift** for larger steps. The hue strip also works with arrow keys, Home and End. Hue is kept when you choose black or grey, so moving back to a brighter or richer shade restores the chosen colour family. In a short window, scroll the controls to reach the palettes, hex and RGB rows; **OK** and **Cancel** remain visible.

Example: Choose green on the rainbow, then move halfway down the field for a darker green. Fine-tune it with the RGB boxes and choose **Keep in my palette**.

The last **12 distinct colours** are remembered on this computer, newest first. Only committed choices enter that history: loading a project or refreshing colour fields does not add colours. Choose **Keep in my palette** in the picker to stage that shade for reuse; it creates **My colours** if you have no personal default palette. Press **OK** to keep the choice and palette changes, or **Cancel** to leave both unchanged. A row set to **Follow the level above** keeps its inheritance when the picker is cancelled.

Edit > Profiles > Colour palettes manages palettes even without an open project: Choose a palette in the profile list on the left; its name, shades and default setting appear in the editor on the right. The editor scrolls when necessary.

- **New...** starts an empty palette; **Duplicate** copies the factory shades or a personal palette so you can adapt them.
- Type a **Palette name** and choose a shade using the integrated spectrum, hue strip, hex, RGB and opacity controls; no extra dialog opens. **Add colour** adds the chosen shade. Select an existing swatch to load it into the chooser, then use **Update selected colour** to replace it, or **Remove colour** to delete it. Colours and recent-colour history remain staged until the outer editor is saved; incomplete hex input disables Add and Update, and duplicates are reported. A palette holds up to **48 distinct shades**.
- Tick **Use this palette in the colour picker** to choose the default; select the factory palette to return to the built-in choices.
- **Delete** asks for confirmation. Factory shades cannot be edited or deleted.
- **Export selected profile...** writes the selected palette, including unsaved edits, to a JSON file. **Import profile...** adds a palette from such a file without replacing an existing palette; duplicate names are given a unique name. Import does not change the default until you choose it. Exported files remain on disk even if you cancel Profiles.
- Palette export names include the palette name, app version and build; the JSON also records the version and build. Import asks before accepting older or unversioned palettes and never changes the source file. Newer or invalid versions are reported instead of being silently imported.
- **OK** saves your palettes and default; **Cancel** discards the panel's edits. Select the factory palette to return to the factory default without deleting personal palettes.

Palettes and recent colours are **app preferences**, not project contents. Choosing a different palette does not recolour any existing caption or subtitle: their chosen colours remain in the app look, project

override or individual title where you set them. Project/title colour changes use normal show undo; palette management and recent-colour history do not change the show or audio playback.

Example: for **The Lighthouse Keeper**, duplicate the factory palette, name it **Lighthouse touring**, add **#E8C989** and mark it as the default. In the caption look choose that warm shade for **Text colour**, leaving **Background** black, then review the live sample and check readability on the real audience screen.

18.3 Arrange the audience-caption table

In **Cue Creator**, select a start cue and open its **Captions** tab. The table holds the caption text, **Start (s)** (from when the sound starts) and **Duration (s)** (0 keeps it up until the sound stops).

Describe relevant audio clearly, for example **[distant thunder grows louder]**, and time it to the sound. Check size, contrast, timing and screen visibility in the venue, ideally with deaf and hard-of-hearing reviewers. Audience captions are authored manually; an empty table does not provide captions automatically.

Drag a column edge to resize it or drag its title to reorder the columns. Right-click the header to show or hide columns, choose **Fit columns**, or use **Reset columns** to restore the default widths, order and visibility. Double-click a column title to fit the columns to their contents. At least one column stays visible so the header menu remains reachable.

This layout is remembered on this computer in the app's table preferences, not in the cue. Switching cues or undoing a caption edit keeps your arrangement; hiding or moving a column does not change the captions, their timing or playback.

Example: in **The Lighthouse Keeper**, select **Phones Announcement**, open **Captions**, widen the text column and move **Duration (s)** beside it. Hide **Start (s)** while reviewing the wording, then right-click the header to show it again when checking timing.

18.4 Choose the second screen panel size

The audience screen is the movable **second screen panel**, and its size belongs to the show: **Project > Project settings > Second screen**. Pick a **resolution preset** — HD 1280 × 720, Full HD 1920 × 1080 (the default), WUXGA 1920 × 1200, QHD 2560 × 1440 or 4K UHD 3840 × 2160 — or type a **custom** width and height between 320 × 180 and 7680 × 4320. The panel opens at that size every time it is shown, shrunk to fit the monitor if the size is bigger than the screen; drag it to place it, and use its menu for full screen.

Example: The Civic Theatre has an old 720p projector, so that show is set to **HD 1280 × 720**; the tour version keeps **Full HD 1920 × 1080** for the LED wall.

New projects start with the size chosen in **Edit > Preferences > New projects > Second screen panel**, so a venue only sets it once. A project always keeps its own value afterwards, and hand-edited sizes are clamped to the usable range when the project opens.

Right-click the panel itself and choose **Second screen size...** to jump straight to that project settings page.

18.5 Show translated subtitles to the audience

The Show Player's **Subtitle regions** button (with a layout icon) opens the subtitle region arrangement dialog. Its tooltip lists enabled languages. It opens directly without a confirmation prompt; playback safety locks still apply. The audience screen can show subtitles alongside its existing cue captions. Choose **View > Subtitle language layouts...** or right-click the audience screen and choose **Subtitle language layouts...**. The editor lists the original subtitle language and every language that has saved translation text. Use the checkbox beside each language in the left list to show it on the audience screen. Select its name to edit its layout, even when unchecked. Checked languages appear together, with no limit on how many you enable; a missing translation stays blank rather than falling back to the original.

When no layouts are saved, the starting arrangement uses one full-width bottom region; two full-width regions at top and bottom; three with two half-width regions at the top and one full-width at the bottom; or four half-width regions in a two-by-two grid. More than four use equal horizontal bands. The matching **Layout** icon restores this arrangement for the currently checked languages without changing their look or unchecked languages. You can then edit each region. Changing checkboxes does not rearrange custom positions, and saved layouts are never automatically replaced.

Square visual buttons beside **Left** and **Top** align the selected region to the screen edges or centre without resizing it. **Regions** offers equal horizontal/vertical gaps within the existing outer span and equal widths/heights based on the selected region. These affect checked languages only. Distribution requires at least three checked languages and is unavailable when the regions cannot fit without overlap; matching dimensions requires at least two languages and is limited so every region stays inside the screen. The **Layout** icons apply the corresponding 1–4-language preset when that many languages are checked. The equal-bands icon restores horizontal bands for five or more checked languages. There is no separate Arrange button; custom region edits are never automatically undone. The text alignment icons align words left, centre or right inside the selected region, not the region itself. The same icon row positions text at the top, middle or bottom. Clicking an active icon again returns that choice to inheritance. These icon-only rows replace the Alignment and Position fields; tooltips describe each action. Regions and Layout appear once above the screen preview as shared controls for the checked languages, separate from each language's editor. Only checked language regions are drawn. Unchecked languages remain editable through the list but have no outline, text or background in the preview. Gold dashed outlines mark the other regions; a solid cyan outline marks

the selected one, matching the selected-marker colour on the waveform. Outlines are editor guides only and are not shown to the audience.

The screen preview preserves the configured output width/height aspect ratio. Click a visible region to select its language for editing. Drag inside it to move it horizontally or vertically. The selected region has visible cyan boxes at its corners and edge midpoints: drag these to stretch or shrink it directly, keeping the opposite edge fixed. Regions stay inside the audience screen; direct resizing stops at 1% in each dimension. Even a full-screen region can be shrunk using its handles. The exact percentage fields and sliders update together with the preview. With the preview focused, arrow keys move the region by 1%; Shift+arrow keys resize its right or bottom edge by 1%. These edits remain staged until you accept the dialog; Cancel preserves the project. The boxes and outlines are editing guides only, never part of audience output. In the region editor, text size shows effective numeric percentages, and its slider follows those values even when inherited. Red arrows restore inheritance. Reset arrows appear only for changed values or explicit overrides and disappear after restoring defaults/inheritance; their space stays reserved to keep controls aligned. All profile-editor previews and subtitle previews without video use the same diagonal patterned backdrop as HoH previews, so transparent boxes are easy to see. Video-backed timing and recording previews keep the actual video instead. The pattern is an editing guide only, never part of audience output or the recording. Percentage font sizes scale with the full displayed screen height, not the height of a language region or the surrounding preview widget. Resizing the preview therefore keeps text proportional to the configured screen.

Each language has its own **Left**, **Top**, **Width** and **Height** as percentages. Width and Height always allow 100%; enlarging a region moves Left or Top back when necessary to keep it inside the screen. Moving Left or Top still reduces the region if needed to fit. All values are percentages of the audience screen. Typeface, bold, text colour and background follow the active app/project profile and title look; this dialog cannot override them. Existing saved language overrides for those properties are retained for compatibility but ignored on the screen and preview. Text size, alignment, and vertical text position can be overridden here. **Region opacity** fills the whole language rectangle using the profile background colour and defaults to 0% (transparent). Its slider and exact field are separate from the background behind the subtitle text block. The region reset returns to 0%. Region and typography settings share aligned labels, stretching sliders and exact numeric fields. Region alignment presets sit together on a separate row so they do not squeeze the position controls. **Region alignment** and **Text alignment** label their right-aligned button groups; horizontal and vertical controls have extra spacing between them. Preview **Regions** and **Layout** actions share one row. Subtitle editors do not show a

separate Box opacity row; choose the background's alpha in the colour picker. Existing saved opacity values and inheritance are retained for compatibility and still multiply the background colour's alpha. Editing the subtitle background colour makes its alpha authoritative (the legacy opacity multiplier becomes 100%); untouched saved looks retain their original multiplier. The background colour remains profile-controlled in the language-region editor. The audience screen background is unchanged. **Copy settings** copies those permitted settings from another language. When Hard of Hearing (HoH) styling is enabled, character colour and feeling effects, typeface and size take priority. Legacy layouts retain their regions and enabled state. Adjust the starting regions to suit your screen and avoid covering important sound captions. New shows use the starting arrangement on the live screen and preview. When saved layouts already exist, additional languages receive full-width bands without changing saved positions. Accepting the layout editor saves your choices in the project so they travel with it. An explicitly disabled language stays disabled.

18.5.1 Subtitle spacing, boxes and readability

Subtitle style editors place the preview above a compact two-column control grid when wide, with one column in compact inspectors. Related settings stay stacked in the same column: text on the left, background and readability effects on the right. Thin, neutral outlines identify related groups: Text, Background, Outline, Shadow and Timing where applicable. Each numeric row keeps its label, slider, exact field and conditional reset arrow on one line. Sliders stretch within their column so value fields and reset arrows remain justified and aligned. New-project subtitle rules use a compact grid too. Caption and HoH style previews likewise appear above their controls. Red arrows appear only for changed values or explicit overrides.

- **Line spacing:** 50–300% of normal font spacing; 100% keeps the existing spacing.
- **Background padding horizontal/vertical:** 0–100% of font height on each side. The default is 0%, preserving the existing tight box. The bracket to the right spans both padding rows; its central chain icon has distinct linked/unlinked states. It links the two controls: editing either slider or precise field sets both to the same value; a linked reset restores both. Linking alone does not change saved values. Unlink to adjust each independently. Numeric entries across the app offer minus/plus buttons beside the value instead of stacked arrows. Exact typing, keyboard stepping, ranges and inheritance are preserved.
- **Extra screen inset horizontal/vertical:** 0–40% of the available region on each side, in addition to the existing safe margin; both default to 0%. Edited only in Show Player's placement dialog, alongside Top/Middle/Bottom text position.

- **Maximum text width:** 10–100% of the remaining text area; 100% preserves the current width. Edited only in the placement dialog. Narrower text areas wrap while respecting left/centre/right alignment, which remains available in style editors.
- **Letter spacing:** -10–50% of text size added between characters; 0% is normal.
- **Background shape:** one box around the whole text block (default), or a box behind each line, word or character. Word/character boxes exclude whitespace; accented characters and joined emoji stay together. Wrapped words get one box per visible fragment. Padding, rounding and colour/alpha apply to every box. These backgrounds follow HoH word/character animation, including their padding.
- **Corner rounding:** 0–100% of font height; 0% keeps square corners. Each box limits the radius to half its shorter side, keeping corners circular, not oval.
- **Text outline width:** 0–20% of font height, with a separate colour/alpha picker. The default width is 0% (off).
- **Text shadow offset:** 0–50% of font height diagonally down/right, with a separate colour/alpha picker and 0–100% opacity. Offset 0% turns it off; default opacity is 60%.

These settings travel with subtitle profiles, projects and individual title overrides; untouched overrides inherit. Existing saved placement values are preserved; explicit placement-dialog choices take precedence over legacy feeling placement overrides. Language regions can adjust spacing and box geometry, while effect colours remain profile/title-controlled. Subtitle spacing settings remain independent when following caption typography. Previews and audience output use the same text-layout rules. HoH feeling outlines, shadows and backgrounds replace the corresponding neutral effect rather than stacking a duplicate. Existing profiles without these fields keep the defaults.

The audience screen and its Show Player preview show sound captions and subtitles together. While both are visible, captions temporarily use the top 30% of the screen and subtitle regions are scaled into the remaining 70%, so subtitles cannot cover captions. When either is absent, the other uses its normal area again; saved subtitle layouts are not changed. Subtitle languages can still overlap one another if their configured regions overlap. Text may shrink to fit the smaller areas, so check readability on the actual screen.

The larger preview on the right shows **all enabled languages together** on a screen with the project's configured aspect ratio and outlines the selected region. It uses the first original title and its translations together; a missing translation stays blank. If there are no titles, labelled samples are

shown. A disabled language's selected region is outlined but its text is not drawn. This is a sample, not the live audience output. Drag the outlined region up or down to change **Top** without changing its height. A full-height region has nowhere to move: reduce **Height** first. The percentage fields provide precise placement and keyboard access. The three icons beside **Left** align the rectangle left, centre or right; the three beside **Top** align it top, middle or bottom. Combine them for nine positions; they preserve the region's width and height.

Choose another language and press **Copy look** to copy its font, size, weight, colours and text alignment to the selected language. Its rectangle and enabled state stay unchanged. Choose **Apply app defaults** to stage saved defaults for matching languages, or tick **Save these layouts as app defaults on OK** to save the edited languages for future projects. Unmentioned app languages remain saved. New projects receive independent copies, including saved languages which are not used in that show yet. Existing projects do not change when app defaults change. To clear saved defaults, use **Preferences > New projects > Clear saved language layout defaults** and accept Preferences. Project settings' **Apply app defaults** and **Save as app defaults** also transfer language layouts, replacing matching languages and keeping the others. Changes in this dialog remain staged until **OK**; **Cancel** discards both project and app-default edits. The **Use Hard of Hearing styling** checkbox previews the same mode as Show Player's HoH button. Its selection takes effect in Show Player and audience output only on **OK**; **Cancel** leaves the live mode unchanged. HoH speaker and description text is added to the original language, not translations. The entire title uses its character colour unless feeling font colour is enabled. Feeling-coloured outlines, shadows, glow and backgrounds can be combined in all languages. Live feeling movement is off by default; when enabled in Project settings it appears on both the Show Player preview and audience screen. Changing audience language layouts is locked after GO until **New performance** and while audio or waits are active. Check the actual audience screen before a performance, including overlaps with other languages and sound captions.

The subtitle clock starts at its own button, **Subtitles GO** (**F9**), beside the main GO, and never at a cue GO. This lets the house-open ambience and pre-show music run while the audience finds their seats; press Subtitles GO when the performance proper begins. It can also be pressed before any cue. It follows the Show Player's **Pause / Resume**: pause freezes subtitle time and resume continues the same pass. **Stop all** and **Panic** clear the subtitle layer and end that pass. To start over, deliberately press **Reset** beside the Show Player timer, then press **Subtitles GO**; the next press starts a fresh subtitle pass at zero. Pressing it while a pass runs does nothing. There are no separate subtitle pause, seek or correction controls. The authored subtitle times are relative to the moment of Subtitles GO, so they can drift from a live performance.

Example: In **The Lighthouse Keeper**, enable en and `pt-PT` in Subtitle language layouts and leave the default separate bands. Open the audience screen, press GO, and verify that both languages change together. Pause and resume to confirm the titles hold their place; use Stop all to clear them, then Reset and GO to begin a new pass.

18.6 The audience screen and its narrow mirror

View > Audience screen (`Ctrl+Shift+C`) opens the screen the audience sees. It requests an always-on-top position without taking keyboard focus; the desktop's window manager ultimately controls window stacking. While you work, the Show Player keeps a **narrow landscape band** of that screen at the bottom of the Now playing column: it fills the column's whole width and is only **72 pixels** tall, so it shows the caption text in the screen's own colours and alignment without taking a second screen's worth of room. It mirrors the combined sound-caption and subtitle output for a quick check.

- Two buttons sit in the Show Player's **right-hand rail**, the vertical strip at the far right under the script tab: the **screen preview** button (below **collapse fired cues**) shows or hides the band, and the **screen** button (a small monitor, under it) opens or closes the audience screen itself, exactly like `Ctrl+Shift+C`.
- Both buttons show their state instead of words: the preview button is pressed while the band is on screen, and the screen button is pressed while the audience screen is open. Their hover text says what the next click does.
- Showing or hiding the band is the same setting as **Edit > Preferences > Audience captions > Show the audience screen preview in the Show Player by default**; set it there to choose how every show starts, and the output panel records "audience screen preview hidden".
- Neither button takes keyboard focus (Space stays GO), and hiding the band gives the height back to the Now playing strips.

18.7 Adjust the booth view before the performance

- Use Show Player cue font size between **12 and 28 px**; the default is 15 px.
- Enable high contrast when the standard table contrast is insufficient.
- Keep the keyboard reminder available in **More** for substitute operators.
- Resize/reorder/hide table columns to bring Action, Settings and State into view.
- Retain state words and authored notes rather than relying on colour alone.
- Maximise or use F11 when a small screen makes script context hard to read.

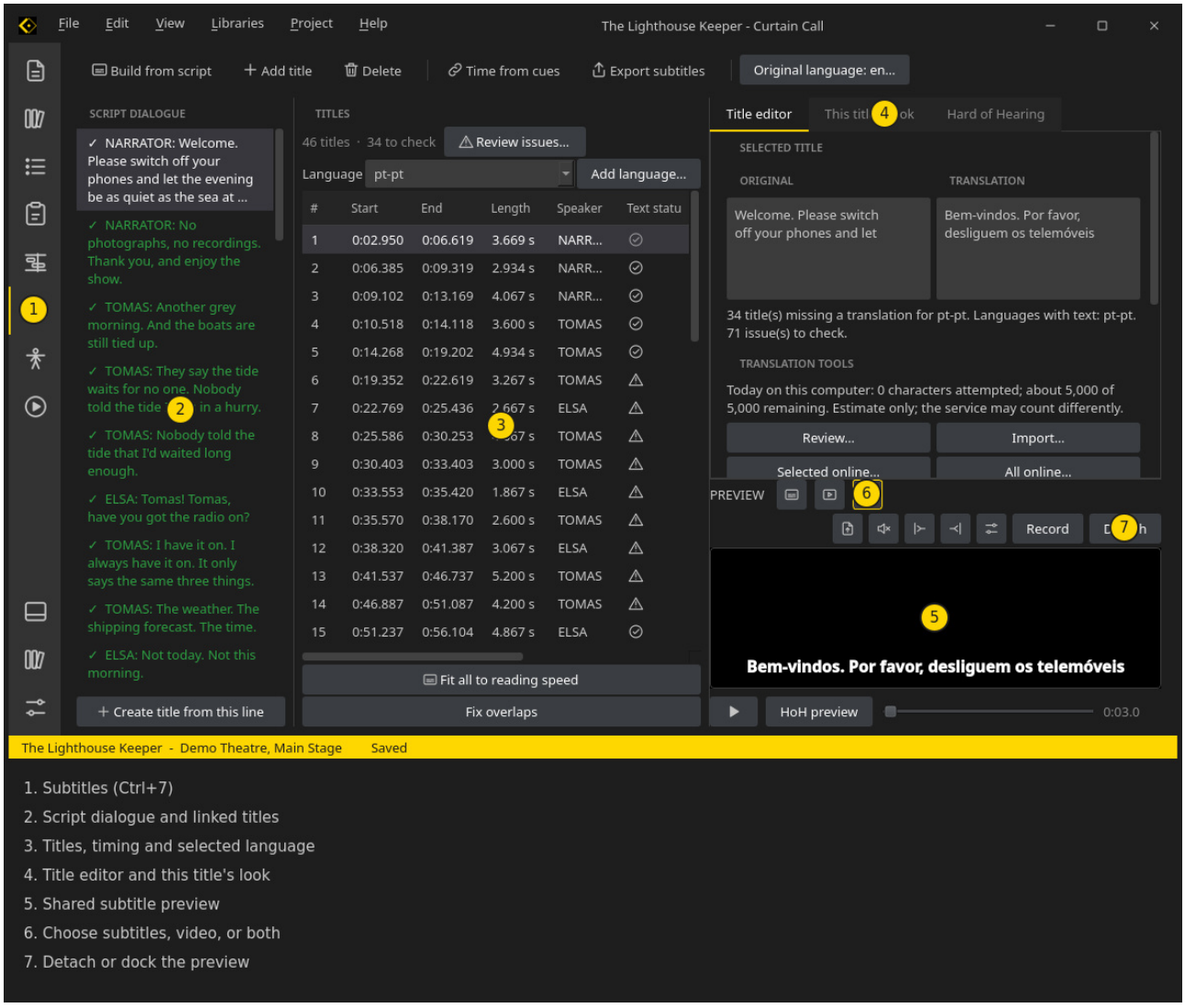
Note: A caption is not proof that the correct recording was assigned. Read its track code, audition safely and compare the actual sound with the intended description.

18.8 Subtitles: turn the dialogue into a title track

Captions describe **sounds**; subtitles repeat the **words**. The Subtitles mode (`Ctrl`+7) reads the imported script, turns each line of dialogue into one or more timed titles and writes the usual subtitle files.

Example: The Keeper says *"The light has gone out again. I told you the lamp needed oil before the storm came over the hill and the whole bay went dark."* At 42 characters per line and 15 characters per second that becomes two titles, starting one after another; the first has two lines and the second has one.

The mode has three columns and a resizable preview at the bottom right:

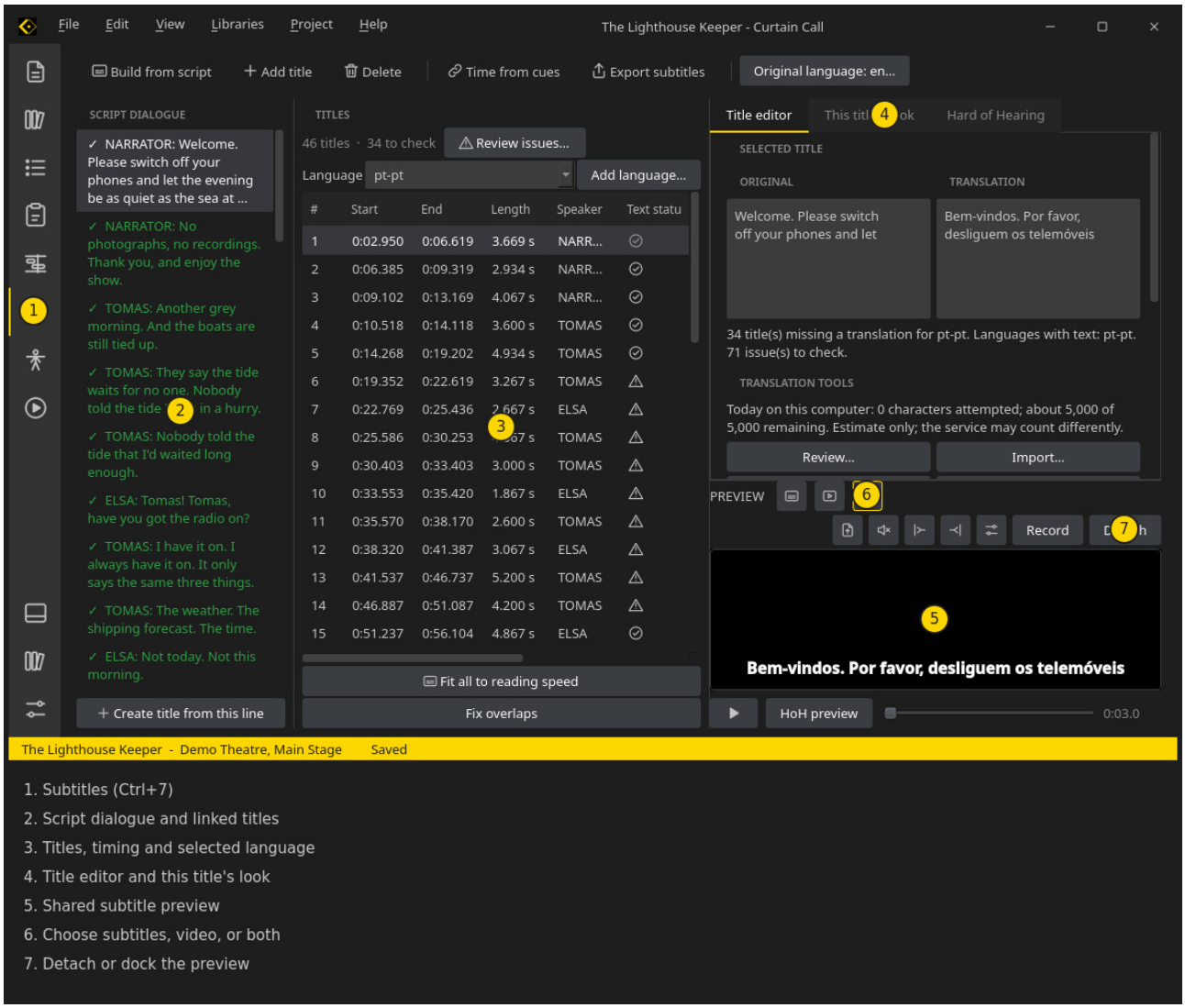


- **Script dialogue** (left) lists every spoken line with its character; a line with a title is ticked ✓. Click a line to open its title, or press **Create title from this line** to write one for it.
- **Titles** (middle) is the track in playing order with **#**, **Start**, **End**, **Length**, **Speaker**, a text-status icon and the selected-language **Title**. The original view is headed **Title (original)**. Use the **Language** selector above the list to display the original or any saved translation for every title at once; **Add language…** sits beside it. A check icon means text is present; a warning icon means it is missing, and hovering the icon explains which. Missing translations stay blank rather than falling back to the original. A title that is too quick to read is shown in red with a hover explanation showing the time on screen and the time needed to the millisecond. **Fit all to reading speed** below the list adjusts every non-empty original title in one undoable change. **Fix overlaps** moves overlapping titles later after confirmation, without shortening them.
- **Selected title** (right) holds the words and the two times. Times are typed as `1:23.5`, `01:23,250` or `83.5`; **Apply times** takes them, **Fit to reading speed** stretches or shortens the title to the time its words need. The **This title's look** tab holds its separate typeface, colour, size and time controls. Original and translated text are side by side; drag the column dividers to give them more room.
- The selected-title panel has **Title editor** and **This title's look** tabs. Each scrolls independently in a short window; the title list and unified preview stay outside the scrolling panel. The look tab changes only the selected title, and its controls follow the project look until overridden. The words editor keeps a readable minimum height instead of being squeezed. Start and End each have their own form row, and long look-field labels can wrap above their controls so a narrow pane does not clip their right edges.
- The **Preview** at the bottom right can show subtitles, **Reference video** or both in the same display. Without a reference video it shows the title that is on screen at the playhead. Selecting a title (or its linked script line) seeks the preview to its start, without pausing a running preview. Even an overlapping or zero-length selected title is shown immediately; once the preview advances, it follows the playhead again. Drag its upper divider to give it more room. **Language**

above the title list switches the whole table, the selected-title translation editor and the silent preview together. A missing translation is blank, not a fallback. Drag the slider, or press play to run the track silently: it never touches the audio engine, so it is safe before a rehearsal. The silent preview shows the selected title's resolved font, bold, colours and alignment, scaled for the available space. It is not a pixel-exact rendering of the audience screen; check the real output before a show.

18.8.1 Time titles against a reference video

Use the three buttons beside **Preview** to show subtitles only, video only, or subtitles over video. They change the visible layers without interrupting the shared preview playhead. The reference video is for preparing subtitles against a recording, not for running a show. Before importing, obtain permission to copy and use the recording. **Import video…** copies the chosen file into `reference-video/` inside the project, without changing the original. The project stores a relative path. Save and move the `.ccshow` file, or copy the entire folder for a folder project. The supplied Lighthouse Keeper example uses a timer video made by **YZONK** with Kdenlive; its soundtrack source and attribution link are listed in `THIRD_PARTY_NOTICES.md`. Save the project after importing. When you reopen it, the saved video is loaded into the combined subtitle and video preview, paused and muted; press Play to review it. If the copied file is missing, the preview reports it. Import another file to replace or relink a missing reference; old copies are not deleted by undo or replacement. Recordings can contain sensitive material: share the project only with people allowed to receive the video.



Play, pause or drag the seek slider; the displayed video and show times have millisecond resolution, but actual seeking and decoded frames depend on the media backend and keyframes and are not frame-accurate. The selected language's title is overlaid on the picture, with missing translations left blank. This preview does not reproduce the audience screen's layout or typography. The player starts

muted; use the **Unmute video audio** button to hear the recording and the same button to mute it again. Leaving Subtitles, changing projects, closing the app or disconnecting a screen stops it and mutes it again. The independent player never drives the show playhead, cues, show audio or audience titles.

The preview toolbar has **Import video…**, mute, **Mark in** and **Mark out** buttons. Select a title, seek to the desired moment and use **Mark in** for its start or **Mark out** for its end. At narrow widths the buttons show icons; hover for their names. Negative show times and an end at or before the start are rejected. These edits are undoable; review reading speed and overlaps afterwards.

On entering Subtitles mode, the first title is selected and any available reference video is shown. Select a different title to start there; the reference preview seeks to that title's saved start time, adjusted for the show offset. The picture updates after the video loads and the seek completes. The exact displayed frame depends on the media backend and keyframes.

Press **Record** to record a sequence. If no video is available, the import and permission workflow opens first. The preview detaches, shows the video and displays the next original and the first available non-empty translation side by side below it. Play the video, then use **Record next title** or **Space** in the detached preview to call that title and record its start at the current video time plus the show offset. During recording, existing title times do not automatically change the displayed title. A called title remains visible for the reading time calculated from the displayed language, project reading speed and the title's minimum display setting. If you press GO before that time ends, it waits until the title has finished before calling and recording the next one. You can queue multiple GO presses, up to the number of remaining manually called titles. The next-title panel shows the title that an additional GO will queue and how many are already queued. The duration follows video playback, so pausing also pauses the countdown. The recording panel reserves space for the next original and translation, warnings and a reading-time countdown, so its video area does not jump as titles change. Very long text is abbreviated with an ellipsis; hover over it to read the full text.

If the called original has no punctuation at its end, the next title is treated as a continuation and called automatically after the reading time. The next title's panel displays a coloured notice, while GO queues the next manual title after the continuation chain. This can continue through several titles. A final punctuation mark, including one before a closing quote or bracket, ends the chain; empty originals do not start one. Seeking or selecting another title cancels pending automatic calls and queued GOs; stopping or leaving recording also clears the queue. Each automatic or queued call records its actual start at the current video time plus the offset and is undoable.

This preview duration does not alter saved end times: each recorded title keeps its previous duration, and other titles are not shifted. Check for overlaps or gaps even when titles were called automatically. The recording sequence follows the title-list order at the start of the pass, even if new times reorder the list. Selecting a different title during the pass seeks to its saved time and makes it the next title to call. Each recorded change is undoable.

Pause playback to seek; resume playback before recording another title. **Stop recording** pauses the video and leaves recorded edits in the project. To continue later, select the next title, press Record and seek to the desired video position if necessary. Recording also stops when the preview docks, the video ends or fails, or the project changes. **Review reading-speed fit…** lists proposed end-time changes for titles recorded in the latest pass. Apply or cancel the review; applying is one undoable edit and does not move other titles. Undo or Redo clears the list for that review; a new recording pass creates a new list. Recorded times or fitted ends can overlap: use **Check subtitle track** and review the results before saving or running the show.

The settings button opens **Show offset (seconds)**, which aligns the video clock with the show clock. Leave it at zero if they start together. For example, if the recording has a ten-second introduction before the show starts, enter `-10`: video 0:15 then means show 0:05. The timing window displays both times as you adjust it. Type a precise offset or use the arrow buttons for one-second steps. Use your computer's decimal separator for fractional seconds. An incomplete number is marked and does not change the saved offset. Changing it is undoable and changes how video positions map to show time; it does not shift existing title times.

Selecting a title seeks the video to its start after applying the offset; if that time is outside the recording, a warning appears instead.

Detach opens the same preview in a separate window in either mode; the source selector remains available there. In that window choose a named screen and **Send to screen**, or close it to dock it. Moving video back into the panel is deferred until the window's close event has returned; recording changes remain in the project and are not discarded by docking. For safety, detaching or moving it is disabled whenever the audience window is open; opening that window automatically stops and docks an already detached reference. Screen removal also stops and docks it. Drag the divider above the docked preview to change its height; in a short window, scroll inside the preview to reach its controls, or detach it for a larger picture. Verify the screen arrangement before use. Common containers such as MP4, MOV, MKV, WebM and AVI are offered, but codecs and seeking support depend on the operating system's Qt media backend. A missing file or playback error is shown in the panel; re-

import a valid copy if necessary.

18.8.2 Build, time and export titles

New projects use the default original language from **App preferences > New projects**, initially the OS language. Unset originals in older projects use the project/app fallback on the first editing or translation action. To override the default, choose the original subtitle language using a searchable list of language names or codes, such as **English — en-us** or **Portuguese (Brazil) — pt-br**. You can also enter a valid custom code if it isn't listed. If the fallback conflicts with an existing translation language, the app asks for the actual original; Cancel leaves the title unchanged. The toolbar's **Original language** button can correct the code later after confirmation; this changes the label, not the words. It cannot be the same as an existing translation language. The choice is saved with the show and can be undone.

Action	What it does
Build from script	Creates a title for every line of dialogue that has none, following the titles already there. Each generated title is filled with the speaker from the script's Character line. Run it again after adding dialogue: existing titles and their timing are untouched.
Add title	An empty title after the last one, ready to type.
Delete	Removes the selected title; the script keeps its words, so it can be built again.
Time from cues	Uses the Show Score's cue firing times as starting points. Overlapping titles are moved later, including titles without a cue, while keeping their lengths; they may no longer coincide exactly with their cues.
Export subtitles	Writes `exports/subtitles.vtt` and `exports/subtitles.srt` (also **File > Export > Subtitles (WebVTT + SRT)**).

Creating a title directly from a selected dialogue line keeps its stable script line ID, and the speaker is filled automatically from the script's Character line; every split title keeps that speaker. When inserted between existing script-linked titles, the new title is placed after the nearest earlier dialogue; later titles ripple forward only as much as needed, retaining their durations, translations and relative spacing. The insertion and shifted times are one undoable edit. Editing a translation changes text on that linked title; it does not create a new title or change its place in the track. **Add title** makes an unlinked, empty title, so it has no script speaker to fill in. To connect it to the script, select the title and choose **Link to script…** in **Title editor**. Search by dialogue or speaker, select a line and choose **Link to selected line**. The picker shows how many titles already use each line; several titles can share a line,

as with a split speech. You can also change the link or choose **Remove link**. The text, speaker, times and translations are not replaced, and linking or unlinking is undoable. If a linked line was removed, the editor says it is missing so you can relink it. Later **Time from cues** uses the linked line's cue when available.

Fit all to reading speed sets each non-empty original title's end to the reading time it needs, respecting that title's minimum display time. Fit all and **Fit to reading speed** for one title automatically move overlapping later titles, using the project's subtitle gap. Increasing a title's minimum display time in its look does the same if it lengthens the title. **Fix overlaps** handles an existing track without changing lengths; it asks first. All these edits are undoable. Moving titles also moves their translations and may take them away from cue or rehearsal timings; review the silent preview and retime as needed. Manual timing edits and rehearsal marks are not silently changed as they are entered. Fit uses the original text, so review translated-language reading checks separately. Fit actions and new script-built titles round their end time up to a millisecond so the reading or minimum display time is not shortened by rounding. When a start has finer precision, fitting also accounts for its rounded export time. The times and length in the title list and selected-title fields show milliseconds; the warning rounds the current duration down and the requirement up so a small shortfall is visible. For example, 17 characters at 15 characters per second need at least **1.134 s** when timed to milliseconds, not **1.133 s**.

18.9 Time subtitles during rehearsal

Build and review the title track before rehearsal. In Show Player, press **F8** at the moment each next title should appear. The first press is available after the first GO starts the show timer. Each press records that timer position as the next non-empty title's **Start**, keeps its existing duration (at least 0.8 seconds), and advances through titles in playing order. The new times are ordinary undoable project edits and are used by later previews and exports.

F8 does not display a title on the audience screen. It is a timing tool for capturing the subtitle track during a rehearsal; a later playback workflow can use those saved times. Timing is measured from **Subtitles GO** (F9) when that has been pressed, otherwise from the first GO; it is refused before either and while paused. Timing is refused before the first GO and while paused. Use **Reset** beside the Show Player timer to start a fresh pass through the titles. If a rehearsal is stopped and restarted without resetting the timer, the next press continues from the next title not yet timed in this pass.

Example: Build titles for The Lighthouse Keeper. Start the scene with GO, then press **F8** when the Keeper's first line should appear, and again at each following line. Review the saved Start/End values in Subtitles, correct any late or early marks, then export WebVTT/SRT.

The status above the title list says how many titles and issues the original track has. Choose **Review issues...** beside it to open the scrollable list of checks; it disappears when the track has no issues. This is separate from the translation-specific **Review...** action. Checks are: empty text, more than two explicit text lines, an end before the start, less than 0.8 s on screen, too quick for the project's reading speed, and two titles overlapping. The original export asks for confirmation if titles have more than two explicit lines.

Enter a line break in either text editor to make a two-line title. Text is never discarded if an existing, imported or edited title has more than two lines: the checks warn instead. The silent preview also warns if the selected text wraps onto more than two lines at the preview's current width and font. That visual warning describes this preview only; another subtitle player or screen may wrap differently. Review the exported file on its intended display.

18.9.1 Reading rules and saved title data

How speeches are broken up is a **project setting**: **Project > Project settings > Subtitles** holds the character limit (default 42 per line), the reading speed (characters per second, default 15) and the gap between titles (default 0.15 s). The same tab holds the project's own typeface, size and time on screen, described in Caption and subtitle look: separate sections. Each generated title fits up to two lines of that length, balanced at word boundaries when possible. A word longer than the limit remains whole. Both **Build from script** and creating a title from a dialogue line use the rule; existing titles and timing are never changed. Titles can still wrap differently on a particular screen, so review the preview and exported file.

Edit > Profiles > Subtitles sets the default for future projects and lets you clear saved language-layout defaults. Existing projects and saved app defaults retain their older **Per title (legacy)** rule until you explicitly choose **Per line**. In the legacy rule, the number limits the entire title, not each line. Both dialogs let you change the number and rule; Apply app defaults and Save as app defaults copy both.

Example: With 20 characters per line, “The light has gone out again and the bay is dark.” becomes “The light has gone / out again and the” followed by “bay is dark.” The slash represents a line break.

Subtitle edits are undoable, and the track is saved inside `.ccshow` like cues and script lines.

18.9.2 Translate and export a subtitle language

Select a title, then choose a saved language from the selector or choose **Add language…** and enter a code such as `pt-BR` or `fr`. Type in the Translation editor beside the original; codes are matched without regard to case. Adding a language alone does not change the show: it remains available for this session, and is saved when text is entered or imported. The number of titles still missing text for that language is shown below the editor. Translations are stored in the project and share their original title's start and end times. Editing them is undoable; changing the timing affects every language. The translation panel also counts timing and reading-speed issues for the chosen language, including explicit titles longer than two lines. The compact **Translation tools** grid puts review/import, selected/all online and local translation, model setup and export actions side by side. Hover a short button label for its full action and privacy details. Choose **Review…** to see which titles need attention: a translation may need more time than the original. Adjusting the shared time affects every language, so review their timing again. Use a different code for each additional language; there is no fixed language count. The status below the editors names languages with saved text. Separate imports are copied into the project rather than linked to their source files, so changing an imported file later does

not change the show.

The demo project includes the full English dialogue title track and five Portuguese (Portugal, `pt-PT`) example translations. Other titles are intentionally untranslated so you can try the missing-translation checks.

Import... accepts UTF-8 SRT or WebVTT plain text. It matches titles in time order and requires the same count and the same start and end times to the millisecond. Identical project time ranges are ambiguous and cannot be imported. Markup, entities, WebVTT regions and styles are not supported; remove them before import. Existing translations in the selected language are replaced only after confirmation, in one undoable change.

Export... writes `exports/subtitles-<language>.vtt` and `exports/subtitles-<language>.srt` with the shared times. If some translations are missing or timing checks find a problem, a confirmation lists the issues before export; missing titles will be omitted. The regular **Export subtitles** action still exports the original language. **Selected local...** and **All local...** in Translation tools offer optional offline machine translation. Select the destination language first; the source is the script's original language. If the engine or model is missing, the installer asks before contacting the language catalogue. Check the requested pair or choose additional pairs, then approve the installation. Use **Models...** in Translation tools, or **Edit > Translation models...** from anywhere in the app, to open the same installer without translating. The Edit menu command is also available on the welcome screen, before opening a project; installed models belong to this computer rather than a project. After approving access to the catalogue, search by language names or codes (for example `English  Portuguese`) and check the pairs you want. Use the availability filter to show all pairs, available, installed or unavailable/incomplete models. Changing either filter keeps checked pairs, including those hidden by the filter. **Select visible** checks selectable pairs shown by the current filter, including installed pairs for uninstall; **Clear all** clears selections, including hidden ones. The selected-pairs box and count show all checked pairs and distinguish those to install, including selections hidden by the filter. Review the space and privacy information, then choose **Download and install selected**. The app verifies each completed installation and confirms the installed pairs. If no engine is installed, you can instead install just the engine, but you will still need a language pair to translate. Installed pairs are never scheduled for another download just because they are checked. Use **Show model details** for the installation location, resource checks and compatibility and licensing cautions.

Translation and AD speech model dialogs share the same default size, search/status row, checkbox

list, **Select visible / Clear all**, checked-model summary and **Download and install selected / Uninstall selected...** action row. They scroll at short window sizes; filters and local Refresh preserve checkbox selections, including hidden ones. Batch actions describe their selected targets before installing or removing files. Failed or cancelled batches retain completed installations and report that partial outcome explicitly.

Installation automatically runs the fixed sample `Test.` through each translation model, without project text. Speech installation similarly loads and synthesizes a fixed sample. There is no separate manual readiness action or expiring status: the list stays **Installed**, or reports incomplete files. The engine is checked again when used. Translation checks retain enforced Linux network isolation; other platforms do not enforce it. Failures are explicit and never certify quality.

To uninstall a directional pair, choose **Edit > Translation models...** (also available on the welcome screen), check installed pairs and choose **Uninstall selected...**, then confirm. There is no redundant Installed models button: **Download and install selected**, **Uninstall selected...** and **Cancel** share one bottom row. The AD speech manager uses the same row with **Close**. AD generation keeps **Generate preview**, **Play preview** and **Generate audio** on one row as well. Search by language name or code to narrow the list; **Refresh** rechecks the local folders without contacting the catalogue. The list marks **Incomplete** if the model weights or its shared sentence detector are missing or empty. Select a pair to see the missing files; **Files present** only checks those files and does not test the engine, file integrity or translation quality. An existing incomplete model folder must be removed before reinstalling that pair. For a missing or empty sentence detector, double-click the installed pair to open its maintenance details, select it and choose **Repair detector...**. After confirmation, the app downloads a replacement from a third-party host without sending subtitle text; other pairs that use that detector benefit too. It does not replace a nonempty detector, and a file-status check does not verify its integrity. The list and confirmation show the size of the model's regular file contents, excluding shared files and links; actual disk space recovered can differ because of filesystem allocation and shared files. The app checks the size again before removing it and asks you to review again if it changed. The installed-only dialog lists local models without an internet connection or a platform check. Opening the catalogue contacts the official service directly, without a separate confirmation; downloading models still requires approval. The selected model's files are deleted, so that direction must be downloaded again before it can translate. The reverse model, other models, the engine, download cache, shared sentence detectors, projects and saved subtitle translations are kept. These shared files may still use disk space. If model metadata is damaged, duplicate or unsafe, the app reports the problem instead of guessing which files to remove. A removal error can leave a

partially removed model; the error explains that possibility.

Already installed pairs are marked and skipped; a model without a supported sentence detector cannot be selected. The reverse direction requires its own model. Some source languages use a fallback sentence detector, which can affect sentence splitting. Neither action calls the online translation service during translation. Local translation passes subtitle text to the installed engine on the same computer; setup can contact third-party software and model hosts but does not send subtitle text. Review the model source and licence before downloading or redistributing it.

The optional engine is Argos Translate, created by Argos Open Technologies, LLC. Argos Translate 1.11.0's engine licence is MIT, which permits commercial use and redistribution with its copyright and licence notice. Its sentence-splitting dependency MiniSBD is credited to Piero Toffanin and the LibreTranslate Authors. The installed MiniSBD 0.9.5 Python package declares AGPL-3.0; consult the project licence and review its requirements if redistributing the engine or offering a modified version over a network. Other dependencies and detector models have their own terms. The installed English-to-Portuguese and Portuguese-to-English 1.9 model READMEs say their original OPUS models are CC-BY 4.0. That licence permits commercial use and sharing with attribution, a licence link and an indication of changes (Creative Commons summary). The precise terms for the packaged models and other catalogue entries have not been independently verified; do not assume the same licence applies to all of them. The catalogue examined did not publish checksums for its entries, so file-presence checks do not establish authenticity or integrity.

The optional Argos Translate engine, models and download cache are kept in `local-models/argos-translate` beside the application, not in the project. Existing `optional-translator` installations continue to be used if the new Argos folder is absent. If both exist, the new folder takes precedence. Do not move a Python environment to the new location; reinstall the engine instead. The folder may be omitted from a distribution; a copied Python environment is not necessarily relocatable or compatible with another machine. The location must be writable. For the compiled Linux download, install Python 3 and its virtual-environment support separately before choosing to install the optional translator; the app uses that Python to create the separate engine. The download does not include the translator engine or language models. For storage planning, allow approximately 3 GiB of engine downloads, 6 GiB installed and 3 GiB of cached downloads, based on a Linux installation. Example language models download around 62–83 MiB each; actual sizes and temporary space vary by model, platform and dependency version. For multiple selections the installer shows an illustrative range, not the actual download size; downloading all available pairs can take many gigabytes. The confirmation

shows free disk space, approximate available physical memory when reported, operating system and architecture, and on x86-64 whether SSE 4.1 is reported (one dependency needs it). In a container, also check the container's own resource limits. Before showing the installer and again before installation, the app creates and removes a temporary test file in existing installation locations (or their nearest existing parent). If installation reports a permission or space error, correct it before retrying. Cancellation or a failure can leave a partial engine, cached downloads or an installed model; completed language pairs remain installed and no subtitle changes are made. The installation progress window shows the current operation and keeps **Cancel** available; scroll its message if it is long. Download activity shows file name/count, actual transferred bytes, speed and estimated remaining time when measurable. Unknown totals use a moving indeterminate bar and explicit unknown size/time labels. Multi-file packages accumulate transferred bytes; their total stays unknown until all needed file sizes are known. Engine installation, unpacking, verification and automatic model loading have distinct moving progress phases. An incomplete engine or model folder may need manual repair before retrying.

The app offers local translation on Linux, Windows and macOS with x86-64 or ARM64 processors. Linux and Windows 11 have been tested; macOS testing status is listed in the work-in-progress section. Installation requires compatible Python package wheels. Linux also requires working unprivileged `unshare` network isolation. Windows and macOS run the engine as a local child process but do not block its network access at the operating-system level. Use a firewall or disconnect networking if that guarantee is required. The app never falls back to the online service. ChromeOS may work when running the Linux app in its Linux environment and unprivileged `unshare` is available. FreeBSD and other BSD systems are not supported. Only a directly available language pair is used; for regional codes, confirmation is required to use a general language model, which may not match the dialect. Review every translation, line break and shared timing. After confirmation, replacements are one undoable edit.

Choose **Selected online…** for just the selected title, or **All online…** for every nonempty title. Both send original text to MyMemory only after you confirm. The confirmation separates the text being sent, privacy warning, estimated usage and replacement details. MyMemory is a third-party service: check that you have permission to share the script, and review its terms and privacy policy. MyMemory may retain data and use external providers. Its published anonymous limit is 5,000 characters per day and 500 UTF-8 bytes per request; no login is required, but service availability and language support are not guaranteed. The usage line shows characters attempted today on this computer and an approximate remaining amount from the published 5,000-character allowance. This

is **not the provider's actual balance**: other devices, other programs, shared connections, counting rules and the provider's reset time can differ. It starts when Curtain Call begins tracking, resets at the computer's local midnight, and counts each request before sending, even if it fails or you cancel. The counter is stored in `translation-usage.json` beside app preferences, not in the project; no script text is stored in that file. If the counter cannot be read or saved, translation stops before sending.

Empty original titles are skipped by the all-titles action. Cancel waits for the current request; a failure or cancellation applies nothing. Once all requests succeed, confirm again to replace the selected language's existing translations in one undoable change. Machine translations can be wrong: review every title, line break and shared reading time. Original text, IDs and timing remain unchanged. To show translations live, enable their languages and set their screen regions in **View > Subtitle language layouts...**. Each translation uses its original title's timing, so review every language and rehearse the screen before use.

Before the first **GO**, check **Subtitles: ...** in the upper Show Player header. It lists the languages currently enabled for the audience screen, shortening long lists; while paused or recording it may read **Subtitles...** or use the captions icon to leave room for live warnings. Hover for the language summary, or use a screen reader for the complete list. Click it and confirm to open the layout editor, where you can choose one or more languages. After the first GO or Subtitles GO, language changes are locked for the current performance; resetting the timer starts a new subtitle pass but does not unlock them. In earlier wording, this lock happened after the first armed **GO**. Stop audio and any recording, then confirm **More > New performance...** to unlock language changes before the next GO.

Example: In **The Lighthouse Keeper**, select `pt-PT` to see the sample translations, including “Bem-vindos. Por favor, desliguem os telemóveis”. Add the remaining text, review reading-time warnings, then export the Portuguese files and compare their timestamps with the English track.

Original titles are typed and shown **left to right**, even when a line starts with a quote or a bracket and even on a computer whose desktop language reads right to left. Translation editing uses the text's natural direction, so right-to-left languages can be entered without forcing left-to-right paragraph formatting. The same holds for every text box in the app where you write prose (cue notes, the drop-pin note, rehearsal notes and the feedback form).

18.10 Caption and subtitle look: separate sections, three levels

Sound captions and subtitles are distinct text layers, so each has its own section in both settings dialogs: **Audience captions** and **Subtitles**. The app preferences hold this computer's starting point, the project settings hold the production's choices, and one title can change both again.

Level	Where	What it holds
App	**Edit > Profiles > Captions**	The caption board itself: font, size, bold, colours, alignment, position and how many captions show at once, plus how long a caption stays up.
App	**Edit > Profiles > Subtitles**	The same rows as the caption page for the titles' own look, and **Subtitles use the same look as the audience captions** (ticked by default). Untick it to give the titles their own typeface, text size and time on screen.
Project	**Project > Project settings > Audience captions**	The caption words for this production, e.g. a bigger size for a large house. Empty rows say **Follow the level above**.
Project	**Project > Project settings > Subtitles**	The rules (longest title, reading speed, gap) plus the same tick-box and, when unticked, this production's own subtitle look.
One title	**Subtitles mode > This title's look** (right panel)	Overrides for the selected title only, e.g. one whispered line that needs longer on screen.

18.10.1 Typography controls and the sample

Both app pages place the **working sample above the controls**, followed by outlined **Text**, **Background**, **Outline**, **Shadow** and **Timing** groups. Subtitle, caption and HoH editing previews have taller sample areas and always centre text vertically. This preview-only centring does not change saved placement or the Show Player placement-dialog preview. Related settings stay stacked in the same column; wide windows use two columns and narrow windows use one. Numeric rows keep the label, stretching slider, precise field and conditional red reset arrow together. The caption board adds **Captions at once** alongside **Minimum display time** in the Timing group. The sample is two lines of text drawn in the chosen typeface, at the chosen size and in the caption board's colours, scaled down

from the configured output size so what fits there fits in the sample too. A long caption wraps onto further lines, and the block is scaled so the whole sample always fits the board — nothing is cut off.

Caption profiles offer the same aesthetic controls as subtitle profiles: line and letter spacing, horizontal/vertical background padding, whole-block, per-line, per-word or per-character backgrounds, corner rounding, outline width/colour, and shadow offset/colour/opacity. The same controls are available as inherited overrides in the project's caption editor. Colour alpha is preserved. As with subtitles, the caption background is painted **behind the text**, not across the entire screen; the surrounding screen remains black. This also applies to existing caption profiles with coloured backgrounds. The editing sample uses the HoH pattern so transparent backgrounds and box geometry are easy to judge; the pattern is never shown to the audience. Old profiles without these settings load with normal spacing, no padding/insets, no outline/shadow, and a whole-block background.

Audience text sizes use percentages of output height, not points. New/reset base sizes start at **5%**. The synchronized horizontal slider moves by **0.1%**, while the editable field accepts **two decimal places** within **0.1–20%**. Defaults are available in tooltips rather than captions below the controls; red reset arrows restore app defaults and overrides restore inheritance. The tooltip gives the nominal pixel size: **5% of 1080 px = 54 px**. This is a font size, not the measured height of every glyph.

Project/title/language controls use the show's configured **Second screen size**, including the layout dialog opened from Show Player. Profile previews without a show use the app's new-project output size. Previews scale the full output, not the height of a subtitle language band or the short mirror strip. Operator interface fonts are unchanged. With percentage sizing, widening the preview also scales its font; shorten the text or reduce its size to address a wrapping warning.

Feeling profiles and explicit HoH title-size overrides also use **absolute percentages**: a feeling set to 6% remains 6% when the base subtitle size changes. Factory feelings retain their expressive differences around the 5% baseline; for example, a previous 1.5-times factory proportion becomes 7.5%.

Older saved point sizes and feeling/title multipliers remain compatible. Their percentage display accounts for output height and screen DPI; opening, editing another field or saving does not round or replace the original size. Editing the size stores an absolute `font_size_percent` (or `force_font_size_percent` for a HoH title). Out-of-range saved sizes remain visible with **Legacy value retained**; new entries must stay in the agreed range. Older multipliers continue following the

old base until you edit or replace them. Existing projects and profiles are not reset automatically.

18.10.2 Following and overriding the caption look

While the subtitles follow the captions their rows stay readable and usable and simply show the caption look, so the page tells you what the titles really use. Change any row by hand and the tick comes off by itself: the titles then have a look of their own, starting from the caption look you saw.

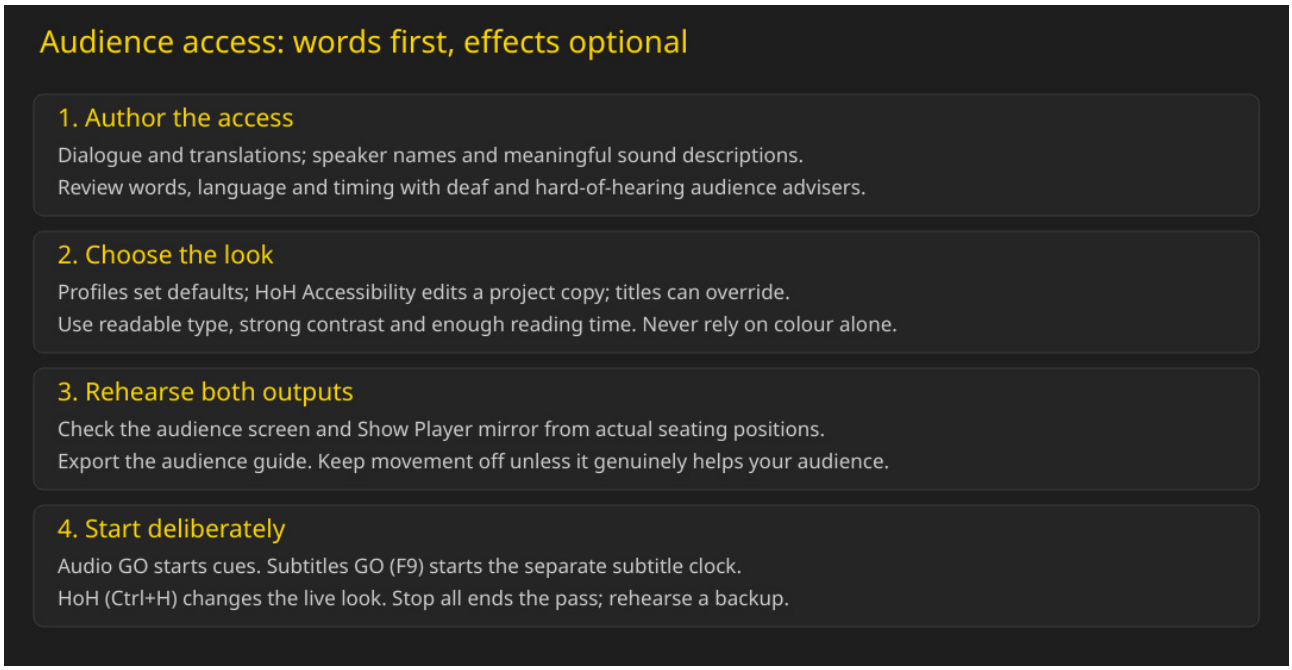
The Subtitles mode shows what really applies below the look controls, e.g. *"system font, 5.00%, at least 5 s · the captions' look, from the project"*, and the preview strip shows the resolved appearance at a scaled size. The **Time on screen** value is a floor, not a fixed length: a title that needs longer to be read still gets the time its words need, and a title shorter than the floor is listed as a check.

Example: the app's caption board is Noto Sans at 5%, and the subtitles follow it, so titles are Noto Sans 5% too. The Lighthouse Keeper wants its titles slightly smaller and leaves the captions alone: in **Project settings > Subtitles** the box is unticked, the project's size is set to 4.5%, and the caption board keeps 5%. Clearing the row (choosing **Follow the level above**, or typing 0) inherits again, and ticking the box makes the titles follow the captions once more.

All of this is saved as plain JSON in `preferences.json` and `project.ccshow`, so it can be read and edited by hand.

18.11 Hard of Hearing subtitles: enhance accessibility for deaf and hard-of-hearing audiences

Access is part of the production, not just a visual effect. Review subtitle wording, translations, speaker identification and meaningful sound descriptions with deaf and hard-of-hearing audience advisers wherever possible. Keep a readable still-text option, check contrast and viewing distance from real seats, and allow enough reading time in every displayed language. Do not communicate essential meaning through colour or motion alone. HoH effects are expressive choices, not a substitute for accurate text or proof of accessibility compliance.



Subtitles describe the words; Hard of Hearing (HoH) enhancements add context from tone of voice, sound effects and delivery. Enable live HoH features with **Ctrl+H** or the HoH button in Show Player, or accept **Use Hard of Hearing styling** in the subtitle language layouts dialog. The button immediately controls both the Show Player preview and audience screen, using the project's HoH settings and any profile already applied to the project. Turn it off to restore normal subtitle styling without changing the project. The separate **HoH preview** in Subtitles mode is an editing preview and does not change

the live mode. Live HoH adds configured speaker identification to each available language, descriptions to source-language titles and feeling-based typography.

The Lighthouse Keeper includes examples to explore in Subtitles mode: “Tomas, have you got the radio on?” detects Question from punctuation, and the line containing “Forty people” keeps the factory Normal feeling without a custom pattern. “But soon…” detects Trailing thought and “TONIGHT WE COUNT THEM FAST!” detects Shouting. “And I'm an old man. I'm tired.” and “Then we oil the memory.” use individual feeling and size overrides; the former also overrides the speaker colour. In the summer memory and storm scenes, “CATCH IT!” and “CAN ANYONE HEAR US?” demonstrate Shouting with word-by-word shaking. The following “Anything at all??” detects Emphatic question with floating words. Select a title and open its **Hard of Hearing** tab to inspect the overrides, or turn on **HoH preview**. For live output, enable HoH in Show Player and start the subtitle clock; the Show Player preview and audience screen use the same resolved settings. The demo project explicitly enables live movement; other projects retain their own settings. Applying another profile replaces those settings, including the live movement choice. These are illustrative demo timings, not synchronized performance cues.

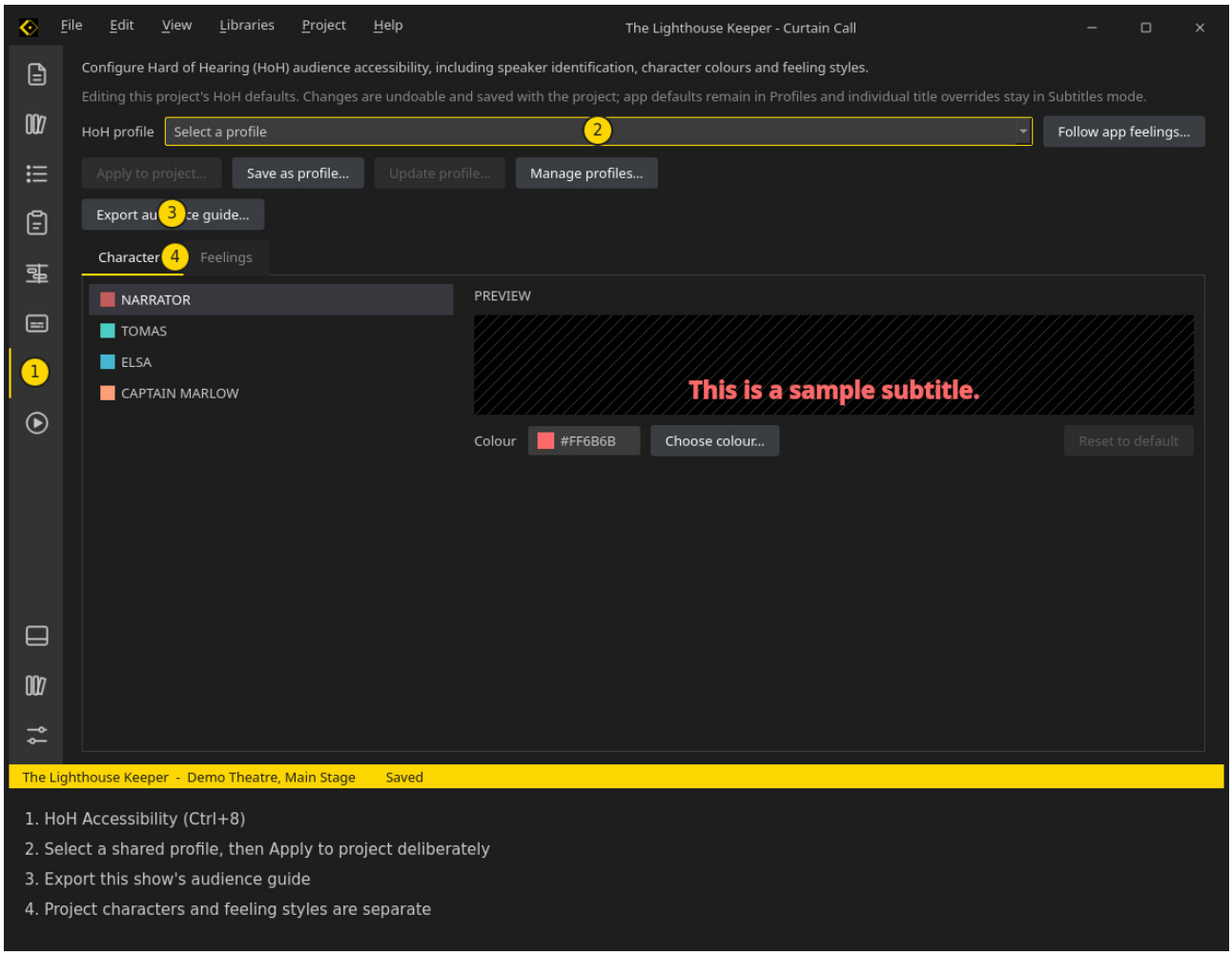
If no HoH profile has been applied to the project, the first time you enable HoH after opening it, Curtain Call offers to open HoH Accessibility. Choose a profile there and use **Apply to project…** if you want to apply it. Declining keeps HoH enabled with the current project settings and inherited defaults. The offer can also appear during a performance, so configure it beforehand if you do not want the question while operating.

A three-level settings hierarchy lets you set HoH defaults once and fine-tune per show or per subtitle:

Where	What it controls
App defaults	**Edit > Profiles > HoH feelings** — your personal baseline for accessibility and feeling styles.
Per show	**Project > Project settings > Audience accessibility > Hard of Hearing subtitles** — overrides for this production (for example, a different venue or cast voice profile).
Per subtitle	**Subtitles mode > Hard of Hearing tab** — fine-tune individual titles.

Use **HoH Accessibility** in the vertical activity bar (**Ctrl+8**) to configure the open project's character colours and choose a feeling profile. Edit feelings in **Profiles**, not in HoH mode. The character preview uses the project's subtitle typeface and background with the selected character's resolved colour. In Subtitles mode, turn on **HoH preview** beside the silent preview controls to see the selected title with

its resolved appearance. This is a visual preview only and does not change the title or its settings.



Choose **Export audience guide...** in HoH Accessibility or **Project > Export > Audience guide (PDF or Markdown)...** to save a PDF or Markdown legend for audiences to read before the show. It lists character colours and feelings used by the current subtitles, including their colour and live movement. Per-title colour and feeling variations are included. Markdown swatches require a viewer that supports inline HTML and embedded data-URL images; use PDF if your viewer blocks them.

Individual title overrides can differ from the chart, and applying a different profile changes the settings: export a fresh guide after changes. Selecting a profile alone does not apply it. The guide describes the HoH look when HoH is enabled, so check the live display and the printed colours before distributing it.

The **Feeling profile** selector uses the shared HoH feelings library. Selecting a profile does not change the project: choose **Apply to project…** and confirm to replace its feeling styles, patterns and profile accessibility options with a copy. This is one undoable project edit. The copy uses factory defaults for any values the profile does not specify, rather than inheriting later app feeling changes. Existing character colours and per-title overrides are not changed; a title that refers to a feeling absent from the applied profile retains that reference.

Edit profile opens the selected profile in the embedded **Profiles** manager. Edit its feeling styles, patterns and general options, then choose **Save profiles**. Return to HoH and choose **Apply to project…** to copy the edited profile into the show; saving the profile alone does not change the show. **Curtain Call** is the single built-in entry. Existing project customizations remain untouched until you explicitly apply a profile.

The character list stays on the left, with the profile selector, **Apply to project…**, **Edit profile** and **Export audience guide…** grouped beneath it. The preview and colour editor stay on the right. Choose a profile from the selector and explicitly apply it. The outlined **Feeling profile** group contains the introduction and project status above its full-width selector, equal-width Apply/Edit actions and full-width export action. There is no separate Follow app feelings action. Existing inheritance in older or untouched projects remains unchanged until a profile is applied.

For a character, the colour field shows the current colour. Select **Choose colour…** to open the colour picker, or type a hex colour. Select **Reset to default** to remove the project choice and use the app default or automatic character colour again.

In **Profiles > HoH feelings > Feelings**, the preview shows the selected feeling's effects, typeface and size over the app subtitle look. Independently enable feeling-coloured **Font colour**, **Shadow**, **Glow**, **Outline** and **Text background**; they may be combined. The background uses a contrasting tint of the feeling colour so it remains distinguishable when feeling font colour is also on. Factory defaults enable outline for all feelings, shadow for Shouting and Trailing thought, and glow for Shouting and Very strong exclamation. Existing explicit choices remain in effect. Font colour replaces the character colour; the other effects leave it in place. The sample animates automatically while visible; no movement-preview button is needed. Hidden previews pause their timer and resume when shown. A

strength of zero or movement of None leaves the sample still. This editing animation does not enable live audience movement. Select **Choose colour...** to open the picker. Use the red reset arrow beside a setting to remove that profile override, or **Reset style to defaults** to reset the selected feeling to factory style values. A title can override the feeling colour and each effect in its **Hard of Hearing** tab.

The feeling editor also offers subtitle aesthetic overrides in outlined **Text**, **Background**, **Outline** and **Shadow** groups: bold, alignment, line/letter spacing, background colour/alpha, padding, block/line/word/character shape, rounding, outline width/colour and shadow offset/colour/opacity. Previews stay above these responsive inline controls. **Font colour** offers **Follow the level above**, **Use feeling colour** and **Keep character colour**. Following inherits the app/project feeling setting; keeping character colour explicitly disables feeling font colour, even if the inherited feeling enables it. It preserves character identification colours (or the subtitle colour where no character colour exists). The same choices are available for an individual title. Other feeling effects remain independent. Typeface and movement fields share the same left-aligned input column; sliders and reset columns remain justified. Detection patterns follow the style groups, with compact red arrows for pattern and ALL CAPS inheritance instead of wide reset buttons. The Feeling effects group sits below Movement. Background is first in the other column, followed by Outline, Glow and Shadow. Glow has its own width slider and exact percentage field (0–50% of font height), plus a colour/alpha chooser. Width 0 disables glow; unset width preserves the original two-layer glow. Unset colour inherits the level above or uses the feeling colour. Red resets restore inheritance, and these settings travel with HoH profiles and projects. Feeling-effect checkboxes use two columns beneath the full-width Font colour choice, falling back to one column when narrow. Detection text occupies the left half, with matching guidance and ALL CAPS controls on the right; compact windows stack these areas. All profile panels use thin outlined groups, including Metadata profile values/required fields and HoH General detection, movement, speaker identification and descriptions. Unset aesthetic values inherit the subtitle's resolved settings, including project, title and language overrides. A feeling's explicit values are then applied on top, only while HoH styling is enabled. Project feeling choices override app feeling choices one property at a time; applied project profiles remain independent of later app changes. Red resets remove individual overrides, even when an explicit value currently matches the inherited value. **Reset style to defaults** clears all aesthetic and existing feeling overrides for the selected feeling.

The existing **Feeling effects** toggles remain independent. When an effect is on, explicit aesthetic parameters replace its legacy width, offset, colour or opacity; an explicit outline width or shadow offset of zero suppresses that effect. Without explicit parameters, the factory feeling effects keep their existing appearance. Background geometry applies to both neutral and feeling-coloured text

backgrounds. These settings persist in HoH profiles and projects and affect both the editing sample and audience output; older profiles inherit subtitle aesthetics unchanged.

The HoH sample always keeps a neutral patterned editing backdrop. Subtitle and feeling background colours paint only behind the text block or individual lines, not across the preview. Use HoH **Background**, **Background opacity**, padding, shape and rounding to override those inherited subtitle properties; resetting restores inheritance. Switching feelings also restores any properties the next feeling does not override.

Enabling **Font colour**, or changing a feeling colour while it is used for the font, displays an accessibility warning: this may replace character colours and make speaker identification harder. Prefer feeling-coloured **Text background** or **Outline** when character colour carries important information. The warning is advisory; intentional font-colour choices remain available. Loading saved profiles does not display the warning.

Live movement is opt-in: in **Project settings > Audience accessibility**, enable both expressive animations and **Animate live subtitles and Show Player preview**. Both displays use the same pause-aware subtitle clock. Existing projects and older profiles start with live movement off; Preferences can set the initial choice for new projects. Check motion, contrast and readability on the actual audience screen before a performance.

Feeling editing is now in Profiles; the old HoH Feelings tab is no longer part of the source-version workflow. Published downloads may still use that older layout.

Movement strength defaults to **0.2** when no app or project override is set. Use the slider or its number box to adjust it from 0 to 2; existing explicit choices are preserved.

For each feeling, set **Move** to **Whole title**, **Each word**, or **Each character**. Individual words or characters move with different stable phases; combined characters and emoji stay together. Factory defaults vary by feeling: Question uses Whole title; emphatic questions, exclamations and shouting use Each word; confused questions, very strong exclamations and trailing thoughts use Each character. Normal, empty and custom feelings default to Whole title. App and project overrides remain in place.

The movement choices include Pulse, Bounce, Shake, Wobble, Fade, Sway, Float, Orbit and Tremble. Factory defaults now give Question a gentle Sway, Emphatic question a Float and Strong exclamation a Bounce. Settings without an explicit movement inherit these updated defaults; existing app or project overrides are kept.

In **Profiles > HoH feelings**, choose a saved profile and a feeling to edit it using the movement preview, individual reset arrows and **Reset style to defaults**. **New feeling** adds a custom feeling to the staged profile; custom feelings can be renamed or deleted, but factory feelings cannot. Without a named profile selected, these controls edit staged app defaults. App feelings are available in projects that inherit app feelings; applying a profile to a project replaces that inheritance with a copy. Renaming or deleting a profile feeling does not change project settings or titles that used its old name. Changes remain staged until **Save profiles**.

Enter detection text or symbols for each feeling, one pattern per line. Text matches whole words or phrases anywhere in a title, without regard to case; symbols match at the end. Normally the longest matching pattern wins; equally long matches use the order of the feelings list. **ALL CAPS triggers this feeling** chooses the single feeling used for all-uppercase titles with at least four cased letters; choosing another feeling replaces the previous ALL CAPS assignment at that settings level. Short acronyms, numbers and mixed-case titles do not trigger it. Matching text patterns take priority, then the selected ALL CAPS feeling takes priority over punctuation patterns. Unchecking disables the ALL CAPS trigger; its separate reset button restores inheritance (factory: Shouting). The matched text remains visible. An empty list disables inherited patterns for that feeling. **Reset patterns** restores factory patterns in Profiles without changing ALL CAPS. Existing project patterns are preserved until a profile is explicitly applied or you choose Follow app feelings. Old projects and profiles retain Shouting as their default ALL CAPS trigger. A feeling selected for an individual title takes precedence over detection.

18.11.1 Show speaker names and character colour coding

The **Show speaker name** setting in Project settings shows or hides the name before both the original and each available translation, formatted as `[speaker name]`. Choose **Always** or a number under **First titles per speaker** (for example 3). The count is per displayed speaker name, in show-time order, not playback order: seeking and replaying do not change it. Titles with a suppressed name or no text in any language do not count. A per-title name override counts under the displayed name. Preferences sets the initial choice for new projects; existing projects keep their own choice. When HoH mode is on, the whole title uses the title override, project character colour, app character colour or automatic palette colour, in that order, even when the name is hidden. Feeling font colour, if selected, takes precedence.

Example: Enable **Speaker identification** in a mystery play to show `[The Detective]` or `[The Suspect]` before their authored dialogue on the audience screen.

18.11.2 Describe feelings with expressive typography

When automatic feeling detection is enabled, punctuation and capitalization suggest a feeling: a question mark suggests Question, an exclamation mark suggests Exclamation, and trailing dots suggest Trailing thought. Each feeling has its own colour, typeface and absolute output-height percentage, configurable in **Profiles > HoH**. Apply the chosen profile to the show through HoH mode. You can instead select a feeling for an individual title in its **Hard of Hearing** tab.

18.11.3 Add sound descriptions and context

The **Show sound descriptions** toggle displays short descriptions of non-dialogue sounds—for example **[door slams]**, **[phone rings]**, **[waves crashing]**, or **[dramatic music]**—written in the dialogue notes at authoring time. These help HoH audiences understand the soundscape and environmental context.

Show speaker identification and **Show sound descriptions** are independent toggles; use them together or separately depending on your show and audience needs.

18.11.4 Customize one subtitle's Hard of Hearing look

Select a title in Subtitles mode and open its **Hard of Hearing** tab:

- **Feeling** — Choose a feeling for this title, or select **Auto-detect from text** to use automatic detection when enabled in the project.
- **Text size** — Set an absolute output-height percentage or use **Auto (from feeling)**.
- **Character name** and **Character colour** — Use this title's own name or colour instead of the inherited character settings.
- **Hide [Speaker:] for this title** — Suppress the speaker prefix for this title.
- **Description** — Enter a sound or visual description for the title.

These settings are saved with the title. For example, choose **shouting** for a question that is delivered as a shout, rather than relying on its question mark.

18.11.5 Export subtitles with Hard of Hearing features

The current `.vtt` and `.srt` exports (**Subtitles mode > Export to VTT/SRT files**) contain authored text and timing, not Curtain Call's percentage sizes, colours, movement or automatically displayed HoH prefixes/descriptions. If a description or speaker name must appear in those files, include it in the authored text. Audience styling stays in the project and reusable profiles; this sizing change does not alter the existing plain-text subtitle export format.

Exported subtitle files are written to `subtitles.vtt` and `subtitles.srt` in the sibling exports folder.

18.11.6 Hard of Hearing configuration fields

The following settings control Hard of Hearing features; they are stored in the project file (ProjectSettings.hoh_subtitle_settings) and your app preferences (AppPreferences.hoh_subtitle_settings):

- `auto_detect_emotion` — Detect feelings from ! and ? marks.
- `character_colors` — Character name to colour mapping (hex, e.g. `#FF0000`).
- `use_default_character_colors` — Use colours from character profiles.
- `punctuation_emotion_map` — Scaling factors for punctuation marks.
- `enable_expressive_animations` — Enable motion-based typography.
- `animation_intensity` — Animation strength (0.5 = subtle, 1.0 = normal, 1.5 = pronounced).
- `show_sound_descriptions` — Display `[door slams]` and `[phone rings]`.
- `show_music_descriptions` — Display musical background cues.
- `show_visual_cues` — Display `[audience laughs]`, `[lights fade]`.
- `show_speaker_identification` — Prepend `[Character:]` to lines.
- `always_show_full_character_name` — Use full names, not abbreviations.
- `line_spacing_multiplier` — Vertical spacing (1.0 = normal, 1.5 = wide).
- hoh_enabled_last_session — Remember HoH state across sessions.

Per-subtitle overrides (Subtitle.hoh_overrides) let you customize individual lines:

- `character_name_override` — Use a different speaker name.
- `force_color` — Override the character's colour.

- `force_font_size_percent` — Absolute percentage of output height; empty follows the feeling.
- `force_font_size_multiplier` — Legacy size scaling retained until the size is edited.
- `force_emotion_tag` — The title's selected feeling, or empty for automatic selection.
- `suppress_speaker_identification` — Hide `[Speaker:]` prefix.
- `custom_description` — Editor-provided sound description.

19 Venue profiles: take the show on tour

Example: Create **Lighthouse tour — Civic Theatre**, trim **−6 dB**, device note **House USB interface**, and checks **Verify house output** and **Listen from back row**. Stop all, preview the switch and approve it. The multiplier is about **50.1%**; demo master **90%** and cue 1 **70%** remain unchanged.

19.1 Libraries: create and edit physical venues

Choose **Libraries > Venues...** or the **Reference libraries** activity-bar icon, even on the Welcome page, to open the embedded workspace. Its icon-labelled tabs switch between venues, video projectors, lighting fixtures and venue projector installations; use the tab arrows if the window is too narrow to show them all. A venue is a place or specific room; it is **not an audio gain profile**. Personal records are plain UTF-8 JSON in `venues-user.json` beside `preferences.json` (inside `portable-data/` in portable mode). They are shared across productions on this computer, not linked live into shows. No data is uploaded or downloaded.

Factory entries have a **lock icon** and are read-only; **[Personal]** marks your own entries. **Create copy** makes an editable personal entry with a fresh ID and clears checked status. **New** creates a draft; **Delete** confirms removal of a personal record. **Save library** validates and saves the personal layer atomically. **Discard changes** reloads it after confirmation. Invalid drafts remain visible; **Back** returns to the previous mode or Welcome, and switching libraries, modes or projects preserves drafts. On quitting, save, discard or cancel for each unsaved library. A file changed by another process cannot be overwritten from an older editor.

Leave unknown dimensions, capacity or coordinates **blank**, not zero. Width, depth and usable height are in **metres**, capacity is a positive whole number, and latitude/longitude are degrees (both present or both unknown). Email must be complete; websites and source references need full HTTP(S) URLs. Technical details depend on the room and seating configuration: ask for the current technical rider, not just a building's overall capacity.

Venue, Contact, Measurements, and Notes and sources are separate tabs; each form scrolls in a short window and the list and save controls remain accessible. Entries are notes for planning: selecting, creating or changing one never starts playback, routes audio, activates a venue profile or changes live levels.

19.1.1 Bundled venues and your own source checks

Source and community contribution. The Portugal venue catalogue draws on OpenStreetMap (OSM), a collaborative mapping project maintained by a worldwide community, including volunteer contributors. The supplied dataset identifies its source snapshot as `2026-07-14-osm-ica-v3`, with an observation timestamp of **14 July 2026, 21:00 UTC**. The CSV export is dated 8 November 2026; this does not imply that every venue was revisited on that date. No institutional endorsement is claimed.

The cleaned catalogue contains **378 active venues**: **373 named OSM-derived entries** and **five**

original starter records with selected facts checked against official pages on **8 November 2026**: Coliseu dos Recreios, Teatro Nacional São João, Teatro Carlos Alberto, Teatro Maria Matos and Teatro Tivoli BBVA. These records are planning references, not certified technical riders.

Data preparation and online cross-checking. The **12 November 2026** review combined automated processing with explicit curation decisions:

- Automated checks validated record structure, source identities and source links, extracted available street addresses and postal codes, and normalized telephone and website formatting. Formatting checks do not prove that a contact is current or a website is reachable.
- Automated retrieval from the public OSM API obtained current classification tags for all **378 active entries**, preserving each object's version and update timestamp separately from the July snapshot. This checks what OSM currently records; it is not independent confirmation of physical conditions.
- Selected facts were cross-checked against online venue/operator pages and linked reference sources. Examples include Estarreja's name and contact details, Bolhão's address and box-office contacts, and the Mascarenhas Gregório spelling. Only the specific supported fields receive checked status; this was not an exhaustive official-source verification of every venue.
- Five reviewed duplicate pairs were consolidated with their source links retained. **171 unnamed rows** and **three out-of-scope records** were omitted from the active list. Nearby buildings, rooms and organisations were not merged solely because they share a name or are close together.

Traceability. Each imported entry's **Notes and sources** identifies its OSM object, original name, snapshot and observation date, supplied administrative labels, and any corrections or classification limitations. The bundled `curtaincall/assets/venues.json` contains the active catalogue and its source metadata together. Its `source_records` section preserves all **557 original rows**, including exclusions, merge decisions and the source-file checksum. The original CSV is unchanged. Personal catalogues and saved project copies are not overwritten by this catalogue cleanup.

Attribution and licence. Venue data © OpenStreetMap contributors, available under the Open Database License; see OpenStreetMap copyright.

Interpreting the space type. Venue type distinguishes physical spaces such as theatres, cinema-theatres, auditoriums/concert halls, amphitheatres, stages and music clubs. An organisation or wider cultural site is not treated as a specific room. Where evidence is insufficient or conflicting, the entry

states that its type is provisional or its physical space is unconfirmed. Name-based interpretation is identified in Notes rather than presented as independently verified. The supporting OSM tags and review metadata are retained in the `typology_sources` section of the same `venues.json` file.

Operational limits. An amphitheatre tag alone does not establish that a venue is outdoors. The catalogue does not establish seating configuration, stage form, roof coverage, accessibility provision, available equipment or fitness for a production unless those facts have been specifically checked. Coordinates locate map objects, not confirmed public, accessible or load-in entrances. A city field may contain the supplied municipality rather than a precise locality. Confirm the intended room, entrance, current contacts and technical rider directly with the venue before planning or travelling.

Each of the five checked records lists its **Source URLs**, check date and the exact fields checked. This is **field-level verification**, not a certified technical rider. For example, São João's official contacts identify **Praça da Batalha** and the general telephone; its stage measurements and capacity stay unknown. Tivoli's listed contact is the **box office**, not the technical department. The five starter coordinates were checked against their **Google Maps place markers** on 8 November 2026; each record includes its Maps source link. These are latitude/longitude in decimal degrees for locating the venue, not surveyed positions or confirmed public, accessible or load-in entrances. For example, Coliseu dos Recreios is at **38.7167438, -9.1401777**. Confirm the correct entrance with the venue before travelling. Stage dimensions and capacities remain unknown; obtain the current measurements from the venue for production planning.

Add one official source URL per line. A URL alone does not verify a fact. **Mark entered facts checked…** asks you to confirm that you checked every entered physical fact against those sources, then records today's date. It also appends a dated, operator-attested note listing the checked fields and sources. Existing notes are preserved; Cancel still discards the draft. Editing a checked field removes that field's checked status; changing the source list clears all checks. Unchanged facts keep their provenance. Local and project drafts use the same rule, and operator-attested checks are not independent certification.

19.1.2 Copy a venue into a project

Open **Project > Project settings > Venue**, press **Choose from venue library…**, search by name/city/country, review the record and press **Copy venue details**. Replacing an existing venue draft asks for confirmation. The chosen record fills the project form, and the name also appears in **Details > Venue**. Change production-specific contact or stage notes here as needed. **OK** applies one normal, undoable project edit; save the project to persist it. **Cancel** does not change the show or local library.

The show saves an **independent venue copy** in `ProjectSettings.venue_details`, alongside the existing venue name. It travels with `project.ccshow`, including its ID and source checks. Editing/deleting a library venue does not rewrite existing projects; copying again is an explicit replacement, not automatic synchronisation. Old projects without this field keep loading normally. A blank venue is allowed; partly filled invalid details must be corrected.

Example: Copy **Teatro Carlos Alberto** into a backed-up Lighthouse show. The project retains the officially listed **Rua das Oliveiras, 43** address. Add the house technician's confirmed stage clearance and load-in notes to the project copy. The shared catalogue stays unchanged, and **Project > Venue profiles...** still needs a separately tested and explicitly approved gain profile for the room.

19.1.3 Venue JSON structure

The local file uses `format_version: 1` and a `venues` list. Each record has a stable `id`; `name`, `address`, `city`, `country`, `type` and `notes`; nullable `latitude`, `longitude` and `capacity`; a `contact` object (`name`, phone, `email`, `website`); and `stage_dimensions` (`width_m`, `depth_m`, `height_m`). Unknown numeric facts are JSON `null`. UTC ISO `created_at`/`updated_at` timestamps and `checked_on`, `source_urls`, `verified_fields` describe identity, editing and provenance. Checked fields use names such as `contact.phone`. **Last update (UTC)** displays the automatic `updated_at` timestamp in the editor and venue picker. Saving an actual edit updates this timestamp; opening or saving an unchanged record does not. It is not a verification date.

Maintainers can run `python tools/check_venue_sources.py` (requires `curl`) to retrieve the bundled catalogue's website and source URLs in batches of 20. The tool records timestamps, redirects, HTTP statuses and failures in the top-level `online_source_review` section and each venue's notes, and updates `updated_at` when those notes change. It never changes `checked_on` or `verified_fields`: a working URL is not proof of a correct venue name, address, phone, email or location. Missing websites and blocked or failed retrievals remain explicit research tasks. The OSM cleanup tool preserves these results. Individually reviewed page contents are recorded separately in `online_source_review.fact_checks`, with their date, specific fields, sources and limitations, and rendered as **Content review** notes. Rechecking only an address or telephone does not refresh an older coordinate verification date. This maintenance command does not touch local user libraries or project copies.

The original list-only JSON shape can be read: missing IDs are derived consistently, and records without provenance are unverified. Malformed records, unknown record fields, duplicate IDs or

unsupported format versions produce an explicit error; they are not silently dropped or replaced by the starter catalogue. The bundled records remain separate from `venues-user.json`; a missing personal file is created when you press **Save library**.

19.1.4 Migrating older local libraries

Older `venues.json`, `datashows.json` and `projectors.json` files remain untouched beside preferences. When the corresponding `-user.json` does not exist, the app reads the old file: entries identical to a current factory entry are not duplicated, changed factory entries become personal copies with fresh IDs, and other personal entries keep their IDs. An older factory entry that differs from today's bundled version is conservatively retained as a personal copy. Nothing is written until **Save library**; afterward the `-user.json` file is authoritative for personal entries and the old file is left as a backup. Malformed files or conflicting edits are reported, not reset.

Source URLs are references only.

19.2 Libraries: video projectors and lighting fixtures

Libraries > Video projectors… opens the video catalogue. **Libraries > Lighting fixtures…** opens the lighting catalogue: despite its legacy filename, this is a **stage-lighting** catalogue, not a second video-projector list. Both commands work from Welcome without opening a show.

Use **New**, enter the **Manufacturer** and **Exact model**, then fill only the specifications you know. **Create copy** creates a fresh ID, a distinct copy name and unverified facts; rename it to the exact model and verify its own documentation. **Delete** asks for approval, including for an incomplete new entry. All changes are staged: **Save library** validates and saves the personal library, while **Discard changes** reloads it. An invalid draft stays selected until corrected or deleted; it is not silently lost when you choose another entry. The form separates identity, named specification sections and sources: video projectors group image, lens and throw, inputs, physical and network protocol details; lighting fixtures group optics, movement, DMX, physical and maintenance details. The list and save controls stay accessible while each section scrolls.

Video projector entries describe a **model**, not its setup in a particular venue. The editor therefore does not offer installed orientation, array ID, IP address or calibration files. Manufacturer-published throw distance limits and lens shift are model capabilities, not the actual placement in a room; record their lens and source conditions. Older video records retain installation-specific fields in the personal JSON file without showing editable controls; the **Notes and sources** section identifies any retained fields. Ordinary edits preserve them, but **Create copy** omits them. Marking entered facts checked also clears any old checked status for these hidden fields after a warning. Installation details can be edited separately as described below.

19.2.1 Venue projector installations

Choose the **Venue projector installations** tab in **Reference libraries**. Select **New**, name the physical unit, and choose a saved venue and video projector model, including factory entries. Enter the installation orientation, array ID, IP address, geometric-correction filename, lens-data filename and notes as needed. Create another installation for each physical unit, even if several use the same model. **Save installations** writes a separate local file; **Delete** removes only the installation. Changing a venue or model never changes the other records or the existing project venue copy.

Use **Placement relative to stage** to describe the projector's position and the stage reference point used for the measurement, for example “balcony, left of centre; measured to stage front”. Enter **Distance to stage reference (m)** as a non-negative distance in metres, or leave it blank if unknown;

zero is a known zero, not unknown. This is a recorded venue measurement, not the projector's manufacturer-listed throw-distance range or necessarily the distance to a screen. Verify the reference point and measurement on site; the app does not calculate geometry or certify an installation. Enter **Height relative to stage floor (m)** as the vertical height from the stage floor to the projector: positive above, negative below, zero level with the stage floor, or blank if unknown. The installation editor groups identity, stage placement and network/calibration details separately.

Save any new venue or model in its own library first: installation selectors read saved catalogues, not other unsaved drafts. If a venue or model disappears, the list marks **Unavailable link**; the original ID is kept. Select the entry and choose a replacement to repair it, or leave it unchanged for later. An external edit to the installation file blocks saving until you discard and reload. Unsaved installations remain staged when switching sections, and exit offers save, discard or cancel. The installation status wraps in narrow windows instead of forcing the reference workspace wider when a draft changes.

To reuse setup data retained in an older video-projector record, select that saved model and choose **Import old model setup…**. Confirm before copying: matching installation fields are replaced, but the original model and its source checks are not changed. Review the copied values for the actual venue; they are not automatically verified.

This is local reference information, not a show assignment or device control. Addresses and filenames are sensitive operational details; nothing is sent to a server.

Example: Link the Civic Theatre's booth projector to a BenQ LK970 model, then enter its ceiling orientation; a second unit gets a separate entry.

19.2.2 Factory video projector examples

The bundled video list includes professional venue and installation models: Barco G60-W7, G60-W8 and G60-W10; BenQ LK970 and LK990; Christie 4K13-HS, Christie 4K22-HS, Christie DWU15-HS, Christie DWU19-HS and Christie DWU23-HS; Epson EB-PU1007B, Epson EB-PU1008B, Epson EB-PU2010B and Epson EB-PU2010W; Panasonic PT-RZ120 and Panasonic PT-RZ990. For the added models, only the manufacturer and model identity are verified; specifications are intentionally left blank rather than guessed. Check the exact regional variant, lens, mode and current manufacturer documentation before using any equipment values.

These are **reference catalogues only**. They do not assign equipment to a show, control DMX or network devices, configure the second screen, calculate a rig or enable video mapping. Lighting IP addresses and calibration filenames are notes only; nothing connects, downloads or uploads

automatically.

19.2.3 Equipment fields, units and unknown values

Field group	What to enter
Manufacturer / Exact model	Exact brand and variant; duplicate manufacturer/model entries are refused.
Lamp type / Net weight / dimensions	Light-source technology, net kg and width/height/depth in mm, not shipping dimensions.
Power consumption / Lamp rating	Total draw in W is separate from the lighting lamp's W rating; state typical/maximum, voltage and mode in Notes.
Manufacturer-listed resolution / brightness / contrast	Manufacturer-listed resolution, lumens **and brightness standard** (e.g. ANSI), and the contrast measurement method; advertised output need not equal physical chip resolution.
Video throw ratio / lens shift	Minimum/maximum throw ratios and horizontal/vertical shift percentages for the actual fitted lens; add its name and configuration in Notes.
Video inputs / formats	Inputs, supported 3D formats, HDR support and colour gamut, with any firmware or mode caveats.
Video capabilities	Edge blending, warping, genlock and audio output are **Unknown / Yes / No**; an unknown capability is not “No”. Audio output does not imply a speaker.
Other video references	Projector type, pixel pitch in mm, maximum refresh rate in Hz, keystone correction in degrees, sync mode, warranty in years, manufacturer-supported projected area in m^2 and minimum/maximum throw distance in m. These limits depend on lens and conditions, not the actual venue layout.
Lighting identity / output	Fixture type, colour temperature in K and beam angle in degrees; exact variant and optical configuration matter.

Lighting DMX	Documented **DMX channel modes** as a list; **DMX channels (selected mode)** and **DMX address** are optional whole numbers 1–512. A selected channel count must match a listed mode; no DMX is sent.
Lighting travel / focus	Focus range in m; zoom, pan and tilt ranges in degrees as **minimum; maximum**. Pan/tilt describe non-negative travel, not signed world orientation. Infinity-dependent focus belongs in Notes, not an invented finite range.
Lighting effects	Shutter, iris and zoom use Unknown/Yes/No; gobo, colour and prism wheels are lists of documented contents. Dimmer curve and control protocol are reference text.
Other lighting references	Pixel pitch in mm, lux per watt, mounting-hole pattern, fan speed in RPM, maintenance interval in months, firmware version and last-calibrated text. Do not guess device-specific values from a model name.
Network / calibration	Video entries offer the model's network control protocol. Lighting entries may hold an IP address, protocol and calibration filenames as references; nothing is executed or used to control equipment.
Notes / caveats / Source URLs	Lens, mode, revision and measurement limitations, plus one full manufacturer HTTP(S) URL per line.

Blank numeric fields are **unknown**, stored as JSON `null` or an omitted specification, never invented zero. Lists use **semicolons between items**; an individual item cannot contain the separator. Ranges need two finite non-negative numbers, minimum first. Negative or non-finite measurements, invalid ranges, unsupported fields and incomplete source URLs block saving. Unknown source data must not be used as a rigging or electrical-load guarantee.

19.2.4 Manufacturer-checked starter equipment

The starter catalogues contain **16 video projector models** and **two lighting models** (Martin MAC Viper Profile and MAC 500). Added video entries include Barco, Christie, Epson and Panasonic large-venue or installation lines. The newer additions have manufacturer and model identity checked against official product pages or manuals, while other specifications remain unknown unless individually verified. This is a representative subset, not an exhaustive catalogue. MAC 500 has only its identity and profile-spot description checked; technical fields remain unknown.

Example: BenQ LK970 is listed at **5,000 ANSI lumens**, with a **1.38–2.02** throw ratio. MAC Viper Profile has **26/34-channel DMX modes**: do not treat it as a fixture with one fixed footprint. Its **1,000 W lamp rating** differs from **1,225 W typical total consumption**; read the source's operating conditions and variation before planning electrical loads.

Source URLs, check date and **checked fields** describe field-level verification, not certification of an entire model. Editing one checked fact revokes its status; changing sources clears all checks. **Mark entered facts checked…** requires sources and explicit confirmation that every entered fact matches the documented lens, mode and revision. Recording a URL alone does not verify a fact.

19.2.5 Local equipment JSON

The files live beside `preferences.json`, including in **portable-data/** when portable mode is enabled. The personal files are `datashows-user.json` and `projectors-user.json`; bundled factory records remain separate. Damaged files produce a visible error instead of silently resetting the library. Saving is atomic, and changes detected in either the personal or old file while the editor is open block overwriting: discard and reload to review them.

The current schema uses `format_version: 1`, `library_kind` (`datashows` or `projectors`) and a correspondingly named list. Records contain `id`, `manufacturer`, `model`, `specifications`, `notes`, ISO `created_at`/`updated_at`, `checked_on`, `source_urls` and `verified_fields`. Specification names such as `dimensions_mm.width` retain units and distinguish nested legacy facts. The original video `{"datashows": [...]}` object and lighting list-only shape can be read; supported facts are preserved as unverified and missing IDs are derived consistently. Unsupported fields or duplicate models fail explicitly rather than disappearing.

19.3 Audio gain profiles are separate

Open **Project > Venue profiles...** to keep tested room adjustments separate from creative cue settings. **Add** a room, rename it in **Name**, or **Duplicate** an existing profile. Record notes, add setup checks and tick completed items. These are manual operator checks, not automatic tests or permission to start audio.

Venue trim ranges from **−24 to +6 dB**, with its output multiplier shown as a percentage. It multiplies the existing master/cue output gain; it does not rewrite cue volumes, fades, tracks or the playhead. Positive trim always shows a **HEADROOM WARNING**. Qt audio outputs cannot deliver gain above 100% and may saturate; this is not a master limiter, signal meter or guarantee against clipping when cues overlap. Test levels with the house technician before an audience arrives.

List selection only chooses the profile to edit. Choose **Use selected venue** (or **Use no profile** for neutral output), inspect the current/proposed trim and device plus the full edit preview, then **Preview and apply** and confirm. All approved profile edits form one undo step. Cancel leaves the show unchanged. Profiles and their active selection are saved in `project.ccshow`; they travel with the project. **Save selected to library** stores a reusable copy in app preferences; **Add from library** stages a fresh-ID copy in the current project. Apply the staged change to make it an undoable project edit. Saving a profile to the library does not change the open project or activate a venue.

Switching profiles, changing active trim/device notes, deleting the active profile and venue-affecting undo/redo are blocked while audio, auditions, fades or scheduled waits exist. Stop all first. Notes and inactive profile edits do not change live gain. **Venue...** in the Show Player header opens the same profiles dialog. An active profile's name and trim appear in the header; positive trim shows **HEADROOM WARNING** and the trim instead, with the full venue description on hover. With no active profile no venue text is shown, and output is neutral. The chosen profile stays active only in this project; a new project does not automatically activate a library profile. **More > Reset venue trim...** confirms a saved, undoable change to 0 dB, also blocked during playback: resetting a negative trim would increase volume. It does not reset the project master or per-cue live trim.

Device note is an optional preferred output device **name** only. **Check available device names** reports a mismatch against Qt's available descriptions. A matching name does not prove that audio uses that device. Curtain Call applies **no routing or EQ**; configure and verify the actual output in the operating system or audio interface manually.

Example: duplicate “Small hall”, name it “Tour stop — civic theatre”, set −6 dB (50.1% output) and add

“Check house output” and “Listen from back row”. Check the device note, inspect the preview and activate only with audio stopped. **Export checklist…** exports the selected draft (not necessarily the active venue) to `exports/venue-<name>-checklist.txt`, with notes, trim, the device disclaimer and `[x]` / `[ ]` checks. Confirm export explicitly; a previous same-named checklist is replaced. Exporting alone neither applies the draft nor starts audio.

19.4 Keep room setup separate from creative cue design

The production's venue text is a descriptive project field. An active venue profile additionally contributes a saved output multiplier. Choosing a list row for editing does not activate it; the staged dialog makes this distinction explicit. Check current and proposed profile, trim, device note and checklist in its preview.

Venue control	**Operator responsibility**
Add / Duplicate / Name	Give each tested room an unambiguous local identity
Trim (−24 to +6 dB)	Test the multiplier with the house technician
Device note / Check available device names	Compare names, then verify routing in the operating system
Notes / setup checks	Record manual evidence and tick only checks actually completed
Use selected venue / Use no profile	Stage active or neutral output before approval
Preview and apply	Inspect and confirm one undoable project change
Save selected to library / Add from library	Reuse independent profiles without hidden live inheritance
Export checklist	Share the selected draft's setup instructions as text

A profile copied into a new show gets a fresh ID and starts inactive. It is not linked live to the app library. Editing the library later does not rewrite all previously created productions or apply a room to the open show.

19.5 Gain is not automatic calibration

Negative trim reduces output; positive trim requests more output and displays a headroom warning. The demo's 90% master and 70% cue setting remain creative values while −6 dB applies a separate multiplier of about 0.501. Qt output saturation, multiple simultaneous voices and recording peaks still require listening and proper external gain staging. There is no automatic routing, EQ or master limiter.

Warning: Returning −6 dB to 0 dB increases output. Reset venue trim is blocked while any audio, audition, fade or scheduled wait exists, even though “reset” may sound like a harmless housekeeping action.

Neutral/no-profile output is 0 dB. Stop all, review the reset, approve it, then sound-check before beginning. Undo/Redo that affects active venue output is subject to the same idle requirement. Notes or inactive-profile edits do not alter live gain.

20 Show-day checklist and safe resumption

Example: Run The Lighthouse Keeper from cue **1** through Phones Announcement during sound check. Stop all, reset temporary strip trim, set the playhead back to **1**, and reset the show timer. At the performance doors-open call verify **70%, Smooth 5 s, loops until stopped** on the next-cue card before GO.

20.1 Before the house opens

Check	Evidence, not assumption
Correct production copy	Name, production version and latest rehearsal corrections agree
Backup available	An independent saved project copy can be opened
Media present	Preflight has no missing tracks or required targets
Actual output verified	A controlled cue is heard through the intended house system
Venue/profile checked	Trim and device notes agree with the technician's setup
Cue chains checked	Infinite loops have deliberate exits; automatic overlaps are rehearsed
Operator view ready	Readable font/columns, script follow and next-cue card visible
First cue selected	Correct armed cue, cue-in instruction and script anchor
Runtime controls ready	Tests stopped, live trims reset, clock/timer and recorder understood

Preflight catches missing files, start cues without tracks, fade/stop cues without valid targets, hotkey clashes and an empty selected-ducking list. A clean result is a consistency check, not acoustic approval. It does not validate amplifiers, headroom, a speaker cable, actor pace or an operating-system device selection.

20.2 During an interrupted rehearsal

1. Stop all or Panic when the immediate situation requires it.
2. Ask which dramatic point the team will resume; do not infer it from the last cue that happened to be highlighted.
3. Check which backgrounds must be re-established and whether a hotkey would restart an existing start cue.
4. Move the playhead without firing to the agreed next cue. Read its action and target: a later Stop cannot recreate an earlier ambience.
5. Restore temporary levels deliberately, verify the script paragraph and call GO only after the stage manager's agreed instruction.
6. Drop a pin or note when useful for explaining the interruption later.

Warning: Skipping forward moves the playhead, not the state of the whole show. It does not run skipped setup cues, stop existing voices or recreate the expected background automatically.

Do not redesign a show live because the card looks different from what you expected. Pause, inspect the source settings in Cue Creator and rehearse a corrected sequence. Keep the run recorder as evidence, not as an automatic replay system.

21 Exports and printing

Example: For The Lighthouse Keeper export **Script PDF**, **Printable cue sheet (PDF)** and the selected **Dress Run 2** rehearsal PDF. Compare each running header and the footer **Exported date/time — Curtain Call version — Page n of N**. The project version **v1.2 - after the dress rehearsal** is a production label, not the application version in the footer.

Deliverable	**Where / purpose**
Rehearsal text or PDF	exports/rehearsals/; corrections and statuses for the team
Script ODT / Markdown / PDF	exports/script.odt, script.md, script.pdf; editable or paginated script
Cue sheet CSV / PDF	exports/cue-sheet.csv, cue-sheet.pdf; operator paperwork
Session log	exports/logs/; complete run events, including pins and detail
Web version	exports/audio-files/; audio-show.js and copied audio, lossy compatibility
Export all / ZIP	New inventoried deliverables folder/ZIP, including a clean show copy by default; not a full historical backup
Clean show copy for sharing	Minimal native `.ccshow` with used assets and resolved settings; original stays unchanged

Use Script's **Print layout** to inspect cue markers and page numbers before distributing a script PDF. Export rehearsal reports after updating statuses: an earlier PDF does not update when the project changes.

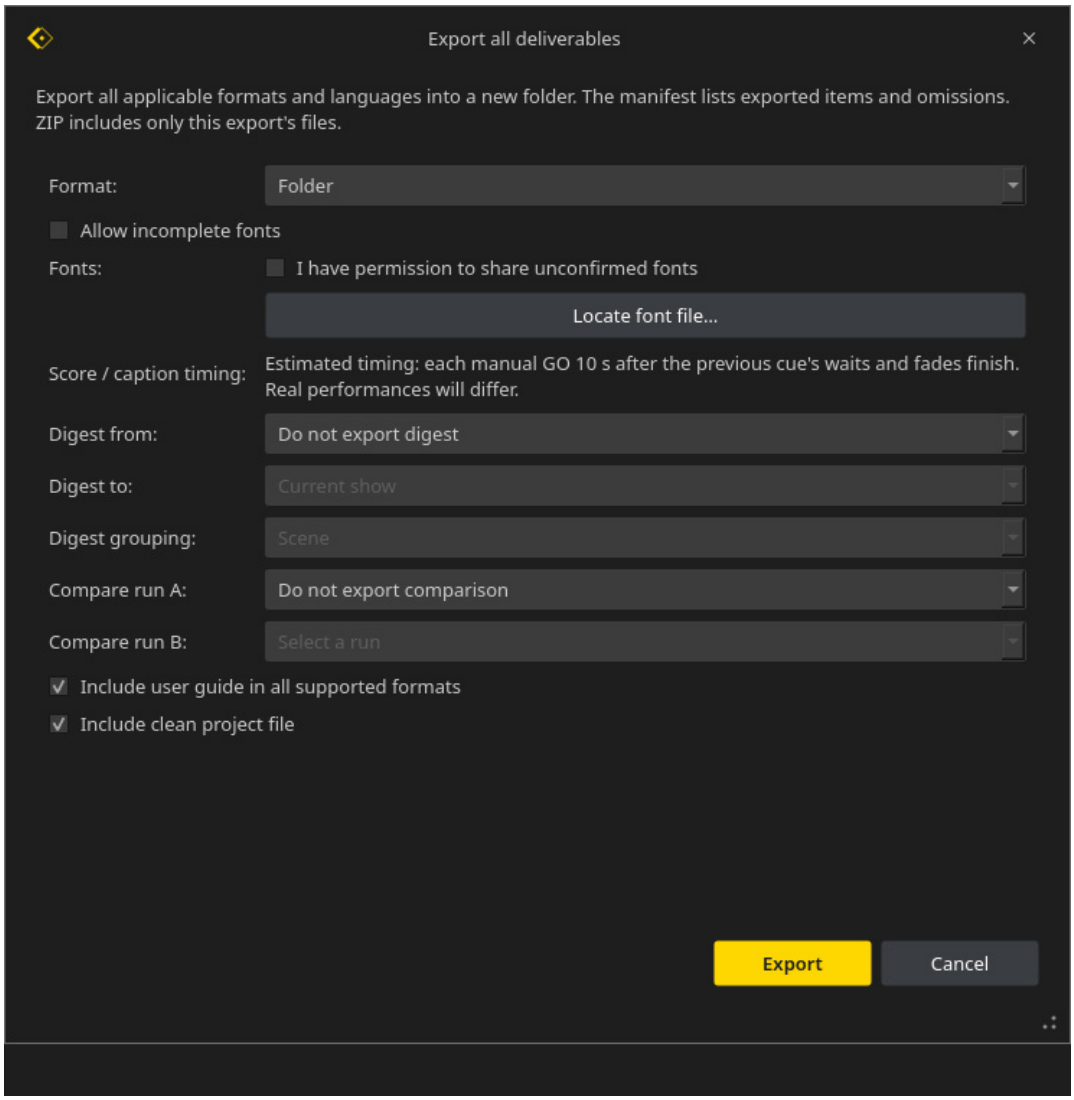
The old web format cannot represent all desktop cue semantics, including pre-wait. Re-test any exported web show; keep `.ccshow` as the authoritative editable production. Export all now generates a new set of deliverables rather than collecting old files from `exports/`; its ZIP contains only that operation's manifest-listed outputs.

The web version keeps all audio in one flat folder. When two media files used by cues have the same file name (for example `act1/rain.wav` and `act2/rain.wav`), the first keeps its name and the next becomes `rain (2).wav`, `rain (3).wav`…; names are compared without case, as Windows and macOS do. Each exported cue refers to its own copy, so nothing is overwritten. Tracks that use the very same audio file share one copy.

21.1 Export all formats and share a clean show

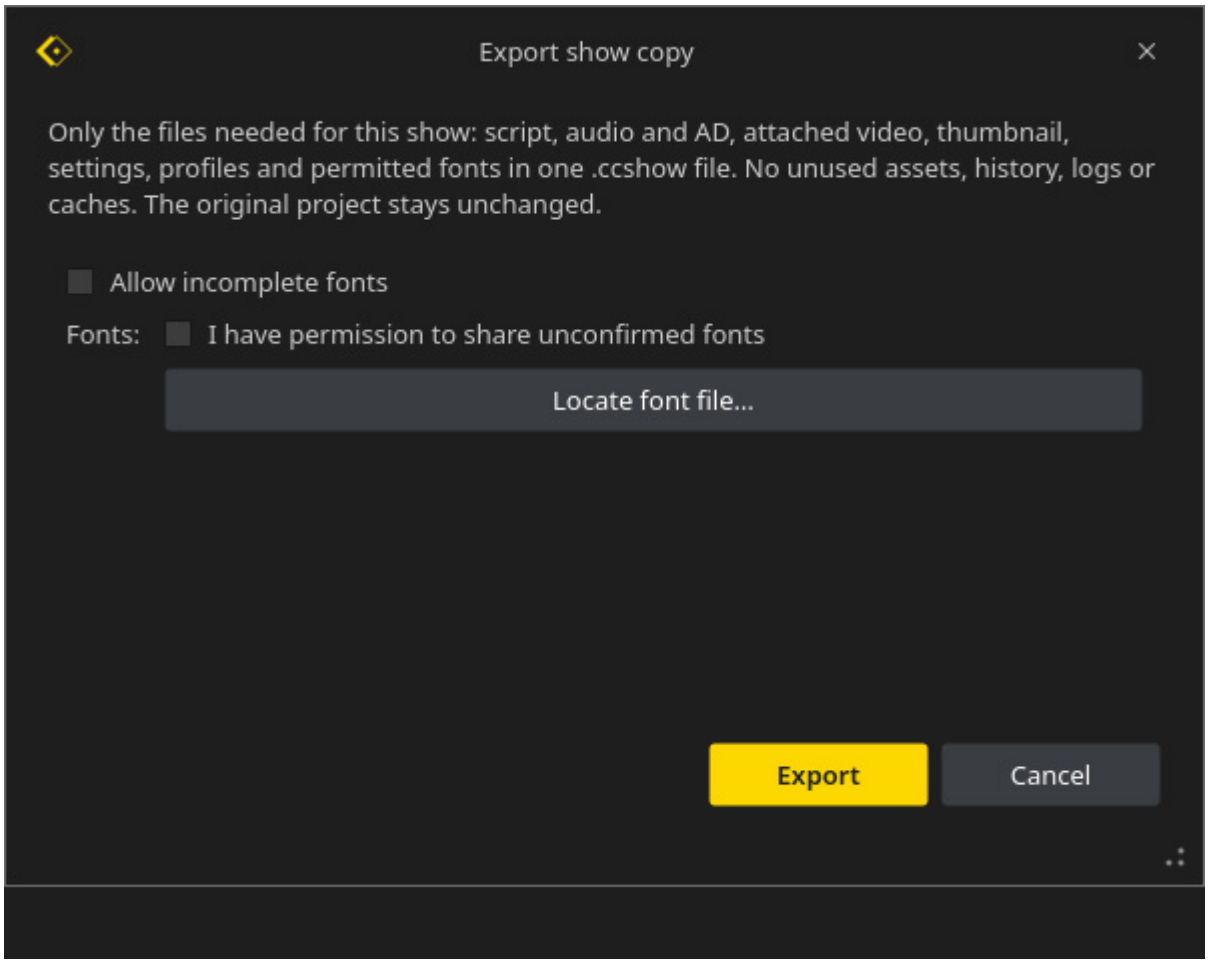
Project > Export > Export all offers folder or ZIP output. The ZIP action and **Zip and send by email…** use ZIP. Each export gets a new folder such as `exports/all-exports/`; previous exports and older ZIPs are not overwritten or swept into it.

All applicable existing formats are included: scripts and character excerpts (ODT, Markdown, TXT and PDF), cue sheets, show score, original and translated subtitles (SRT/WebVTT), audience captions and guide, rights reports, saved rehearsal reports, venue checklists, log, web version and project profile snapshots/palettes with permitted fonts. The user guide (PDF, Markdown with its images, TXT and MHTML) and a clean native `.ccshow` are included by default; you can explicitly decline them.



The score and audience captions use the timing currently selected in **Show Score**. Choose the digest revision range/grouping and run comparison pair in the export dialog if you want those reports. No-content, declined and failed items are recorded separately in `delivery-manifest.json`, alongside file sizes, checksums, languages, timing and font warnings. Failed outputs never produce a success ZIP; a partial deliverables folder may remain for inspection.

Clean show copy for sharing… creates one native `.ccshow` file with its dependencies. It keeps the current script/document, cue references (including disabled cues), used audio, selected current AD recordings for all saved languages, attached subtitle reference video and offset, thumbnail, and effective caption/subtitle/HoH and active venue settings. Required profiles are resolved into the native project; unrelated user profile libraries are not copied. The thumbnail retains the standard filenames required by the recent-project list and settings preview. Media retains its original filenames and in-project folder structure. Required external media is brought into the copy; genuine case-insensitive filename collisions use separate folders, not renamed files. References are updated where necessary, so playback and attachments do not depend on the source computer's paths. Non-portable names must be corrected before sharing; export does not silently rename them. Old takes/versions, unused assets, caches, recovery files, backups, exports, logs, rehearsals and script history are omitted. The source remains untouched, including unsaved edits; this is not a full backup.



Required media is checked before exporting a clean copy or deliverables that include one. Missing files are listed with their track name, AD character and language, or script/video role. Restore the file or correct its assignment before retrying. The app opens the correction section for the first unavailable item. For AD it selects the character, recording and language in **Audio description**; library audio opens its track, scripts open Script, and video opens Subtitles. An unavailable thumbnail opens the project settings' Details page. For AD, regenerate/import the required recording or deliberately remove its missing saved entry if it is no longer needed. Export never silently drops a language or replaces an edited recording with an older take.

Fonts whose licence metadata permits redistribution are included automatically, without a permission question. **I have permission to share unconfirmed fonts** is one confirmation covering all required fonts with unconfirmed licences for this export only; there are no per-font permission prompts. Restricted fonts are never copied, even with confirmation. **Locate font file…** selects a matching file without changing the source project. Missing fonts/glyphs or imitated faces normally block a complete package; **Allow incomplete fonts** explicitly permits a copy with warnings and an incomplete manifest. Missing required media or broken runtime references still stop the clean copy.

Progress and Cancel are available; cancellation leaves no unfinished clean project file or temporary project folder. Open the shared `.ccshow` directly without extracting it. A completed deliverables folder may remain if its separate ZIP creation is cancelled. Check audio/video codecs, output devices, display layout and font rendering on the recipient computer before rehearsal or performance. A complete file inventory is not show-readiness or cross-platform playback certification.

21.2 After every export: open, show or email

Every export is written in a sibling `<project name> -- exports/` folder beside the project folder or single-file project, with subfolders such as `rehearsals/` and `logs/`. Existing exports in older locations remain there. When it is done, an **Export complete** question shows the path and offers:

Project > Export > Open exports folder appears when that folder exists.

Choice	What happens
Open file	Opens the exported file in its default program (PDF viewer, spreadsheet, text editor)
Open folder	Opens the folder that holds it in the file manager
Send by email…	Opens a new email draft in your own email program with the file attached where possible
Close	Does nothing more; the file stays in `exports/`

Exports that are a whole folder (Web version, Export all) only offer **Open folder**. **Zip and send by email…** and the Notes **Send by email** button skip the question and go straight to the draft.

Attachments. On Linux the draft is opened with `xdg-email`, which attaches the file in Thunderbird, Evolution, KMail and other programs that support it. Elsewhere (Windows, macOS, webmail) an email link cannot carry a file, so the draft opens together with the file's folder: drag the file into the draft. The draft body names the file and its folder either way. Always check the attachment and the recipient. Nothing is ever sent by the app.

21.3 Common PDF page furniture

All PDF exports—script, printable cue sheet, show score, rehearsal report and help—use **generous 20 mm margins on every side**, a running header with the document title, and a footer with **export date/time**, **application version** and **Page n of N**. The export date is the local time at export. The version comes from the app's single version source, `curtaincall/version.py`, not the editable project version. These details let a team distinguish old paperwork from newly exported copies. Large pictures, such as the screenshots in this guide, are shrunk to fit inside the margins and kept whole on one page.

PDFs use readable dark text on paper, even though the live app and its illustrations use a dark theme. Script and rehearsal reports use portrait A4; the cue sheet uses landscape A4 for wide operator

information; the show score uses landscape A3. Inspect the preview or saved PDF before printing: a footer cannot guarantee the content is the latest project state.

21.4 Complete export format reference

Format / command	Output and purpose	Important limit
Script Export ODT	exports/script.odt; styled editable stage script with scene/cue margin comments	Re-import ignores generated annotations and keeps internal speech breaks
Script Export Markdown	exports/script.md; portable editable text with scene/cue quoted annotations	Re-import ignores generated annotations; internal breaks use ` `
Script PDF / Print layout	exports/script.pdf; paginated script with cue/scene markers	A printed document cannot follow live edits
Cue sheet (CSV)	exports/cue-sheet.csv; spreadsheet-readable operator cue data	Spreadsheet formatting is external
Printable cue sheet (PDF)	exports/cue-sheet.pdf; landscape performance paperwork	Includes human action/settings, notes and script context
Show score (PDF)	exports/show-score.pdf; A3 landscape poster of the whole show	Programmed levels on estimated or recorded timing, not measured sound
Rehearsal Export report	exports/rehearsals/; UTF-8 corrections report	Export reflects current selected rehearsal statuses
Rehearsal Export PDF	exports/rehearsals/; full paginated correction report	Read missing-reference warnings before sharing
Change digest Export text	exports/rehearsal-digest-....txt; rehearsal snapshot facts	Missing snapshots cannot supply historical changes
Run timeline Export CSV	exports/; selected run comparison	Event offsets are not recorded audio
Venue Export checklist	exports/venue-...-checklist.txt	Can describe a selected draft, not necessarily active output
Session log	exports/logs/; complete timestamped log	Screen filters and Clear do not remove exported entries

Web version	exports/audio-files/; audio-show.js and copied audio	Lossy compatibility; no desktop-equivalent pre-wait semantics
Export all	New folder with every applicable supported format, profile snapshots and optional clean native copy	Select ranges/timing deliberately; not a full historical backup
Export all as zip archive	ZIP of this operation's inventoried outputs	No stale exports or previous ZIPs; inspect omissions/warnings
Clean show copy for sharing…	Minimal runnable `.ccshow` with script, used audio/AD/video, settings and permitted fonts	Recipient codecs/devices need checking; original untouched
Zip and send by email	Export ZIP and compose a draft with it	Attached where the system allows; operator checks and sends
Help PDF / Print	Whole manual, chapter or selected document fragment	Review illustration size in preview
Help MHTML web archive	Standalone illustrated manual, chapter or selection	Includes HTML, inline styles and images; requires an MHTML-capable browser or viewer
Help Markdown / text	Editable or plain-text manual copy	Plain text has no visual images; Markdown image refs need accompanying assets

21.5 Choose a paper workflow

For a director who is revising dialogue, use ODT or Markdown. For a sound operator calling from printed pages, use script PDF and the cue sheet. For a rehearsal follow-up, select the correct rehearsal and export its report after reviewing note statuses. A change digest communicates exact differences; a run comparison communicates execution pacing. They answer different questions.

Use Script **Print layout** to inspect cue markers and pagination, then print or save PDF. Help **Print** opens its paper preview with page navigation, **Print…** and **Save PDF…**. The guide scope can be Whole guide, This section or Selected text. With a project open, help exports go into `exports/`; without one a save window opens in the home folder.

Tip: Put the document title, export timestamp and application version in your team's check before

using a printed copy. Also check the production label: an app release number does not tell you which script revision the director approved.

21.6 Distribute without losing the editable show

Save project edits first, then export. Exported PDFs, CSV and text are snapshots; they do not update after a later cue or note edit. A ZIP built from exports is a communication bundle, not a substitute for a complete project backup. Keep a saved `.ccshow` copy, or the whole folder for a folder project, for rollback and touring.

Email commands only open a draft in your own email program (with the file attached where possible, otherwise next to its folder). Check the attachment and the recipient. Review project notes and third-party media files for permissions before distributing them. No background email service sends the show for you.

22 Output log and troubleshooting

Example: If cue **4A Stop House Ambience** seems slow in The Lighthouse Keeper, read its **2 s pre-wait** and **5 s fade** before calling it a fault. Filter the log for **House Ambience**, switch to Verbose to inspect waits, then export the complete log even if the screen is filtered.

Ctrl+J shows or hides output. Severity, verbose-detail and search filters change only what you see; exported logs retain all entries. Double-click a cue-linked entry or choose Jump to cue to inspect its cue in Cue Creator without firing it. Entries from removed or other projects cannot be navigated.

If a track will not play, check its missing-file flag, format and output-log errors. If GO skips a cue, check that it is armed. If the script stops following, press the follow cues button in the right-hand rail after manual scrolling.

22.1 Diagnose one layer at a time

Symptom	Check / next action
"Project needs a newer Curtain Call" on open	The project was saved by a newer release. Update Curtain Call; the file has not been changed
GO appears to do nothing	End of show (click a cue or press Up to replay)? Correct playhead? Armed? Pre-wait still counting (an extra GO is ignored)? Memo/delay/placeholder rather than audio?
Track cannot play	Library missing-file marker, selected track, permissions, media-format error in output
File previews but cue is silent	Cue volume, master, live mute/trim, ducking and fade target
Fade/stop has no effect	Target is the correct currently playing start cue, not a library track
Follower never starts	Continue none? Infinite start with auto_follow? Audio stopped rather than naturally ended?
Follower starts during a fade	Non-start auto_follow does not wait for fade completion; adjust post-wait
Background stays low	A ducking source is still playing, muted or faded to zero; stop that source
Batch relink leaves files unresolved	Exact stored filename has zero or multiple matches; manually Relink
Links vanish on re-import	Paragraph wording changed or was removed; assign links again
Script no longer follows	Press the follow cues button in the right-hand rail after manual scrolling; check that cue has a link
Save or export fails	Destination writable? Disk space? Read the error and preserve an independent backup

If a codec fails, try a known-good WAV copy produced with an audio editor, relink and test it. Merely renaming an extension does not convert audio. If there is no sound from any file, inspect operating-system output routing and the physical signal chain. Do not increase every cue to maximum to compensate for an unverified device or amplifier.

22.2 Keep evidence, not just the visible filter

Use output severity/search filters to isolate a failure. Verbose detail can show scheduled followers and fades. Filtering does not remove log records; export the complete log for investigation. Cue-linked events can navigate to the relevant cue without firing it. A deleted cue or an event from another project cannot be resolved in the current show.

When reporting a problem, include the cue number/name, action taken, exact output error, operating system and audio format. Do not send media files or project notes you lack permission to share. Keep the original project intact while testing a copied version.

22.3 Understand the output controls

Control	What changes	What stays preserved
Normal / Verbose	Whether detailed waits/fades/loops are visible	Full session log retains detail
Severity filter	Visible info/warning/error subset	Hidden entries remain exportable
Search	Case-insensitive visible message/time/severity/source matches	Incoming matching entries are filtered too
Clear	Current visible output presentation	Session export is still complete
Jump to cue / double-click	Selects the referenced cue in Cue Creator	No audio fire or live-playhead movement
Export log	Writes the full session to exports/logs/	Includes filtered-out events and pins
Ctrl+J	Shows/hides the pane	Does not enable/disable engine logging

Every entry carries severity and source (such as system, cue or preflight). Cue-linked events remember the originating project. An old event cannot silently navigate into a different show with a similar cue number. After renumbering, references use stable identity rather than the printed number.

22.4 Respect the playback limitations

QtMultimedia timing is not sample-accurate. This app is not a multi-output router, latency compensator, acoustic measurement system or hardware audio limiter. A successful silent simulation does not guarantee media decoding. Test the actual file format and operating-system output on the show computer.

For a missing recording, relink through Library rather than rewriting every cue. For a wrong level, trace cue volume, fade, ducking, master, venue and live trim before raising gain. For a timing surprise, trace pre-wait, continuation, post-wait and the next cue's pre-wait. For a missing anchor, inspect the approved script revision and explicit Unlink choices.

Warning: Do not raise every fader to compensate for silence until output routing, mute and physical cabling have been checked. A delayed or newly connected output can otherwise become unexpectedly loud.

23 Frequently asked questions

Example: In **The Lighthouse Keeper**, Cue **3 House Ambience down to 30%** changes cue 1, not the AMB track. If you need that ambience gone, fire **4A Stop House Ambience** instead. If you need an immediate emergency cut, use Panic rather than expecting a fade-to-zero cue to cancel all playback.

23.1 Does opening another mode stop audio?

No. Script, Library, Cue Creator and Notes remain available while audio runs. Emergency Escape/ Shift+Escape work in app modes. Help is a separate window: close it or refocus the performance window before relying on performance Space/hotkeys.

23.2 Why does GO skip a row?

It may be disarmed. GO and automatic/group/hotkey paths skip disarmed cues. Check Armed for playback in Identity. A collapsed fired row is a different display choice: collapsing does not delete or disarm the cue. A disarmed row is easy to spot without checking anything: it is dimmed, italic and its state says **DISARMED** — see *An armed cue can execute* above.

23.3 Why did a follower play during a fade?

Auto-follow on non-start types acts immediately like auto-continue, not after the fade has finished. Post-wait must account for the desired fade interval. A delay cue can express a wait explicitly. Test fire suppresses followers, so an audition is not a complete timed-chain test.

23.4 Can a script line trigger itself automatically?

No. A script anchor provides context and navigation; it does not hear actors or predict their pace. What-if can resolve a paragraph's simulated time only when an executed linked cue provides the anchor.

23.5 Why is the background still ducked after muting the announcement?

Mute changes output but leaves its playback active. The ducking source must finish or be stopped for release. Fading it to zero can likewise leave an active silent voice. The lowest concurrent duck gain wins rather than multiplying sources.

23.6 Can I share only a .ccshow file or exports ZIP?

A saved single-file `.ccshow` carries its project data and media. For a folder project, copy the entire folder; its `project.ccshow` JSON alone is not enough. An exports ZIP is not a full historical project backup.

23.7 Why are my user recipes missing on another computer?

Recipes live beside preferences in recipes.json. They are not embedded in the show. Venue-profile library is also app-level, but copied project profiles travel inside project settings. Copy the appropriate app data separately when needed.

23.8 Does a matching venue device name route audio there?

No. It checks descriptions only. Verify operating-system routing and house audio manually. A profile trim is not EQ, calibration or clipping protection.

23.9 Will PDF import read a scan?

No. Optional `pdf-import` support extracts selectable text. Scanned image pages need OCR elsewhere. ODT/DOCX paragraph styles generally communicate script elements better than the visual layout of a PDF.

23.10 Will exported paperwork change after I edit the show?

No. Re-export the relevant report. Use its title, export timestamp, app version and production label to avoid handing an old copy to the operator.

24 Keyboard reference

Example: In The Lighthouse Keeper use **Ctrl+3** to inspect cue 2; use **Ctrl+4** to return to performance. With the performance window focused, **p** fires Phones Announcement directly. **Ctrl+5** opens the correction backlog without cutting the currently sounding ambience.

Key	**Action**
F1	Open this searchable user guide
Ctrl+P / Ctrl+E (in the guide)	Print / export the guide: whole, section or selection
F11 (in the guide)	Maximise or restore the guide window
Ctrl+1	Script
Ctrl+2	Library
Ctrl+3	Cue Creator
Ctrl+4	Show Player
Ctrl+5	Rehearsal Notes
Ctrl+6	Show Score
Ctrl+7	Subtitles
Ctrl+8	HoH Accessibility
Ctrl+9	AD — Audio Description
Ctrl+S	Save project
Ctrl+Shift+S	Save as: an independent named copy
Ctrl+Z / Ctrl+Shift+Z	Undo / redo
Ctrl+J	Show / hide output
Ctrl+Shift+L	Lock the screen: the show keeps playing
Space	GO in Show Player
Ctrl+Space	Pause / resume the show without stopping it
Up / Down	Move Show Player playhead
Esc	Stop all

Shift+Esc	Panic
F11	Full screen

24.1 Editing and navigation shortcuts

Key / context	**Meaning**
Ctrl+N / Ctrl+O	New / Open project using the selected file-dialog preference
Ctrl+W	Close project, subject to unsaved-change handling
Ctrl+Q	Quit on platforms with that standard quit shortcut
Ctrl+C / Ctrl+V in Cue Creator	Copy selected cues / paste after current cue
Delete in Cue Creator	Remove selected cues; project Undo can restore them
Enter in Script	Continue inside Dialogue; otherwise create the usual following element
Alt+Enter in Script	Create the usual following element, including Character after Dialogue
Shift+Enter in Script	Insert an internal line break without creating another element
Tab / Shift+Tab in Script	Cycle current paragraph kind forward/backward
A programmed cue hotkey in Show Player	Fire that armed cue directly, not a text search
Enter in Help search	Next global match, wrapping at the end

Cue hotkeys never take over application keys. A cue hotkey that matches an application shortcut (Space, Ctrl+Space, Up, Down, Escape, Shift+Escape, F1, F11, Ctrl+1 to Ctrl+6, Ctrl+J and the File/Edit shortcuts such as Ctrl+S or Ctrl+Z) is **not registered**: two shortcuts on one key would make Qt fire neither, which could silence GO or Panic. When two cues share a hotkey (capitals do not matter), only the first armed one responds. The Cue Creator warns under the Hotkey field as you type ("Reserved for GO", "Already used by cue 3", or an unknown key name), and the Show Player preflight lists every ignored hotkey, including in imported or older shows.

The menu uses platform standard New/Open/Save/Undo/Redo/Preferences/Quit bindings. On macOS some standard bindings use Command rather than Ctrl. Read the menu's actual shortcut labels on the show computer. Prefer its displayed Redo binding when the operating system uses a variation;

project Undo/Redo is the authoritative history, including Script and rehearsal-note edits.

Emergency keys cancel tracked waits as well as audio. They do not require the “ask before quitting” confirmation because stopping and quitting are different operations. Full screen is an app display choice; Help F11 maximises its own window.

25 Cue field reference

Example: In **The Lighthouse Keeper**, Read the Identity, Playback, Ducking, Timing and Notes tabs for cue **2 Phones Announcement**. Match **100% volume**, **duck_gain 0.25**, **attack 0.2 s**, **release 4 s**, **hotkey p** and its sound caption to the reference below. Only the selected start cue supplies the ducking source.

25.1 Every cue field

The defaults below are the model's backward-compatible defaults for an absent field in an older show. **New cues created in the UI also copy project defaults**: a Start copies new-cue volume; non-start cues receive the project default fade and a zero volume target. The demo defaults are 80% and 3 s. Existing cues do not inherit new volume/fade defaults after creation. Ducking remains inherited only while Use project ducking defaults is on.

Field / stored name	Meaning	Model default	Applies to
Type (`cue_type`)	Which instruction GO executes: start, fade, stop, stop_all, group, delay, memo or placeholder.	start	All
Number (`number`)	Free-text operator label, such as 4A. Next follows list order, not this label.	Empty	All
Name (`name`)	Readable action name used in cards, tables and reports.	Empty	All
Armed for playback (`is_armed`)	False skips execution through GO, hotkeys, groups and automatic paths.	True	All
Group (`group_name`)	Group membership/ name. A group cue matches start-cue membership exactly.	Empty	All; membership on start
Library track (`track_id`)	Stable identity of the recording to start, not a filename you type.	Empty	Start
Target cue (fade/stop) (`target_cue_id`)	Identity of the playing start cue to change or remove.	Empty	Fade, Stop
Group name to fire	Name of the group	Empty	Group

(group_target)	whose armed start cues fire directly.		
Delay (s) (delay_seconds)	0–3600 s before a delay schedules its follower; pre/post waits remain separate.	0.0	Delay
Volume % / fade-to (volume_percent)	0–150%; start level or fade target, not a guaranteed acoustic output.	100.0	Start, Fade
Fade in (s) (fade_in_seconds)	0–600 s entry ramp from silence.	0.0	Start
Fade out (s) / duration (fade_out_seconds)	0–600 s exit/target fade. A start does not automatically fade at file end.	0.0	Start exit; Fade, Stop, Stop all
Fade in curve (fade_in_curve)	One of five curve shapes through the entry level change.	linear	Start
Fade out curve (fade_out_curve)	Shape used toward a lower/target level or stop.	linear	Start exit; Fade, Stop, Stop all
Loops (0 = infinite) (loop_count)	1–999 total passes; 0 endless, not zero playback.	1	Start
Loop crossfade (s) (loop_crossfade_seconds)	0–60 s overlap between passes; engine caps at half recording duration.	0.0	Looping Start
Ducking: lowers (duck_mode)	off, selected or all other active cues. Source never targets itself.	off	Start source
Selected ducking target list (ducked_cue_ids)	Stable IDs selected in the Ducking tab; empty selected mode is a	Empty	Start with selected ducking

	preflight warning.		
Use project ducking defaults (`duck_use_project_defaults`)	True resolves current project gain/attack/release instead of the next three values.	True	Start with ducking
Duck: target gain (`duck_gain`)	0.05–1 multiplier on target levels, not dB or absolute cue volume.	0.5	Start with custom ducking
Duck: attack (s) (`duck_attack_seconds`)	0–60 s to lower target gain while source becomes active.	0.3	Start with custom ducking
Duck: release (s) (`duck_release_seconds`)	0–60 s return after the source leaves active playback.	1.5	Start with custom ducking
Pre-wait (s) (`pre_wait_seconds`)	0–3600 s between fire request and this action.	0.0	All
Post-wait (s) (`post_wait_seconds`)	0–3600 s after continuation is scheduled, before follower fire.	0.0	All with continuation; Delay
Continue (`continue_mode`)	none, auto_continue or auto_follow; natural media end only matters on Start.	none	All; Delay has its own continuation
Hotkey (`hotkey`)	Direct single-key fire in Show Player; check collisions and reserved controls.	Empty	All armed types
Linked script line (`script_line_id`)	Stable paragraph identity set via Script/Link/drag. Context only, not an automatic trigger.	Empty	All
Script marker links	Cue-owned track	Empty	Start

(`script_marker_links`)	marker and script paragraph associations for silent navigation.		
Sound caption (`sound_caption`)	Human-authored intent shown on performance cards/strips and cue sheets.	Empty	All; most useful on Start
Audience captions (`audience_captions`)	List of timed on-screen captions for the audience screen, each with its own text and timing.	Empty	All
Cue in (note) (`cue_in_note`)	Operator instruction describing when to call the cue.	Empty	All
Cue out (note) (`cue_out_note`)	Human exit/end instruction; does not schedule an automatic stop.	Empty	All
Memo (`memo_text`)	Multiline annotation used as a memo action description, without audio.	Empty	Memo
Internal cue identity (`id`)	App-generated stable ID; ordinary operators do not edit it.	Generated ID	All; internal

25.2 Read defaults without accidentally rewriting a cue

A plain model default of zero fade does not mean a UI-created Stop has no fade: the creator copies the project's default fade. Likewise a model default of 100% is not the demo's UI new-cue default of 80%. The fields are saved as concrete values after creation, except the deliberate project-ducking-default option.

Hidden or irrelevant fields can remain stored when you change a cue type. Inspect the fields relevant to the new type; an old track assignment does not make a Memo play audio, and an old target ID does not give a Start a Stop action. Internal IDs are reference plumbing; use the app's Link/Relink controls instead of hand-editing them in your only production file.

26 Library fields and project settings reference

Example: The Lighthouse Keeper settings name **Estuary Theatre Collective**, venue **Demo Theatre, Main Stage**, operator **Sound operator: Rita Campos**, master **90%**, new-cue volume **80%** and default fade **3 s**. Its AMB track describes the recording; cue 1 separately chooses **70%**.

26.1 Every library field

All text fields default to empty. Library metadata does not change audio file contents. Empty tags can be filled with Use file info; authored values are retained.

Field	Meaning	Default
`name`	Readable library name.	Empty
`code`	Optional short code, such as AMB, ANN or STM.	Empty
`file_path`	Project-relative audio location; managed by import/relink.	Empty
`audio_versions`	Retained audio-version paths; cleanup preserves these while the track is retained.	Empty list
`duration_seconds`	Measured duration; 0 means not yet known, not a silent zero-length file.	0 s (unknown)
`artist`	Performer/recording artist.	Empty
`composer`	Composer credit.	Empty
`publisher`	Publisher or rights administrator supplied by the operator.	Empty
`rights_source`	Source of the recording or licence, for rights paperwork.	Empty
`paid_rights_status`	Operator's paid-rights classification: unknown, `paid` or `not_paid`.	unknown
`album`	Collection or score name.	Empty
`year`	Free-text recording/release year.	Empty
`genre`	Descriptive genre tag.	Empty
`copyright`	Rights holder/licence reminder; not automated permission verification.	Empty
`notes`	Recording/source notes rather than playback settings.	Empty
`id`	Generated stable identity that cues reference.	Generated ID

26.2 Every project setting

Project > Project settings has Details, Credits, Notes, Defaults, Audience captions, Subtitles, Second screen and Venue tabs. Profiles are managed in Project > Venue profiles. Defaults in this table are the initial model values; the New projects preference page can copy different values at creation, and projects then remain self-contained.

Setting / stored name	Location and meaning	Model default
`name`	Details: Production name; displayed in window/status and exports.	Untitled show
`company`	Details: Theatre group or production company.	Empty
`venue`	Details: Descriptive venue text, separate from active venue profiles.	Empty
`venue_details`	Venue: Independent physical-venue record copied from Libraries > Venues or entered locally; contact/dimensions/ provenance, never routing or gain.	null
`operator`	Details: Sound operator/stage manager credit.	Empty
`version`	Details: Free-text production revision, not the app version.	Empty
`credits`	Credits: Role/name pairs for creative and production credits.	Empty
`notes`	Notes: Timestamped project-level production notes, separate from rehearsal corrections.	Empty
`thumbnail_image_path`	Details: Relative path to the project's square thumbnail, stored inside the project folder.	Empty
`master_volume_percent`	Defaults: 0–100% multiplier on all playback.	100.0
`default_fade_seconds`	Defaults: New non-start cue fade and Escape Stop all fade, 0–600 s.	3.0

`default_cue_volume_percent`	Defaults: New Start cue level, 0–150%; does not rewrite existing cues.	100.0
`default_duck_gain`	Defaults: Inherited duck multiplier, 0.05–1.	0.5
`default_duck_attack_seconds`	Defaults: Inherited duck attack, 0–60 s.	0.3
`default_duck_release_seconds`	Defaults: Inherited duck release, 0–60 s.	1.5
`venue_profiles`	Venue profiles dialog: Project-local profiles, manual checks, notes and trims.	Empty
`active_venue_profile_id`	Venue profiles dialog: Selected active profile; empty is neutral 0 dB output.	Empty
`subtitle_maximum_characters`	Subtitles tab: Maximum per line or legacy per title, 20–200 characters.	42
`subtitle_limit_per_line`	Subtitles tab: Applies the character limit to each line rather than the whole title.	`true`
`subtitle_reading_characters_per_second`	Subtitles tab: Reading speed, 5–60 characters per second; sets how long a title stays.	15.0
`subtitle_gap_seconds`	Subtitles tab: Silence between two titles built one after another, 0–10 s.	0.15
`subtitle_display_layouts`	Subtitle language layouts: per-language enabled state, rectangles and typography. Saved app defaults are copied into new projects, which can edit them independently. Without saved layouts, 1–4 languages use starting arrangements; more use equal horizontal bands. Layout icons restore positions for enabled languages.	Empty
`caption_overrides`	Audience captions tab: caption words for this project — typeface, text size and time on screen. Empty values follow the app preference.	`null` (inherit)
`subtitles_follow_captions`	Subtitles tab: the titles take this project's caption look. Untick to give them their	`true`

	own look below.	
`subtitle_overrides`	Subtitles tab: the titles' own typeface, text size and time on screen, used while the box above is unticked. Empty values follow the app.	`null` (inherit)
`second_screen_width`	Second screen tab: Width of the movable audience screen panel, 320–7680 px.	1920
`second_screen_height`	Second screen tab: Height of that panel, 180–4320 px; it never opens larger than the monitor.	1080

26.3 Apply or save app defaults deliberately

Apply app defaults fills the Project settings dialog's fields; the open show changes only after accepting that dialog. **Save as app defaults** asks before copying the current project values into the app's new-project defaults. Neither operation silently updates every existing cue or activates an inherited venue. Credits and timestamped project notes remain production-specific.

27 Profiles

Example: For The Lighthouse Keeper, use **Duplicate** to copy the selected HoH profile. Edit the saved copy, then choose **Save profiles**. Apply it in HoH only when you want to change the show's look.

Open the **Profiles** button in the lower left navigation or **Edit > Profiles**. The lower group is **Profiles → Reference libraries → App preferences → Console log**. The eight mode buttons remain above it; reference libraries include venues, video projectors and lighting. Project settings remain under **Project**. The **embedded Profiles manager** works without an open project and contains Captions, Subtitles, HoH feelings, Colour palettes and Metadata. **Save profiles** commits staged changes; **Discard changes** reloads saved settings. **Back** or switching modes/libraries keeps the draft. Closing the app protects even hidden drafts with **Save**, **Discard** or **Cancel**. A failed save keeps the staged changes for retry, with an explicit error. Exported files remain on disk even if staged edits are discarded. **Back** is at the top right beside the profile tabs, matching reference-library navigation; returning keeps the draft. The footer uses one row: transfer actions on the left and Discard changes and Save profiles on the right. In narrow windows the four transfer actions are grouped into **Import / export…** without losing any operations. Footer guidance is available on hover; only a failed save adds a visible error line, which clears after a successful retry or discard.

All five profile editors use the same list width, editor area, spacing and **New…**, **Duplicate**, **Rename…**, **Delete** button positions and sizes. Category-specific actions occupy a reserved area below those buttons, so the lists and editors stay aligned when switching tabs. Button groups sit at the bottom of the sidebar without unused space below them. Colour palettes also has **Use as app default**, synchronized with **Use this palette in the colour picker**; the confirmed choice is staged until saved and does not change existing colours. Metadata rules are inside its scrolling editor, rather than shrinking its profile list. Palette and metadata name fields both use **Profile name**, with matching dimensions. Form labels and row spacing are shared, and scrollbar space is reserved so fields do not widen or narrow when a page needs scrolling.

Each tab has **New…**, **Duplicate**, **Rename…** and **Delete** profile actions. Palette and metadata edits update the selected staged profile as you edit; the palette's default choice and the metadata default for new audio are independent. The Metadata tab also offers **From a track…** and required-field rules.

On the Captions, Subtitles and HoH feelings tabs, **New…** saves the settings shown under a name. Select a saved profile to edit its copy; subsequent edits are staged back into that named profile,

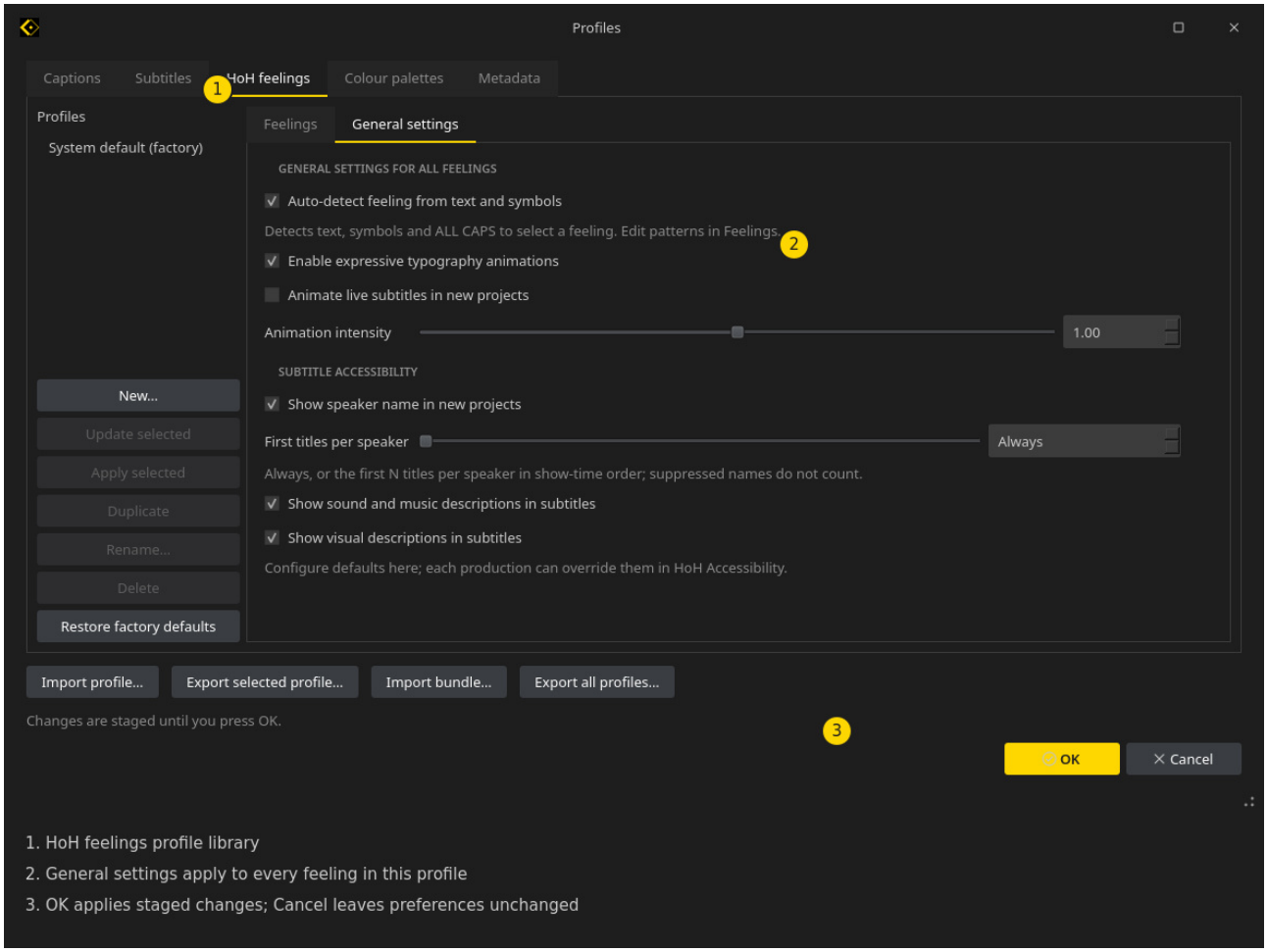
including when you select another profile. No separate **Update selected** action is needed in the embedded editors; edits are staged automatically. **Save profiles** writes those library changes without altering app defaults or shows that inherit them. **Use as app default** is a separate, confirmed action; when saved it changes the default and can affect inheriting shows, but never replaces project or title overrides. The single **Curtain Call** entry edits the current app defaults instead of a named profile; there is no second factory-template row. Duplicate, rename and delete manage saved copies; **Reset to system defaults** resets only the selected profile's values, preserving its name and ID. Applying a profile does not bind settings to it or rewrite project or title overrides. Each editor starts with one built-in **Curtain Call**, including Colour palettes and the HoH selector. HoH's Curtain Call entry uses the current editable app baseline. The redundant **Profiles** heading above the lists is omitted. Saved custom profiles, import/export and duplication remain available; their IDs and names are preserved. **Reset to system defaults** is the explicit way to reset style settings; opening the manager never resets user preferences. **Use as app default** is disabled while Curtain Call itself is selected; it applies only to custom profiles. Rename and Delete are also disabled for the built-in Curtain Call. Palette selection controls the colour picker's default; Metadata's default checkbox controls future audio imports, so these editors do not need style-specific update, app-default or show-capture buttons. Every selected profile can be reset, including custom palettes and metadata profiles. Reset is confirmed and staged until **Save profiles**; other profiles and existing projects or tracks are unchanged. The built-in Curtain Call palette can be edited, saved and duplicated; its protected name and factory reset source remain available. Metadata's Curtain Call profile starts with **YZONK** as Artist and Publisher, **Curtain Call** as Album, and the app website in Notes. Track name, code, composer, recording source, year, genre and copyright/licence remain blank; the software licence does not establish rights to imported audio. Review or edit these starting values before **Save profiles**; saving with the default checkbox selected reuses them only for empty fields on future imports. Opening the editor does not write preferences or alter existing tracks. Previously saved built-in values, including deliberately cleared fields, are preserved. The built-in name cannot be renamed or deleted. The demo uses the current factory caption, subtitle and HoH defaults, with live movement off. Its existing subtitles, translations and timing remain authored examples; changing the default reading rule does not regenerate them.

Caption profiles contain the caption appearance, including the spacing, background geometry, outline and shadow controls shared with subtitles, and display time, not monitor or window geometry. Subtitle profiles include their separate appearance, whether they follow captions, and new-project reading and language-layout defaults. HoH profiles contain accessibility options, factory and

custom feeling styles, and detection patterns; character-specific colours stay outside the profile. Venue profiles remain in their project-specific manager.

Closing App Preferences, Project settings, Subtitle language layouts, Track metadata, an Audio marker, Venue profiles or the Venue library with **X** also asks whether to apply staged edits. Applying project or track edits changes the open project; save the project separately. Venue profile changes still require their normal preview and safety checks. The explicit **Cancel** button discards staged edits without applying them.

Within **HoH feelings**, use **Feelings** for individual styles and detection patterns and **General settings** for options shared across feelings, including animation and subtitle accessibility. Both subtabs edit the same staged selected profile or app defaults; **Save profiles** commits their changes. **New…** creates a profile from the current editor settings; **Duplicate** copies the selected profile. The profile editor does not save or capture the open show. Options labelled for new projects are copied into new projects, not existing ones.



Export selected profile… and **Import profile…** transfer one profile as JSON; palette JSON from earlier releases remains supported. **Export all profiles…** writes all saved profiles to a ZIP bundle, and **Import bundle…** imports its contents together. Imports add uniquely named copies without changing current defaults. Older or unversioned imports require confirmation; newer or invalid files are rejected. The original file is never modified. Factory profiles are generated from the running app's defaults, not maintained as separate bundled JSON files. New exports, including factory-profile exports, record the running app's version and build; previously exported files retain their original version for compatibility checks. Bundles do not include recent colours, monitor settings or project data.

28 Preferences reference

Example: For new Lighthouse touring projects set operator **Rita Campos**, company **Estuary Theatre Collective**, master **90%**, cue volume **80%** and fade **3 s** in New projects. Existing Lighthouse cues keep their current levels. Choose **Show Player** as opening mode only on a show-day computer where that immediate performance view is useful.

Open **Edit > Preferences** or the **App preferences** button above Console log in the lower left navigation. Options are grouped in tabs:

Tab	Options
General	Default projects folder, mode a project opens in, recent projects to keep (1-30), reopen the last project
Saving & safety	Autosave minutes (0 = off), save before entering Show Player, ask before quitting while audio plays, copy outside audio into the project when opening, screen-lock password
Show Player	Cue font size, high contrast, collapse fired cues, script panel open, script follows fired cues, keyboard reminder, sound captions, full screen on entering
Logging	Show the log pane, start-up log detail, export the session log when a project closes
New projects	Operator, company, playback and ducking defaults, preset/custom panel size, venue-library count
Interface	Window and control options (title bar and cursor changes require restart), tips, guided tour and untested-platform feedback reminders, grouped by section

Example: in **New projects**, set operator to **Alex**, master volume to **85%** and the default fade to **4 s**. The next project starts with those values; editing the preferences later does not change that project.

The venue profile count reports reusable profiles managed in **Project > Venue profiles…**. In the new-project form, **Copy venue profile library** is on by default; turn it off to create a project without those profiles. Copied profiles get fresh project-local IDs and none is activated automatically.

For a touring operator, also choose **Open projects in: Show Player**, tick **Save before entering the Show Player** and **Go full screen**, and keep **Ask before quitting while audio is playing** on. Escape and Shift+Escape never ask for confirmation.

Restore this panel resets the current Preferences tab to factory values; **Restore all defaults** resets

the options on every tab. Other staged edits stay in place when resetting just one panel. Both actions keep the screen-lock password, recent projects, table layouts, reusable profiles and tips/tour history; use their own controls to change those. When subtitles follow captions, resetting the Subtitles panel restores that relationship and displays the current caption look. Press **OK** to save the resets, or **Cancel** to discard them. If the preferences file cannot be written, one warning explains it; the choices stay in use until the app closes.

For folder projects, autosave keeps `project.autosave.ccshow` beside the primary file. Open the backup, check it, then **Save** to restore the primary file. For single-file projects, autosave stays in the private workspace; reopen the `.ccshow`, choose recovery when offered, check it and **Save**. Autosave is not a complete history. Keep independent backups too.

28.1 Reset one preference

A small reddish counterclockwise circular arrow appears beside an option when its value differs from the factory default, including the caption and subtitle look and the defaults for new projects. Click the arrow to reset **only that value**; it disappears once the value matches the default again. Defaults mean the values shipped with the app, not the last saved choices.

For example, change **Autosave every (minutes)** to **10**, then click its arrow to restore **5** without changing your opening mode. Resets are staged like other preference edits: **OK** saves them and **Cancel** discards them. Resetting a subtitle look field gives subtitles their own look, just as editing that field does; reset **Follow the captions' look** to follow captions again.

The arrows do not erase passwords, recent projects, saved layouts, reusable profiles or tips/tour history. Use the existing password and history buttons for those actions. New-project defaults affect future projects only; existing shows and sounding audio are not changed.

28.2 Defaults for new projects

Use the **New projects** Preferences tab to set the operator, company, master volume, default fade, new-cue volume, ducking values, **subtitle rules** and **second-screen size** used at creation. Subtitle maximum characters, reading speed and gap have the same ranges as Project settings > Subtitles. Select a resolution preset or **Custom size**, then enter **Panel width** and **Panel height** in pixels. Custom dimensions are retained, not replaced with 1080p. In **Project > Project settings**, **Apply app defaults** fills the dialog fields without changing the show until you accept Project settings; **Save as app defaults** asks before updating the app preferences from the current project's values. Both commands include subtitle rules and the exact screen size as well as operator, company and playback defaults. Apply app defaults does not rebuild existing subtitles, change their saved times, or rewrite individual cue settings. It stages saved language layouts, adding missing languages and retaining unmatched project layouts; changing those layouts is blocked while a performance is active. This is a **one-time copy, not live inheritance**: editing app defaults or opening an existing project never changes that project's own values.

Example: Set subtitle maximum characters to **100**, reading speed to **18 characters/s**, gap to **0.5 s** and a **1500 × 850** custom panel in Preferences > New projects. The next Lighthouse project starts with those values, but an existing Lighthouse show keeps its saved subtitle rules and size.

28.3 Which settings have app defaults and project overrides?

Reusable **show configuration** has a factory value, an app-level default and an editable project value. The current audit covers playback defaults, subtitle rules and second-screen dimensions as one-time copies; caption/subtitle typography already has its explicit **Follow the level above** inheritance and per-title overrides. That existing typography inheritance is unchanged by this policy. Venue profiles and physical venue records use independently editable library copies rather than a live link to shared records.

Settings that belong only to the **computer/operator** stay in Preferences: window appearance, Show Player readability/layout, log viewing, autosave policy, keyboard hints, import metadata policy, reusable palettes and libraries, recent history and credentials. They are not copied into a show as artificial overrides. Authored content (script, characters, notes, credits and production revision) and project identity/selected venue are not factory templates; an active venue profile selection remains local to that show. New reusable settings must ship with Preferences and Project settings controls, save/load compatibility, factory resets, copy/transfer tests and manual coverage. This requirement is recorded in the project's **AGENTS.md** developer guide.

28.4 Every app preference

Use Edit > Preferences, then OK to apply. Cancel discards the dialog changes. The window opens large (about 980 x 780) so a whole tab is readable at once; it never opens bigger than 90% of the screen, keeps a 700 x 520 minimum, can be dragged smaller and has a maximise button in its title bar. Interface frame/cursor choices need a restart. Restoring defaults retains recent projects, table layouts and the reusable venue library; inspect the reset values before accepting. Defaults below are for a clean preferences file.

Tab / displayed name	Stored name and meaning	Default
General: Default projects folder	`projects_folder` — Suggested parent for new projects; does not move existing shows.	Documents/Curtain Call Projects
General: Reopen most recent project	`reopen_last_project` — Open the latest recent project on startup; otherwise use Welcome.	False
General: Open projects in	`opening_mode` — Script, Library, Cue Creator, Show Player, Notes, Show Score, Subtitles, HoH Accessibility or AD when opening.	library
General: Recent projects to keep	`recent_projects_limit` — 1–30; lowering it trims the list immediately.	10
Saving & safety: Autosave every (minutes)	`autosave_minutes` — 0–120; zero disables; writes recovery snapshot and primary when edited.	5
Saving & safety: Save on entering Show Player	`save_before_show_player` — Save unsaved edits at the performance-mode switch.	False
Saving & safety: Ask before closing during playback	`confirm_quit_while_playing` — Protect against closing active audio; emergency stops	True

	still require no confirmation.	
Saving & safety: Copy outside audio when opening	`copy_outside_audio_into_project` — Copy audio referenced outside the project folder into `library/` and store relative paths (an undoable edit).	True
Saving & safety: Ask for a password to unlock the locked screen	`screen_lock_use_password` — The screen lock (Ctrl+Shift+L) asks for the operator password instead of opening with Esc alone. Needs a saved password; the dialog refuses to close with the box ticked and none set.	False
Saving & safety: Set / change / remove password	`screen_lock_password_hash` — Salted PBKDF2 hash of that password, kept for this computer only and never written into a project; only the hash is stored, never the password.	(none)
Logging: Show the log pane	`show_log_pane` — Initial pane visibility, also changed with Ctrl+J.	True
Logging: Log detail at start-up	`default_log_level` — Normal or Verbose; visibility only, not which events are retained.	normal
Logging: Export log on project close	`export_log_on_close` — Save complete log to exports/logs/ when closing.	False
Show Player: Collapse fired cues	`collapse_used_cues` — Hide fired rows without removing their saved cues.	False
Show Player: High contrast	`show_player_high_contrast`	False

	— Stronger table text, grid and selection colours.	
Show Player: Cue font size (px)	`show_player_cue_font_size` — 12–28 px, independent of waveform zoom.	15
Show Player: Show script panel	`show_player_script_panel_` expanded — Initial read-only script-panel visibility.	True
Show Player: Scroll script to fired cues	`show_player_script_follow s_cues` — Follow linked cue paragraphs; manual scrolling suspends following.	True
Show Player: Show keyboard reminder in the More menu	`show_player_keyboard_hint` — Offer GO/playhead hints in the Show Player More menu.	True
Show Player: Full screen on entering	`show_player_full_screen` — Enter full screen when performance mode is selected.	False
Show Player: Show sound captions	`show_sound_captions` — Display authored captions on next/active cue cards.	True
Audience captions: Preview in the Show Player	`show_audience_caption_pre view` — Open the Show Player with its narrow audience-screen band (full column width, 72 px tall); the preview button in the right-hand rail toggles the same setting.	True
Audience captions: Caption look and placement	`audience_caption_style` — Font family, `font_size_percent` of output height, bold, colours, alignment and vertical position.	Sans-serif, 5%, bold, white on black, centred, bottom
Audience captions: Captions stay up at least	`caption_typography` — its `display_seconds` is the	4 s

	shortest time a caption stays on the board; longer words still get the time they need.	
Audience captions: board font, size, colours, placement	`audience_caption_style` — The sound-caption board on the audience screen, sampled live in the same tab.	System font, 5%, white on black
Subtitles: Subtitles use the same look as the audience captions	`subtitles_follow_captions` — Ticked, titles take the caption board's typeface and size; unticked, the titles have their own look.	True
Subtitles: typeface, text size, time on screen	`subtitle_typography` — This computer's own subtitle look, including `font_size_percent`; project/title/language overrides retain inheritance.	System font, 5%, 4 s
Interface: App-drawn dark title bar	`use_custom_window_frame` — Restart required; off uses operating-system frame.	True
Interface: Native OS file and folder dialogs	`use_native_file_dialogs` — Applies immediately after saving preferences; off uses themed pickers.	True
Interface: Hide the top bar in full screen	`hide_menu_bar_in_full_screen` — Hide the top menu and custom title bar in full screen; tap Alt to reveal them temporarily. The activity bar stays visible.	False
Interface: Pointing-hand cursor	`use_pointing_hand_cursor` — Restart required; clickable enabled controls use a hand.	True
Interface: Show a tip of the day	`show_tip_of_the_day` — Offer one unread tip after start-	True

	up (not when opening straight into the Show Player); Help > Tip of the day always works.	
Interface: Guided tour	`show_guided_tour` — Point at the parts of the window on first use and after an update that adds stops; Show the tour again forgets which stops this computer has seen.	True
Interface: Seen tour stops	`seen_tour_step_ids` — Ids of the tour stops this computer has already been shown, so none repeats and a new version teaches only what it added; "Show the tour again" clears the list.	`[]`
Interface: Read tips	`viewed_tip_ids` — Ids of the tips this computer has already been shown, so none repeats; "Show every tip again" clears the list.	`[]`
Metadata: required on import	`require_metadata_on_import` — Ask for the required metadata when audio is imported, and open the metadata editor for incomplete files.	True
Metadata: required fields	`required_metadata_fields` — Track attributes a recording must carry (`name`, `copyright` by default); they are marked * in the editor and flag the row in the library.	`["name", "copyright"]`
Metadata: profiles	`metadata_profiles` — Reusable named metadata values (`id`, `name`, `values`),	`[]`

	created in Edit > Profiles > Metadata. App-level, never stored in a project.	
Metadata: default profile	`default_metadata_profile_id` — Profile filled into every newly imported track (empty fields only); unknown ids are repaired to none.	`""`
Colour palettes	`colour_profiles` — Named lists of up to 48 distinct hex shades, managed in Edit > Profiles > Colour palettes.	`[]`
Style profiles	`style_profiles` — Named caption, subtitle and HoH feeling settings managed under Edit > Profiles.	`[]`
Colour picker default palette	`default_colour_profile_id` — Personal palette offered first; empty or missing ids select the factory palette.	`""`
Colour picker history	`recent_colours` — The last 12 distinct committed colours, newest first, shared by all colour choosers.	`[]`
Automatic / table headers: Remembered table layouts	`table_layouts` — Per-table column widths/order/hidden state; not an editable text option.	Empty
Automatic / split panels	`splitter_layouts` — Remembered proportions of resizable panels.	Empty
Automatic / dialogs	`dialog_sizes` — Remembered dimensions of resizable dialogs.	Empty
Automatic / File > Open recent: Recent project history	`recent_project_files` — Newest-first remembered	Empty

	project paths; not the project contents.	
New projects: Defaults group	`new_project_defaults` — One-time creation defaults detailed below, not live inheritance.	See table below

28.5 Every New projects default

Set **App preferences > New projects > Default original language** to a chosen language or **System language (automatic)**. Automatic uses the OS language and region (for example `pt-PT`); a language-neutral C locale uses en. New projects copy the resolved language into their project fallback and initial subtitle/AD originals, so moving a project to another computer does not change it. For an unset original in an older project, priority is the project's saved fallback, then this app preference, then the OS. Explicit subtitle and AD originals always win and remain independent. The fallback is saved as the original on the first editing/translation action, with Undo support; opening a project alone does not rewrite it. A conflict with an existing translation requires choosing the actual original language instead. This is a default, not automatic text-language detection.

Stored name	Meaning	Clean default
`operator`	Suggested operator credit.	Empty
`company`	Suggested company/group credit.	Empty
`original_language`	Default original language; empty follows the OS. Also fallback for unset originals in old projects.	Empty (system language)
`master_volume_percent`	Initial master percent, 0–100.	100.0
`default_fade_seconds`	Initial default fade, 0–600 s.	3.0
`default_cue_volume_percent`	Initial new-start level, 0–150%.	100.0
`default_duck_gain`	Initial inherited target multiplier, 0.05–1.	0.5
`default_duck_attack_seconds`	Initial inherited attack, 0–60 s.	0.3
`default_duck_release_seconds`	Initial inherited release, 0–60 s.	1.5
`second_screen_resolution`	Preset or Custom size for the second screen panel copied into new projects.	1080p
`second_screen_width`	Panel width in pixels for Custom size, 320–7680; presets supply exact dimensions.	1920
`second_screen_height`	Panel height in pixels for Custom size, 180–4320; presets supply	1080

	exact dimensions.	
`subtitle_maximum_characters`	Initial per-line or legacy per-title limit, 20–200 characters.	42
`subtitle_limit_per_line`	Initial subtitle character limit applies per line rather than per title.	`true`
`subtitle_reading_characters_per_second`	Initial subtitle reading speed, 5–60 characters/s.	15.0
`subtitle_gap_seconds`	Initial gap between built subtitle titles, 0–10 s.	0.15
`venue_profile_library`	Reusable profile list managed in Venue profiles; count displayed in New projects.	Empty

28.6 Keep preferences separate from the production

The preferences JSON is per user/computer, or under portable-data/ in portable mode. Hand-edited damaged content loads with safe defaults instead of creating a new project silently. Invalid opening modes and recent limits are repaired. A save failure shows a warning; in-memory choices can still apply for this run, but they are not guaranteed to persist after closing. Resolve permissions or disk space rather than repeatedly assuming OK means the file was written.

Table layouts are saved by the header controls, not by changing cues. Their retention when restoring defaults is intentional: losing a carefully arranged performance table should not be a side effect of resetting an autosave preference.

29 Menus and toolbar reference

Example: In The Lighthouse Keeper, **Project > Import script** enters Script and starts its import flow. **Project > What if…** opens the silent snapshot tool; **Cue Creator > Test fire** auditions real audio instead. Check the distinction before clicking a similarly placed toolbar icon.

29.1 Every main menu item

Ellipses indicate another dialog or choice. Project-dependent commands are disabled on Welcome; Help and Preferences are available without a production.

Menu / item	What it does
File > New project…	Confirm unsaved handling, choose name/ location, create a single `.ccshow` file
File > Open project…	Select a .ccshow file using the native/themed preference
File > Open recent	Submenu of remembered production files
File > Save	Persist the open project
File > Save as…	Copy current project and pending edits to a new independent folder and switch to it
File > Close project	Confirm unsaved handling and return to Welcome
File > Quit	Close the app; configured audio/unsaved safety prompts may apply
Edit > Undo	Restore previous project content
Edit > Redo	Reapply an undone project edit
Edit > Preferences…	Change user/computer settings
Edit > Track metadata…	Open the metadata editor for the selected library track (required fields enforced)
Edit > Profiles…	Manage caption, subtitle, HoH feeling, colour and metadata profiles; available without a project
Edit > Translation models…	Choose, install or uninstall local language pairs; available without a project
Edit > AD speech models…	Review, check, explicitly install or uninstall local narration voices; no automatic downloads
Edit > Check for model updates…	Compare installed local models with their trusted catalogues; updates need explicit approval

View > Script	Open mode 1
View > Library	Open mode 2
View > Cue Creator	Open mode 3
View > Show Player	Open mode 4; preflight and performance preferences apply
View > Rehearsal Notes	Open mode 5 without stopping audio
View > Show Score	Open mode 6: the whole show drawn as one silent timeline
View > Subtitles	Open mode 7: turn the script's dialogue into a timed subtitle track
View > HoH Accessibility	Configure project character colours and feeling profiles (Ctrl+8)
View > AD — Audio Description	Prepare character texts/translations and generate, import or locally audition narration (Ctrl+9); no live AD routing
View > Show / hide output	Toggle the session-log pane
View > Audience screen preview in the Show Player	Show or hide the narrow audience-screen band (full column width); the same setting as the preview button in the Show Player's right-hand rail
View > Lock screen	Cover the window with the screen lock (Ctrl+Shift+L); the show keeps playing
Help > Tip of the day	Open the tip dialog at any time; Next tip browses the list
Help > Guided tour	Dim the window and point at one area at a time, from the first stop; Next/Back move, Skip for now keeps it for later, Close forever switches it off
Project > Project settings…	Edit production-specific details, credits, notes, defaults, subtitle rules and the second screen panel size
Project > Project health…	Review preflight issues and current audio decode results, and navigate to their editors

Project > Compare show revisions…	Compare current or selected project data read-only, side by side
Project > Import script…	Enter Script and begin import/revision review
Project > Rehearsal run timeline…	Open non-modal silent run review/comparison
Project > What if…	Open silent detached cue simulation
Project > Venue profiles…	Stage room trim, notes, checklist and active venue changes
Libraries > Venues…	Create, edit, duplicate and delete local physical venue records without changing existing project copies
Libraries > Video projectors…	Edit local video-projector specifications and source checks; no video mapping or device control
Libraries > Lighting fixtures…	Edit local stage-lighting specifications and source checks; no DMX output or project assignment
Project > Copy outside audio into project	Copy audio referenced outside the folder into `library/` now and report the result
Project > Clean Up Project…	Select cleanup categories, review per-category and total sizes, then permanently delete selected files and report before/after sizes after explicit confirmation
Project > Export > Web version	Produce lossy compatibility audio-show.js and audio files
Project > Export > Cue sheet (CSV)	Write spreadsheet-readable cue sheet
Project > Export > Printable cue sheet (PDF)	Write landscape operator PDF, then offer open / folder / email
Project > Export > Show score (PDF)	Write the A3 landscape show score, then offer open / folder / email
Project > Export > Audience guide (PDF or Markdown)…	Export an inclusive subtitle colour and movement guide for the audience
Project > Export > Subtitles (WebVTT + SRT)	Write `exports/subtitles.vtt` and `exports/subtitles.srt`, then offer open / folder / email

Project > Export > Session log	Export complete unfiltered session evidence
Project > Export > Export all	Generate all applicable formats with an output/checksum manifest
Project > Export > Export all as zip archive	ZIP only the current operation's inventoried exports
Project > Export > Clean show copy for sharing…	Export a minimal native `.ccshow` without cleaning the original
Project > Export > Zip and send by email…	Generate ZIP and open an email draft with it (attached where possible)
Project > Export > Open exports folder	Open the production's outputs in the desktop file manager
Help > User guide	Open/reuse this offline manual, F1
Help > Send feedback or ideas…	Review and submit a report securely to YZONK with consent, private email and a tracking ID
Help > About Curtain Call	Show release, author and application information

29.2 Script toolbar and panels

Action	Effect
Import	Read a supported script; revision review protects existing anchors
Element selector	Change current paragraph kind
Delete line	Remove current paragraph as an undoable edit
Add cue here	Create a cue linked to the cursor paragraph
Template	Confirm replacement when needed and create starter theatre script
Export ODT / Export Markdown	Save styled/text script deliverables
Script PDF	Export numbered script pages with cue markers
Print layout	Open themed paper preview before printing/saving
Find box	Find script text; not a whole-manual or Library search
Outline click / double-click	Navigate heading; collapse/expand descendants

29.3 Library toolbar and detail controls

Action	Effect
Add audio	Select files and copy/reference them in library/
Relink	Replace selected missing/wrong file while preserving track identity
Relink missing	Scan chosen folder recursively for exact unique filename matches
Remove	Remove selected library track; review cue references
Add to cue list	Create a Start using selected track
Use file info	Fill empty project metadata from audio tags
Play preview / stop preview	Audition the whole file, not cue semantics
Library waveform: Zoom in / Zoom out / Fit	Magnify the time axis up to 16x or return to the whole recording
Library and Cue Creator waveform: Ctrl+mouse wheel	Zoom around the pointer without changing playback or the show
Library waveform: horizontal scrollbar / click	Scroll the magnified recording; click to start preview at that time

29.4 Cue Creator toolbar and script panel

Action	Effect
Add cue > Start / Fade / Stop / Stop all / Group / Delay / Memo / Placeholder	Add chosen instruction with relevant project defaults
Insert recipe…	Choose slots and preview bundle before one undoable insertion
Save as recipe…	Store selected self-contained pattern in app recipes.json
Move up / Move down	Reorder selected cue without firing
Duplicate	Clone selected instruction; inspect links afterward
Renumber	Set sequential cue numbers in list order
Remove	Delete selected cues, undoable
What if…	Detached silent explanation of sequence logic
Test fire	Real audition without pre-wait or automatic followers
Script Import	Start script import/revision flow
Script Link / Unlink	Assign/remove selected cue's paragraph reference
Script Add cue here	Insert linked new cue after selected cue
Script search	Filter visible script lines without editing them
Ducking All / None / Copy to all cues / Preview	Select targets, copy settings or audition the reduction
Waveform fade/duck handles	Edit cue envelope seconds/level, not the audio file

29.5 Show Player controls

Action	Effect
GO (Space)	Fire next armed cue with programmed waits and continuation
Pause (Ctrl+Space)	Freeze sounding cues, fades and waits in place; press again to carry on
Stop all (Esc)	Cancel waits and fade all using project default fade
Panic (Shift+Esc)	Cancel waits and cut all immediately
Up / Down or cue selection	Position the next cue without firing
More > Reset show timer	Reset timer; next GO starts it again, not audio rewind
Subtitles (Show Player header)	Shows enabled audience subtitle languages; click and confirm to change one or more before the first GO or Subtitles GO
Full screen	Toggle performance display size, also F11
Record / Stop	Start named event recording / save separate run file
Drop pin	Optional text bookmark in session log; Cancel creates nothing
Drop note	Persistent correction under a dated rehearsal
Lock	Cover the screen while the show continues, also Ctrl+Shift+L
Audience screen (right-hand rail)	Two buttons: show or hide the audience-screen preview band (remembered as a preference), and open or close the audience screen itself
More > Reset venue trim	Confirm return to 0 dB; requires idle output/waits
Right-edge Script icon	Expand/collapse read-only script panel
Right-edge Follow cues icon	Switch script following on or off (also after manual scrolling)
Right-edge Collapse fired cues icon	Toggle hidden fired rows without deleting cues
Follow cues	Resume script following after manual browsing (the button is in the right-hand rail)
Active strip Mute / Fade / Stop	Modify sounding voice, not the saved cue list
Active strip live trim / Reset	Temporary gain multiplier / return to programmed 100% multiplier

29.6 Rehearsal Notes toolbar

Action	Effect
New rehearsal	Create a dated project-level session with an optional iteration label and baseline snapshot
Add note	Add correction to selected rehearsal
Show Player (Ctrl+4)	Return to performance without stopping audio
Run timeline	Review/compare recorded event files silently
Change digest…	Compare note, cue and script facts between rehearsal snapshots
Update snapshot…	Confirm an undoable replacement with today's baseline
Export report	Write selected rehearsal as complete UTF-8 text
Export PDF	Write selected rehearsal as paginated A4 report
Send by email	Export report and open an email draft with it (attached where possible)
Go to script line / cue / library track	Exact silent navigation to stable reference

29.7 Shared table, output and Help controls

Resize column edges and drag titles to reorder. Right-click a table header to show/hide individual columns, **Fit columns** to content, or **Reset columns**. The current layout is remembered per table, including the Cue Creator's **Captions** table. Table arrangement is an app preference, not a cue order change; dragging Cue Creator rows is a different operation.

Output uses Normal/Verbose, severity and search filters, Jump to cue, Export log and Clear. Help uses chapter index/picker, search Previous/Next, Maximise, Print and Export, with whole/chapter/selection scope. See their dedicated chapters for the distinction between visible content and complete export data.

30 Glossary

Example: In The Lighthouse Keeper, **AMB** is a track code; **1** is a cue number; the stored cue ID keeps its references stable; **Pre-show** is the linked script heading; **0.25** is the announcement ducking multiplier; **−6 dB** would be a separate venue trim, not a cue-volume edit.

30.1 A

- **Activity bar** – The narrow vertical strip on the left edge of the app window. Each icon switches to one of the nine modes, in this order: Script, Library, Cue Creator, Rehearsal Notes, Show Score, Subtitles, HoH Accessibility, AD and, last, Show Player. Clicking an icon never stops audio that is already playing.
- **Anchor / script link** – A stable reference from a cue or rehearsal note to a specific paragraph in the script. It provides visual context and navigation for the operator — for example, showing which action line should trigger a cue — but it does not listen for actors or fire the cue automatically. When you edit the paragraph text normally, the anchor survives. When you re-import a revised script, the revision assistant tries to find the same paragraph in the new version and asks you to confirm or correct any uncertain matches.
- **Armed / disarmed** – An armed cue is eligible to be fired by GO, a hotkey, a group instruction or automatic continuation. A disarmed cue stays in the list as documentation but is silently skipped by all of those paths. Disarming is useful for an alternative effect you want to keep visible but not accidentally call. It is not a live mute of a recording that is already playing.
- **Attack (ducking)** – The time in seconds it takes for a ducking source to lower its target cues from their normal level to the reduced duck gain. A short attack (e.g. 0.2 s) dips the background quickly; a longer attack (e.g. 1 s) makes the dip feel gradual. See also *Ducking*, *Release*.
- **Autosave** – A periodic recovery snapshot for unsaved edits, not a continuous history. Folder projects write `project.autosave.ccshow` beside the primary file. Single-file projects write into the private workspace; reopen the `.ccshow` and choose recovery. Explicit Save updates the archive.

30.2 C

- **Caption** – See *Sound caption*.
- **Caption look** – The typeface, text size and shortest time on screen used for captions and subtitles. Set at three levels: the app preference, the project, and one title; each level inherits what it does not change.
- **.ccrun.json** – The file extension for a flight recorder run file. Each recorded run is saved as a separate `.ccrun.json` file in the project's `runs/` folder. It is a plain JSON text file containing timestamped event records (GO presses, cue fires, pins, failures). It is not part of the main project file and is not affected by project Undo. Copy the `runs/` folder when handing off a project if rehearsal evidence matters.
- **.ccshow** – A Curtain Call show file. A single-file project is a ZIP containing project JSON and media; save and copy the file to move it. A folder project's `project.ccshow` is JSON and must travel with the entire project folder. A file association can open either in Curtain Call.
- **Change digest** – A read-only factual comparison between two rehearsal snapshots. It lists exactly which cues changed (and which fields), which script paragraphs moved or were edited, and which rehearsal notes changed status. It does not judge whether the changes are artistically correct; it only reports what the data says. Produced from **Rehearsal Notes > Change digest...**.
- **Codec** – The software that decodes an audio file format so the computer can play it. MP3 and WAV are tested and always available. FLAC, OGG and M4A depend on the FFmpeg build that ships with your operating system or PySide6. If a file plays in a media player but not in Curtain Call, the output log will show a format error; convert the file to WAV with an audio editor and relink it.
- **Collapse fired cues** – A Show Player display option that hides cue rows that have already been fired, making the list shorter and easier to read during a long show. Collapsing does not delete or disarm the cues; they remain in the project and can be uncollapsed at any time. The playhead still moves forward correctly through collapsed rows.
- **Continuation** – The setting on a cue that determines whether and when the next armed cue is automatically scheduled after this one runs. The three options are: **none** (no automatic follower), **auto-continue** (schedule the follower as soon as this cue's action starts) and **auto-follow** (on a Start cue, schedule the follower when the recording reaches its natural end; on

other cue types, behave like auto-continue). See also *Auto-continue*, *Auto-follow*, *Post-wait*.

- **CSV** – Comma-Separated Values. A plain text file format where each row is a line and each column is separated by a comma. Curtain Call exports the cue sheet and run comparison as CSV files that can be opened in any spreadsheet application (LibreOffice Calc, Excel, Google Sheets, etc.). CSV files contain no formatting or images; they are useful for further processing or printing in a custom layout.
- **Cue** – An ordered instruction in the cue list. Each cue has a type (Start, Fade, Stop, Stop All, Group, Delay, Memo or Placeholder), a number, a name and various settings. When GO is pressed, the next armed cue executes its instruction. Cues are not audio files; they are instructions that reference audio files stored in the Library.
- **Cue in note** – A short operator instruction written in the cue's Notes tab that describes *when* to press GO for this cue — for example, "After the door slams" or "On the word 'now'". It appears on the next-cue card in Show Player. It is authored by you; the app never fills it in automatically.
- **Cue number** – A free-text label you assign to a cue, such as `1`, `4A` or `Pre-show`. It is for human reference only. The actual order of cues in the list determines what fires next, not the number. Two cues can share a number without causing an error, though that is confusing in practice.
- **Cue out note** – A short operator instruction describing when or how the cue ends — for example, "Fade when lights go to half". Like the cue-in note, it is authored by you and does not schedule any automatic action.
- **Cue sheet** – A printable or exportable document listing every cue in order with its number, name, type, settings, sound caption, cue-in note and linked script context. Useful as paper backup for the operator or as a handoff document for a substitute. Exported as CSV (spreadsheet) or PDF from **Project > Export**.
- **Cue type** – The kind of instruction a cue executes. The eight types are: **Start** (plays a recording), **Fade** (changes the level of a playing cue), **Stop** (fades a playing cue to silence and removes it), **Stop All** (stops everything), **Group** (fires all armed start cues that share a group name), **Delay** (waits a set number of seconds before scheduling the next cue), **Memo** (displays operator text without playing audio) and **Placeholder** (a numbered slot whose instruction is not written yet; it plays nothing and the preflight lists it).

30.3 D

- **Delay cue** – A cue whose only job is to wait. When fired, it counts down its configured delay in seconds, then schedules the next armed cue. Useful for inserting a timed gap in an automatic chain without attaching that gap to a specific recording. Its pre-wait runs before the delay, and its post-wait runs after.
- **Device note** – An optional text field in a venue profile where you record the name of the preferred audio output device for that room — for example, "USB Audio Interface (Focusrite)". Curtain Call can check whether a device with that name is currently available (**Check available device names**), but it does not automatically route audio to it. Actual output routing must be configured in the operating system's audio settings.
- **DOCX** – The file format used by Microsoft Word and compatible applications (LibreOffice Writer, Google Docs, etc.) for word-processed documents. Curtain Call can import `.docx` files as scripts. Paragraph styles in the document (Heading, Scene, Action, etc.) are used to assign script elements automatically. Plain text without styles will import but may need manual element corrections.
- **Drop note** – A persistent correction attached to a named rehearsal. Created from Show Player during a run, it pre-fills the currently selected cue, track and script line as references. It has a title, a details field and a status (Open, In Progress, Resolved). Unlike a drop pin, it survives closing the app and is tracked through later rehearsals. See also *Drop pin*.
- **Drop pin** – A timestamped bookmark added to the session log during a run. It can include optional free text. It is useful for marking a moment you want to review later ("Paused for lighting reset") without creating a formal correction. It is not saved to the project file; export the session log to preserve it. See also *Drop note*.
- **Duck gain** – A multiplier between 0.05 and 1.0 that sets how far a ducking source lowers its targets. A value of 0.25 means the target is reduced to one quarter of its current programmed level. It is not a decibel value. The perceived result also depends on the target's own volume, master level and fade state.
- **Duck preview** – A short audio audition in Cue Creator that lets you hear how ducking will sound before committing to it. It plays the target cue alone for 2 seconds, then brings in the source cue for 8 seconds so you can hear the reduction, then fades both out. It plays real audio and suppresses automatic followers. It is not the silent What-if simulator.

- **Ducking** – A feature where a foreground cue automatically lowers the level of one or more background cues while it is playing. For example, a phone announcement can duck the house ambience so the announcement is clearly audible. Ducking is configured on the foreground (source) cue, not on the background (target) cue. When the source stops, the targets return to their normal levels over the release time. Ducking is triggered by active playback, not by microphone input or speech detection.
- **Duplicate (cue)** – Creates an exact copy of the selected cue and inserts it immediately after the original. The duplicate gets a new stable ID and blank hotkey but inherits all other settings. Useful for reusing a complex cue setup as a starting point. After duplicating, inspect target references and ducking settings to make sure they still point to the right cues.

30.4 E

- **Element (script)** – The type of a paragraph in the script editor. The seven elements are: **Heading** (play title or major section), **Scene** (scene title or location), **Action** (stage business and sound triggers), **Character** (speaker name), **Parenthetical** (delivery instruction inside a speech), **Dialogue** (spoken text) and **Transition** (end or change of scene). The element controls the paragraph's layout and its role in the script structure; it is not just a font choice.
- **End of show** – The state Show Player enters after the last armed cue has been fired. The next-cue card reads "End of show" and further GO presses do nothing (the log records "end of show"). To replay a cue, click its row or press Up to select it, then press GO. Opening another project always starts at its first cue.
- **EndOfMedia** – The event that fires when a finite audio recording reaches its natural end. This is what triggers auto-follow continuation on a Start cue. An infinite loop (loop count = 0) never produces an EndOfMedia event, so auto-follow on an endless loop will never schedule a follower. Manually stopping or fading a cue to zero also does not produce EndOfMedia.
- **Export** – A frozen deliverable file (PDF, CSV, Markdown, MHTML, plain text) written into the sibling `<project name> -- exports/` folder. Exports are snapshots: they do not update when you later edit the project. Re-export after making changes. An exports ZIP is a communication bundle, not a substitute for a complete project backup.

30.5 F

- **Fade** – A gradual change in a cue's playback level over a set duration. A fade in raises the level from silence when a Start cue begins. A fade out lowers the level toward a target (which may be zero). The shape of the change is controlled by the *fade curve*. A cue faded to zero is still active and still applies ducking; use a Stop cue to fully remove it from playback.
- **Fade curve** – The mathematical shape that describes how a level change progresses over time. The five curves are: **Linear** (steady, equal progress throughout), **Smooth / S-curve** (gentle start and end, faster in the middle — natural-sounding for most music), **Fast start** (most of the change happens early, then eases off), **Slow start** (little change at first, then accelerates), **Exponential** (rapid initial change with a long, slow approach to the target). The same curve can describe a fade in or a fade out.
- **FFmpeg** – An open-source multimedia library used by Qt to decode audio files. It handles MP3, WAV, FLAC, OGG, M4A and many other formats. The specific formats available depend on which FFmpeg build is included with your PySide6 installation. If a file fails to play, the output log will show an FFmpeg error message.
- **Flight recorder** – The feature that records a timestamped log of every operator action during a run: GO presses, cue fires, Stop All, Panic, drop pins, drop notes and playback failures. It does not record audio. Each run is saved as a separate `.ccrun.json` file in the project's `runs/` folder. Runs can be reviewed on a timeline and compared against each other to analyse pacing changes.

30.6 G

- **Gain staging** – The practice of setting appropriate signal levels at each stage of an audio chain (recording, software volume, amplifier, speakers) so that the signal is loud enough to be clear but not so loud that it distorts. Curtain Call controls software-level gain (cue volume, master, venue trim, live trim) but has no control over amplifier gain, speaker sensitivity or room acoustics. Proper gain staging requires listening and adjusting the physical equipment.
- **GO** – The main operator action in Show Player. Pressing Space or the GO button fires the next armed cue with all its programmed settings: pre-wait, fade in, continuation, etc. A double press during a pre-wait countdown is ignored (the log says "extra GO ignored") — it never starts a second copy of the cue. Use Stop All or Panic to cancel a cue that is waiting.
- **Group cue** – A cue type that fires all armed Start cues whose group name matches the group target name. All matching members start simultaneously (subject to their own pre-waits). Useful for triggering a set of related sounds — for example, all the sound effects for a scene opening — with a single GO.

30.7 H

- **Headroom** – The margin between the current output level and the maximum the system can handle before distortion or clipping. Positive venue trim (above 0 dB) requests more output than the neutral level and displays a headroom warning because Qt's audio output cannot deliver gain above 100% and may saturate. Headroom is not automatically managed by the app; verify levels with the house technician.
- **High contrast** – A Show Player display option that increases the contrast of the cue table — stronger text, grid lines and selection highlight — for operators working in bright environments or with visual accessibility needs. Enabled in **Edit > Preferences > Show Player > High contrast**.
- **Hotkey** – A single keyboard key assigned to a cue that fires it directly from Show Player, without moving the playhead. For example, pressing p fires the Phones Announcement cue in the demo. Hotkeys cannot use keys reserved for app functions (Space, Escape, arrow keys, F1, Ctrl+1–6, etc.). If two armed cues share a hotkey, only the first one responds. Preflight lists all hotkey conflicts.

30.8 I

- **Iteration / run label** – Optional free text on a project-level rehearsal session, such as "Run 2" or "Dress". It is a label, not a separate object or a parent for rehearsals. Each rehearsal session has its own snapshot for change-digest comparisons; multiple sessions may share a label.

30.9 J

- **JSON** – JavaScript Object Notation. A plain text format for structured data. Folder projects use `project.ccshow` JSON; single-file projects contain JSON and media in a ZIP. Preferences, recipes and run files also use JSON. Keep a backup before editing project data by hand.

30.10 L

- **Library** – The collection of audio media files imported into a project. Each entry (called a *track*) stores the file path, a short code, metadata (artist, composer, album, etc.) and notes. The Library does not store playback settings; those belong to individual cues. One library track can be used by many cues with different volumes, fades and loops.
- **Live trim** – A temporary gain adjustment applied to a currently playing cue from the Now Playing strip in Show Player. 100% means the cue's programmed volume; values above or below scale it up or down. Live trim is not saved to the project and does not affect the cue's stored settings. Reset returns it to 100%.
- **Loop count** – How many times a recording plays in total. 1 means it plays once. 3 means three complete passes. 0 means it loops indefinitely until stopped. Note that 3 means three total plays, not three *extra* repeats after the first.
- **Loop crossfade** – An overlap between the end of one loop pass and the beginning of the next, using complementary gain curves so the transition is smooth rather than a hard cut. The overlap duration is limited by the engine to half the recording's total length. With a loop count of 1 (plays once), crossfade has no effect because there is no second pass to overlap with.

30.11 M

- **Markdown** – A lightweight plain-text format that uses simple symbols to indicate formatting: `**bold**`, `# Heading`, `- list item`, etc. Curtain Call can import `.md` files as scripts and export the script, user guide and rehearsal reports as Markdown. It is useful for version control, sharing editable text, or opening in a text editor. The visual layout of a Markdown file differs from a formatted stage script.
- **Master volume** – A project-level multiplier (0–100%) applied to all playback output. It is set in Project Settings and travels with the project. It is separate from individual cue volumes, venue trim and live trim. Reducing the master affects everything simultaneously without changing any cue's saved settings.
- **Memo cue** – A cue type that displays operator text in Show Player without playing any audio. Useful for reminders, scene change instructions or notes to a substitute operator. Memo rows appear with a subtle diagonal hatch pattern in the cue list so they are visually distinct from audio cues.
- **MIME type / file association** – The operating system's way of knowing which application should open a file with a given extension. On Linux, `tools/install_linux_integration.sh` registers the `.ccshow` extension so that double-clicking a project file opens it in Curtain Call. This is a desktop convenience; it does not affect how the app reads or writes project files.
- **Mode** – One of the nine main views of the app, selected from the activity bar. In bar order: Script (`Ctrl+1`), Library (`Ctrl+2`), Cue Creator (`Ctrl+3`), Rehearsal Notes (`Ctrl+5`), Show Score (`Ctrl+6`), Subtitles (`Ctrl+7`), HoH Accessibility (`Ctrl+8`), AD (`Ctrl+9`) and Show Player (`Ctrl+4`, last in the bar). Switching modes never stops audio.
- **Mute** – Suppresses the audio output of a playing cue without stopping it. The cue remains active, continues looping and continues applying ducking to its targets. To fully remove a cue from playback, use Stop.

30.12 N

- **Next-cue card** – The prominent panel at the top of Show Player that shows the full details of the cue that will fire on the next GO: its number, name, sound caption, cue-in note, type, key settings (volume, fade, loops, waits) and the linked script paragraph. Reading this card before pressing GO is the primary safety check during a performance.

- **Now Playing strip** – A row in Show Player's lower panel that represents a currently active (sounding or fading) cue. Each strip shows the cue name, elapsed time, loop pass, state word, a mini waveform, the current level and controls for Mute, Fade, Stop and live trim. Newer voices appear above older ones.

30.13 O

- **OCR (Optical Character Recognition)** – Software that reads text from a scanned image. Curtain Call's optional `pdf-import` feature can only extract *selectable* text from PDFs — text that was typed and saved digitally. It cannot read text from a scanned page (a photograph of a printed script). If your PDF was created by scanning a physical document, you need to run OCR software on it first (such as Tesseract or Adobe Acrobat) before importing it into Curtain Call.
- **ODT** – OpenDocument Text. The file format used by LibreOffice Writer and other open-source word processors. Curtain Call can import `.odt` files as scripts and export the script as ODT. ODT files preserve paragraph styles (Heading, Scene, Action, etc.) which Curtain Call uses to assign script elements automatically. ODT is generally the best format for round-tripping a script between a word processor and Curtain Call.
- **Outline (script)** – The panel on the left side of the Script mode that lists all Heading and Scene paragraphs in the script as a collapsible tree. Clicking an entry scrolls the editor to that section. Double-clicking a heading collapses or expands its descendants in the outline without deleting any paragraphs. Useful for navigating long scripts quickly.
- **Output routing** – The operating system's configuration of which audio device receives sound from an application. Curtain Call plays audio through whatever device Qt selects (usually the system default) or the device named in the active venue profile's device note. The app cannot change the routing itself; use the operating system's sound settings or audio interface software to direct output to the correct device (house speakers, headphones, etc.).

30.14 P

- **Pacing change** – In the run timeline comparison, the difference in the time gap between corresponding cues across two runs. A positive value means the gap was longer in the second run (slower pacing); a negative value means it was shorter (faster pacing). It measures operator timing and actor pace, not audio accuracy.
- **Panic** – An emergency action (`Shift+Esc`) that immediately cuts all audio without any fade and cancels all pending pre-waits, post-waits and delay timers. Unlike Stop All, which uses the project's default fade duration, Panic is instantaneous. Neither Panic nor Stop All can be undone; after either, assess the situation and deliberately select the correct next cue.
- **pdf-import** – An optional Python package (`pypdf`) that adds the ability to import scripts from PDF files with selectable text. Install it with `pip install '.[pdf-import]'`. Without it, the PDF import option is greyed out. It does not add OCR capability; scanned PDFs still need OCR elsewhere first.
- **Playhead** – The marker in the Show Player cue list that indicates which cue will fire on the next GO. Moving it with the Up/Down arrow keys does not fire any cue. The playhead is tied to the cue's stable identity, not its row number, so reordering or renumbering cues does not move it unexpectedly.
- **Portable mode** – A way of running Curtain Call from a USB stick or shared drive without writing any settings to the host computer's user account. When a file named `portable.txt` exists next to the app folder, all preferences and user data are stored in `portable-data/` next to the app instead of in the user's home directory. Set up with `tools/setup.sh --portable`.
- **Post-wait** – A delay applied *after* continuation is scheduled but *before* the follower cue is fired. It is not a pause between every manual GO press; it only applies when a cue has auto-continue or auto-follow set. For example, a post-wait of 2 s on a fade cue means the next cue fires 2 seconds after the fade begins, not 2 seconds after it finishes.
- **Pre-wait** – A delay between the moment a cue is fired (GO pressed or automatic continuation triggered) and the moment its action actually executes. During the countdown, the cue is shown as "Pre-wait" in Show Player. A second GO press during the countdown is ignored. Use Stop All or Panic to cancel a waiting cue.
- **Preflight** – A set of consistency checks run automatically when you enter Show Player. It checks for missing audio files, Start cues with no track assigned, Fade/Stop cues with no valid target,

hotkey conflicts and empty selected-ducking lists. A clean preflight result means the project data is internally consistent; it does not verify that speakers are connected, amplifiers are on or that levels are appropriate for the room.

- **Print layout** – A preview mode in the Script toolbar that shows how the script will look when printed or exported as PDF, including page breaks, cue markers in the margin, scene numbers and page numbers. Use it to check pagination before distributing a printed script to the director or cast.
- **Project** – A `.ccshow` file for one production. Older folder projects remain supported; copy the whole folder, not only its `project.ccshow`. New generated deliverables go in a separate sibling exports folder.
- **project.autosave.ccshow** – The latest recovery snapshot. In a folder project, open it in its original folder and save after verification. In a single-file project, it lives in the private recovery workspace; reopen the archive and choose recovery. It is not a full backup history.
- **placeholder cue** – A cue that reserves a numbered slot in the list before its real instruction exists. Added from **Add cue > Placeholder**, it plays nothing: GO fires it, writes "nothing written yet" to the output panel and moves the playhead on, so numbering, the cue sheet and automatic continuations stay intact while a sound is still being designed or recorded. Its row carries a cool slate tint, diagonal hatching and the state word **PLACEHOLDER**, and the preflight names it every time you enter the Show Player until the cue is replaced (change Type in Identity) or deliberately disarmed.
- **project.ccshow** – The JSON manifest of a folder project, or a working copy inside a single-file project's private workspace. It stores editable show data; a saved single-file project also contains its media. See also *.ccshow*.

30.15 Q

- **Qt / PySide6** – Qt is a cross-platform application framework used to build the Curtain Call user interface, audio engine and PDF output. PySide6 is the official Python binding for Qt 6. Together they provide the windows, widgets, audio playback (via Qt Multimedia and FFmpeg) and print/PDF support. You do not need to know Qt to use Curtain Call, but error messages from the audio engine often mention Qt or FFmpeg component names.

30.16 R

- **Recipe** – See *Scene recipe*.
- **Release (ducking)** – The time in seconds it takes for a ducking source's targets to return from the reduced duck gain back to their normal level after the source stops playing. A short release (e.g. 0.5 s) snaps the background back quickly; a longer release (e.g. 4 s) makes the return feel gradual and natural. See also *Attack*, *Ducking*.
- **Relink** – The process of replacing a missing or wrong audio file reference in the Library while keeping the track's stable identity and all cue references intact. Use **Relink** for a single track or **Relink missing** to scan a folder for exact filename matches across multiple missing tracks.
- **Renumber** – A Cue Creator action that replaces all cue numbers with sequential integers (1, 2, 3…) in the current list order. It is a production edit (undoable) that affects the human-readable labels only; it does not change cue identities, targets, links or the playback order. Use it to tidy up a cue list after inserting, deleting or reordering many cues.
- **Run timeline** – A silent, read-only tool for reviewing and comparing flight recorder runs. The Timeline tab shows every event in a selected run; the Scrub slider lets you move through the run and see which cues had fired and what was sounding at any moment. The Compare tab shows timing differences between two runs. No audio is played during review.

30.17 S

- **Scene recipe** – A reusable template that defines a bundle of cues with placeholder track slots. When you insert a recipe, you assign real library tracks to the slots and the app creates a complete set of cues with fresh identities, blank hotkeys and no script links. Built-in recipes cover common patterns (scene opening with ambience, crossfade, announcement with ducking, blackout). You can also save your own from selected cues.
- **Scrub** – In the run timeline, a slider that lets you move through a recorded run's events without playing any audio. As you drag the scrub position, the display updates to show which cues had fired, what was sounding and what notes existed at that moment in the run. It is a silent inspection tool only.
- **Session log** – A running record of everything that happens during an app session: cues fired, errors, warnings, preflight results, drop pins and system messages. Visible in the output panel (`Ctrl+J`). Filters change what you see on screen but do not remove entries from the log. Export the complete log from **Project > Export > Session log** to preserve all evidence.
- **Show Score** – A read-only visual overview of the entire show drawn as a timeline, similar to a musical score. It uses the What-if simulator to place every cue in time, showing voice lanes, fade ramps, loop boundaries, ducking dips, silences and the loudest moment. It never plays audio. Accessible via `Ctrl+6`.
- **Pause** – A performance control (`Ctrl+Space` or the Pause button) that freezes the show where it is: every sounding cue holds its position, running fades and ducking keep their level, and a pre-wait or post-wait holds its remaining time. GO and editor previews are refused while paused. Pressing Pause again (**Resume**) carries on with the same cues from the same positions. Stop all, Panic and stop-all cues still work and end the pause. Pause differs from both Stop all and Panic, which stop audio instead of holding it, and from the screen lock (`Ctrl+Shift+L`), which covers the window while audio plays on normally.
- **Signal chain** – The physical path audio travels from the computer's software output through the audio interface, amplifier and speakers to the audience. Curtain Call controls the software end of this chain (cue volumes, fades, master, venue trim). Problems at any point in the physical chain — a muted amplifier, a disconnected cable, a wrong output selected in the OS — will cause silence that the app cannot detect or fix. Always verify the full signal chain before a performance.
- **Snapshot / baseline** – A lightweight copy of the show's state (cue fields, script structure, library

identities, note statuses) captured at a specific moment and stored inside the project. Created automatically when you create a rehearsal session; updated manually with **Update snapshot…**. Used by the change digest to compare two points in time. A snapshot is not a full backup and cannot be used to restore an earlier version of the show.

- **Sound caption** – A short human-authored description of what the audience should hear when a cue plays, written in plain language rather than technical settings. For example: *[soft harbour ambience with distant gulls and water]*. Captions appear on the next-cue card and Now Playing strips in Show Player, and are included in the printed cue sheet. They are optional and authored manually; the app never generates them automatically.
- **Stable ID** – An internal identifier generated by the app when a cue, track, script paragraph or rehearsal note is created. It never changes, even if you rename, renumber, reorder or undo edits. This is what keeps cue-to-track references, script anchors and ducking targets intact through normal editing. You never need to type or edit a stable ID directly.
- **State word** – A text label shown on cue rows and Now Playing strips in Show Player that describes what a cue is currently doing: **Pre-wait** (fired, counting down), **Playing** (audio active), **Fading** (level changing), **Muted** (output suppressed), **Ducked** (level reduced by a ducking source), **Paused** (the show is frozen by Pause and the cue holds its position), **Error** (playback failed). State words are shown in addition to colour so the view is readable without relying on colour alone.
- **Stop All** – An action (`Esc`) that fades all currently playing audio to silence using the project's default fade duration, and cancels all pending pre-waits, post-waits and delay timers. It is not instantaneous; use Panic (`Shift+Esc`) for an immediate cut. Stop All can itself have a continuation set, so a Stop All cue in the list can schedule a follower after it runs.

30.18 T

- **Test fire** – A button in Cue Creator that plays a cue immediately, ignoring its pre-wait and suppressing any automatic followers. It is an audition tool for checking a single cue's audio, volume and fade settings. It is not a simulation of the full sequence; use Show Player's GO for that.
- **Track** – An entry in the Library representing one audio recording. It stores the file path (relative to the project folder), a short code (e.g. AMB, BELL), metadata (artist, composer, album, year, genre, copyright, notes) and a stable ID. Playback settings (volume, fades, loops, ducking)

belong to cues, not tracks. One track can be used by many cues.

- **Track code** – A short, optional label you assign to a library track for quick identification, such as **AMB** for house ambience or **BELL** for a doorbell. It appears in the Library table and in cue inspector dropdowns. It has no effect on playback.

30.19 U

- **UTF-8** – A text encoding standard that can represent every character in every language, including accented letters, non-Latin scripts and special symbols. All Curtain Call project files, export text files and preferences are stored as UTF-8. This means project names, script text, cue names and notes can safely contain any language or character without corruption.

30.20 V

- **Venue profile** – A saved set of room-specific settings attached to a project: a trim value in dB (−24 to +6), an optional preferred output device name, setup notes and a checklist of manual checks. Activating a profile applies its trim as a multiplier on all output without changing any cue's saved volume. Profiles travel inside the project file and can also be saved to a reusable app-level library.
- **Venue trim** – The dB adjustment stored in an active venue profile. It multiplies the overall output level: −6 dB reduces output to about 50%, +3 dB requests about 141% (which may saturate the audio output and shows a headroom warning). It is separate from the project master volume and individual cue volumes. Changing venue trim requires audio to be stopped first.
- **Vibecoded** – A term used in this project to describe software written entirely by AI coding agents directed in plain language by a human author, without the author writing source code directly. The term is used honestly in the About section and this manual to set expectations about testing and reliability.

30.21 W

- **Waveform** – A visual representation of an audio file's amplitude over time, shown in full-width panels in Library and Cue Creator. In Library it can be zoomed and scrolled; clicking positions the playhead, then Space plays or pauses. Audio-processing previews overlay the original in purple. In Cue Creator it shows the selected cue's fade envelope and ducking band overlaid on the track. The waveform is peak analysis, not a calibrated loudness meter. It is cached in `library/.waveforms/` and rebuilt automatically if missing.
- **What-if simulator** – A silent, read-only tool that models how a sequence of cues will play out in time, given a set of assumed operator GO presses. It uses a detached snapshot of the show and never plays audio, moves the live playhead or changes the project. Useful for checking that automatic chains, ducking and timing work as intended before a rehearsal. Reopen it after editing to take a fresh snapshot.

30.22 X

- **xdg-email** – A Linux command-line tool that opens a new email draft in the user's default email application (Thunderbird, Evolution, KMail, etc.) with a pre-filled recipient, subject and body. Curtain Call uses it on Linux when you choose **Send by email** or **Open email draft**. If no email application is configured, the message is copied to the clipboard instead and a warning is shown. On Windows and macOS, a similar mechanism uses the system's default mail handler.

30.23 O

- **Operator** – The person running the show, configuring cues, handling pre-show checks, and operating the Show Player during performance.

30.24 P

- **Preflight** – Automated checks performed before starting a show (missing files, cue errors, metadata issues). Run via the preflight panel in Show Player.

30.25 V

- **Venue** – The location where a production is performed, including its acoustic profile and hardware setup. Venue profiles store gain settings and equipment lists.
- **Version** – The application’s release identifier (e.g., 1 . 2 . 3). Shown in Help > About Curtain Call and included in exported PDFs.

30.26 A work in progress: where Curtain Call may go next

This final section collects **development verification notes and proposed ideas**. Testing notes describe the scope of checks performed; proposals are not available features or delivery promises. There are no committed dates for proposals. The rest of the manual describes the app as it works today.

30.26.1 Platform testing and local-model verification

The author has tested the app's functionality on Windows 11 as well as Linux. On 12 November 2026, the author confirmed successful checks of numeric-field readability, window resizing/scaling and file associations, reference-video subtitle recording and overwrite with pause/seek/resume, second-screen subtitle and HoH presentation, and local-model installation, translation, cancellation and offline use, plus packaged-app installation on clean Linux and Windows 11 computers. These are author-reported workflow checks, not a complete rehearsal or live-show test. The author has not yet tested the app in a complete rehearsal or live performance on real show equipment. Review with deaf/hard-of-hearing audience participants also remains open. Automated regression checks support the workflow checks; they do not measure room acoustics or certify every hardware configuration. macOS remains untested. The startup feedback notice distinguishes Windows 11 from other Windows versions; this does not certify every configuration. Release engineering also tracks a native abort observed during a larger Python 3.10 / PySide6 6.12.0 test run with repeated HoH UI construction and outstanding final hosted full-suite checks. Focused lifetime/reset checks passed on Python 3.10 and 3.14; these do not close the wider investigation. Local-model installation sizes vary by platform and package version; further work includes measuring RAM, CPU and temporary-space requirements and checking model integrity. Individual packaged-model terms have not been independently verified; review them before use or redistribution.

30.26.2 Shared venue and equipment libraries

Online discovery, account management, publishing and synchronisation are not implemented. Possible additions include reviewed venue/projector suggestions and uploads, shared reference catalogues and project equipment selection. These remain proposals; current library data stays local.

The ambition is not simply to add more controls. These proposals explore how the script, cues, rehearsal history and audience access might work together, while keeping the operator in charge. The potentially innovative part is the combination in a theatre workflow; none of these ideas is claimed to be unique or proven better than other software. Each needs design, real-world testing and feedback

before it can earn a place in a show.

30.26.3 Further audience access ideas

Already available: HoH Accessibility, shared feeling profiles with general settings, character identification, sound descriptions, optional live expressive typography, an audience guide export, and editing-only reference video subtitle timing. These are described earlier in this manual, not future promises. Reference-video recording captures title start times; it does not record a new movie or provide live projection cues.

Still to verify with real audiences and venue equipment: translation accuracy, readability, speaker-name clarity, contrast, motion comfort and adequate display time. The author's successful second-screen checks above do not replace audience evaluation. Automatic punctuation-based feeling detection is a starting point for editorial review, not reliable interpretation of every performance. Keep movement optional and provide a still-text alternative. The project's roadmap also retains unresolved native-runtime and original Wayland-freeze verification; successful focused tests are not live-show certification.

Proposed: a synchronized rendered recording that includes authored sound captions, a more complete portable access/rehearsal handover pack, and broader language/localization support. These need permission and privacy controls, measured synchronization, and testing with deaf and hard-of-hearing people before they can be offered as dependable audience-access tools.

30.26.4 A show rendered for rehearsal and review

An offline **show bounce** could mix a recorded run into a single audio file, reconstructing cue times, fades, loops and ducking without playing it live. Markers could link listening positions to cue numbers, scenes and script lines. The distinctive opportunity is to turn the same show data used by the operator into a navigable rehearsal reference, rather than asking a team to assemble a separate recording by hand.

That foundation might support per-scene actor rehearsal tracks with space for spoken lines, or a phone video aligned to a recorded run so a team can jump to a cue. These uses require measured synchronization, rights to recordings and clear labelling of a rendered approximation versus actual captured output. Neither a bounce nor playback events would by themselves prove what an audience heard or establish performance-rights compliance.

30.26.5 Script-aware assistance with human approval

The script could become a source of **cue suggestions**: a direction such as “thunder” might be matched with a suitable library track and presented as a reviewable proposal linked to the line. An optional rehearsal assistant might later suggest the current script location from local speech recognition. The useful idea is to connect written dramatic context to the existing cue workflow, not to let software make artistic or performance decisions.

Suggestions would need to show uncertainty, handle ambiguous and multilingual dialogue and be accepted by a person before changing the show. Any speech recognition would require explicit privacy and recording choices. It must never press GO autonomously.

30.26.6 Reliability and show-day confidence through rehearsal tools

A proposed **readiness passport** could compare a portable summary of media fingerprints, expected equipment, rehearsal checks and unresolved notes between computers. A separate handover pack might collect now/next context, recent changes and emergency alternatives for a substitute operator. Instead of presenting only an isolated checklist, these concepts would tie preparation to the particular production and its rehearsal history. A sign-off would not be automatic permission to start playback.

A silent **ghost operator** exercise could compare a trainee's GO timing with a recorded run, highlighting early, late or missed cues without driving the house audio. Another proposal is a reviewed Go To: first show the operator which loops and levels would be needed at a chosen point, then require confirmation before any live reconstruction. Both would need trials with operators and clear safeguards around live audio.

30.26.7 More expressive audio and projection

Future audio concepts include cue trim points, separate house and headphone audition paths, measured signal meters and optional sound processing. These would let a designer prepare and audition more of a show inside one project, but depend on verified backend capabilities, real equipment and explicit handling of unavailable devices.

Video playback and mapping is a longer-term exploration: cues could eventually control video, while surface shapes, projector specifications and four-corner warping help place images on scenery. Connecting a venue's physical surfaces to a cue-driven show could help sound and projection teams discuss the same moments. Geometry, projector coverage and performance must be measured and

tested; a visual plan would not replace on-site alignment.

30.26.8 Shared context without surrendering control

A read-only stage-manager companion might show now/next cue and script context on a tablet, with an unmistakable disconnected indication. A separate research idea is a standby computer that can take over a show. Their potential value is shared situational awareness and a more informed handover, but they require tested networking, asset matching and clear live-machine status. Remote GO would be a separate, deliberately armed capability, not a default.

Other ideas include a Portuguese interface and broader localization, plus optional local search for sounds by description. All remain proposals, subject to practical testing, privacy, permissions, licences and available resources. You can help shape the direction through **Help > Send feedback or ideas…**. Do not share scripts or recordings without the necessary rights, and review the report and diagnostics before sending them.

Curtain Call

About this guide

A practical reference for preparing, rehearsing and operating theatre audio shows.

Version: v1.1.2 (build 20261117)

Website: https://curtaincall.yzonk.com

Feedback: curtaincall@yzonk.com

Before a performance, rehearse on the actual computer, audio interface and speakers, and keep a backup playback method ready.