Replica 11 — documentation & manual

A Windows tool that turns a long script into structured data for ElevenLabs: it detects scenes, characters, dialogue, narration and stage directions, lets you assign a `voice_id` to every character, and then exports the result or generates the audio directly.

1. Introduction

Replica 11 works in two modes that can be combined:

- **Local parser** — fully offline, no API key required. It detects the script structure using the rules described in Script format.
- **Process with ChatGPT** (optional) — sends the script to the OpenAI API, which cleans the text, converts it into structured JSON and optionally enhances it (adds emotion and interjections). Requires an OpenAI API key.

Voice generation through ElevenLabs is optional as well. Even without an ElevenLabs key you can process a script and export it to every supported format.

> **Nothing is sent without your consent**
> Text goes to OpenAI only after you confirm the dialog, and to ElevenLabs only after you press “Generate with ElevenLabs”.

2. Quick start

1. Paste your script into the **Script** panel on the left (or use **Load TXT**).
2. Click **Clean text** to remove separators and formatting noise.
3. Click **Analyze locally**. Characters and validation appear on the right.
4. In the **Characters & voice IDs** table, fill in an ElevenLabs `voice_id` for each character.
5. Review the **Dialogue** table at the bottom, then either **export** or click **Generate with ElevenLabs**.

3. User interface

Top bar

Element	Meaning
Project name	Used in exported file names.
Search	Finds a phrase in both the script and the dialogue table, scrolls to it and highlights it.
● Ready / ● Working…	Status indicator. Orange “Working…” means the app is waiting for an API response.
Save / Open project	Project file handling (see Projects).
Settings	API keys, models, UI language and main language, and other options.

Panels

- **Script** — input text and processing buttons.
- **Analysis result** — characters, validation, export, generation and log.
- **Dialogue** — an editable table of all lines (appears after analysis).

4. Script format

The local parser reads the script line by line and applies these rules in order:

Element	Rule	Example
Scene	An entire line in UPPERCASE (digits, spaces and dashes allowed).	`STREET`
Narration	A line starting with `Did you know?` (English) or `Věděli jste?` (Czech).	`Did you know? Front gardens…`
Dialogue line	Name, colon, text. The name may be at most 40 characters and 5 words, and must not contain `. ! ? …`	`Jacob: Grandma, will you buy me an ice cream?`
Timed line	A dialogue line with a cue in front. Translation keeps the cue and the speaker name. One speaker per line.	`[00:00:01.200-00:00:04.800] Jacob: Grandma, will you buy me an ice cream?`
Section	A short line up to 40 characters and 4 words, without a colon, not ending in `. ! ? …`	`Child`
Stage direction	Everything else.	`Grandma smiles and walks away.`

Example of a well-formatted script

```
STREET

Child
Jacob: Grandma, will you buy me an ice cream?
Anna: You say “please”, silly!
Grandma: Don't argue!
Did you know? Front gardens and active ground floors transform a street…
```

Narrator and Vypravěč

`Narrator` and the Czech `Vypravěč` are treated as **one and the same person**. They are merged into a single entry, so you assign the `voice_id` only once — English passages are then read by the same voice, just in English.

Parentheses as audio tags

Notes in round brackets are automatically converted into ElevenLabs audio tags:

In the text	Resulting tag
(směje se)	`[laughing]`
(šeptá)	`[whispering]`
(do telefonu)	`[on the phone]`
(nejistě)	`[hesitant]`
(chvíle ticha)	`[pause]`
(zavře oči)	`[softly]`
(úlevně)	`[relieved]`
(zklamaně)	`[disappointed]`

Unknown parentheses are left in the text unchanged.

5. Settings

UI language and main language

The list shows the English name, a dark gray separator, then the native name, for example Simplified Chinese | 中文（简体） or Spanish | Español (España). A parenthesis stays the regional name. This setting is the language of the interface and the main language of the app. Spoken numerals, dates and times in Clean text follow it, not the language of the sentence. The local cleanup speaks them in Czech or English. In Script to Speech the Clean text button does this even when an OpenAI key is saved, so the numerals are not rewritten into English. For any other language it leaves the digits; ChatGPT cleanup speaks them in the selected language. A language without its own translation keeps the rest of the interface in English. Help opens this manual in the same language. A language without its own manual stays in English. The change applies immediately; the application menu follows once settings are saved.

ElevenLabs

Option	Description
API key	Key from your ElevenLabs account. Stored locally and encrypted.
Default model	`eleven_v3` by default.
Output – quality / format	**HQ** = WAV 48 kHz, **LQ** = MP3 128 kbps. HQ may require a higher plan.
Default narrator voice ID	Voice used for the narrator when no dedicated voice ID is set.
Language tag conversion	`accent_tags` turns `<czech>` into `[Czech accent]`; `plain` just removes the tags.
Language override	Controls the `language_code` parameter sent to the API — see Audio tags & language.

OpenAI

Option	Description
API key	Key from platform.openai.com. Stored locally and encrypted.
OpenAI model	Model used for script normalization through Chat Completions. The default is GPT-6.2 Sol. GPT-6.2 Astra is the strongest option. A model already saved in Settings stays selected.
Processing mode	`local_parser_only` – local parser only. `openai_normalize` – structure determined entirely by OpenAI. `local_parser_then_openai_cleanup` – local parser as a hint, OpenAI cleans up the result (default).
Text enhancement	`off` – structure only. `tags_only` – add emotional audio tags. `tags_and_interjections` – tags plus short interjections.
Mark Czech names in English text	OpenAI wraps Czech names, cities and institutions in `<czech>…</czech>` tags.
Remove human separators	OpenAI strips formatting noise added by a human editor.

6. Processing a script

Clean text

A local pass over the input text that removes:

- lines made only of separators (`---` , `***` , `===` , `~~~` , `###` ...),
- stray punctuation artifacts (a lone period or ellipsis on its own line),
- bullets and numbering at the start of a line,
- inline separators and redundant spaces,
- three or more blank lines, collapsed into one.

It also fixes doubled colons after a speaker name (`Karolína:: text` → `Karolína: text`).

Numerals, dates and times are spoken in the **UI language and main language**. A Czech sentence with the main language set to English is read with English numerals. An English sentence with the main language set to Czech is read with Czech numerals. Local cleanup does this for Czech and English, and in Script to Speech the button uses it even when an OpenAI key is saved, so the numerals are not rewritten into English. Spanish and the other languages keep the digits in the local pass.

Analyze locally

Runs the local parser. Besides the structure it also suggests a **voice description** (always in English, because ElevenLabs expects English descriptions) and an **English audio tag** in the Direction column based on context — for example `[curious]` for a question, `[excited]` for an exclamation, `[thoughtful]` for an ellipsis.

Process with ChatGPT

Sends the script to the OpenAI API. If enhancement is enabled in Settings, the button shows **(+ enhance)** and emotion plus interjections are added as well. A confirmation is always requested before sending.

> **Assigned voices are preserved**
> Existing `voice_id` values, voice descriptions and stability are not overwritten by ChatGPT processing — character matching ignores letter case and diacritics.

Review result (OpenAI Review)

Opens a window comparing the original text with the detected structure: scenes, characters, lines and any warnings with a confidence level. You can edit character names and line text here. The result is applied to the main app only when you press **Approve**.

Narrator reads everything without a role

A checkbox below the buttons. When enabled, scene titles, section titles and stage directions are also turned into spoken narrator lines. Useful for audiobooks and podcasts where everything should be heard.

Translate transcription

Optional ChatGPT tool (**File** → **Translate transcription...** or the Script toolbar). Choose a target language and optionally keep secondary-language words. You can translate the whole script, a selection in the Script panel, or a selection inside a dialogue row.

- **Selection in a dialogue row** — the phrase is replaced in that row only. The script window stays put and the script is not analyzed again.
- **Whole script / selection in the Script panel** — the Script text is replaced (the view may jump to the start). Dialogue rows update only after you analyze and approve the structure again.

Requires an OpenAI API key. It does not call ElevenLabs.

7. Characters & voices

The **Characters & voice IDs** table shows for every character:

Column	Description
Character	Name; editable.
Line count	How many times the character speaks.
Voice ID	ElevenLabs `voice_id`. Without it the line will not be generated.
Voice description	A suggestion in English (age, gender, tone). Use it as a basis for picking or designing a voice.
Stability	**Creative** (0.0) – most expressive, **Natural** (0.5) – balanced, **Robust** (1.0) – most stable and least variable.

Below the table you will find **Merge characters**, which combines two names into a single character (for example a typo or an alias).

> **Where do I find a voice_id?**
> Open the voice library in your ElevenLabs account; the `voice_id` is the voice identifier shown in its detail view or returned by the ElevenLabs API.

Voice library

Open the **Voices** tab next to **Script**. Pasting a Voice ID is still the main way to assign a voice. **Fetch voices** loads ElevenLabs My Voices with the API key already saved in Settings. For each voice you can set a role, a description, Stability and Similarity.

- A new fetch adds voices and keeps the roles, descriptions and presets you already filled in.
- A voice that is no longer in My Voices stays in the library and is marked.
- After analysis, a role that matches a character fills an empty Voice ID, Stability and Similarity.
- A Voice ID already typed on the open project is not overwritten.
- The library is stored in `%APPDATA%\d3arts\Replica 11` and survives a reinstall.

Dubbing

Dubbing in Replica 11 is text to speech. It does not call the ElevenLabs Dubbing API and it does not copy the original performance. Tone comes from the Voice ID, Stability, Similarity and the audio tags already in the line.

1. Put timed lines in the script, or click **Transcribe audio** when an OpenAI key is saved. Without that key, paste the timed lines yourself. The app does not bundle a local Whisper model and it does not detect who is speaking: each line is one speaker named `Speaker` until you rename it.
2. Analyze, assign voices and translate as usual. A translation may change the words. The timestamps and the speaker names stay.
3. Set the output format to HQ WAV, then click **Build dub mix**.

Each line is generated on its own and placed at its cue. The gaps are silence and the mix is as long as the source. If a line is longer than its slot, it is kept in full and the app reports it. Nothing is time-stretched, and the mix does not use ffmpeg.

8. Dialogue table

Appears at the bottom of the window after analysis and is fully editable.

Column	Description
Scene / Section	Where the line belongs (for orientation only).
Character	Who is speaking.
Voice ID	Taken from the characters table; can be overridden per line.
Text	The spoken text itself.
Direction	Audio tag in square brackets, e.g. `[excited]`.
Language	`cs` or `en` — affects batch splitting and `language_code`.
Chars	Text length (relevant for the 2000-character batch limit).
Skip	A checked line is neither exported nor generated.

Tools above the table

- **Filters** by scene, character and language.
- **Czech / English** — sets the language for all lines at once.
- **+ Add line** — appends at the end; the row-level + inserts a line below that row.
- 🗑 on a row deletes a single line.
- 🗑 **Delete all** — deletes all lines; if a character filter is active, only that character's lines are deleted.
- ↩ **Undo** (Ctrl+Z) — reverts the last change; the number of steps (10–200, default 50) is set in Settings → Editing. The button shows how many steps are available. Clean text, Translate and a new analysis can be undone too. History is kept in memory only and is cleared on New / Open project.
- ↪ **Redo** (Ctrl+Y or Ctrl+Shift+Z) — reapplies an undone change.

> **Undo history is cleared**
> Running a new analysis, approving a ChatGPT structure or opening a project empties the history.

9. Audio tags & language

Audio tags

The content of the **Direction** column is prefixed to the line text during export and generation, so ElevenLabs v3 receives e.g. `[excited] That's great!`. You can use any tag that ElevenLabs supports.

Language tags for Czech names in English text

ElevenLabs has no dedicated per-word language tag. The app therefore uses its own paired tags, converted to accent audio tags on export:

```
Input:   We visited <czech>Hradec Králové</czech> last summer.
Output:  We visited [Czech accent] Hradec Králové [English accent] last summer.
```

The `plain` option in Settings only strips the tags without switching accents. ChatGPT can insert the tags for you (the “Mark Czech names…” option), or you can write them manually.

Language override

Value	Behaviour
`auto`	Language is derived from the lines in the batch (default).
`en`	Always sends `language_code: "en"`.
`cs`	Always sends `language_code: "cs"`.
`off`	The parameter is not sent; ElevenLabs detects the language itself.

10. Export

Export works even without an ElevenLabs key.

Format	Contents and use
TXT	Readable transcript grouped by scene as `Character: [tag] text`. Good for pasting manually into Text to Speech.
JSON (Text to Dialogue)	Payload with `model_id` and an `inputs` array (`text` + `voice_id`) for the Text to Dialogue API.
Studio JSON	Structure for ElevenLabs Studio (`from_content_json`) split by scene and section.
CSV	A table of all lines including scene, section, voice ID, direction, language and character count. Handy for review in Excel.
Markdown	Readable script with scene and section headings. Handy for proofreading.

Lines marked as **Skip** are excluded from all exports.

11. Audio generation

The **Generate with ElevenLabs** button splits the lines into batches and sends them to the Text to Dialogue API one by one. A new batch starts whenever a limit would be exceeded:

- at most **2000 characters** per batch,
- at most **10 distinct voice IDs** per batch,
- a change of **language** (each batch is single-language),
- a change of **stability** (each batch is homogeneous).

A single line is never split and the order is always preserved.

Audio folder

The default location is `Documents\Replica 11\exports\audio`. The same folder holds the WAV and MP3 files and `generation.log`. Use 📂 **Open** to show it in File Explorer and 📁 **Change** to pick a custom location. **Delete audio files** removes the WAV and MP3 files and leaves the log. The log has no folder of its own. A new generation replaces `generation.log`.

Generation log

Reports the result of every batch, including the language and stability used, and writes the same lines to `generation.log` in the audio folder. Failed items include the API response, so you can spot issues such as exhausted credit or an invalid voice ID.

12. Projects

Save project (Ctrl + S) creates a `*.replica-11-project.json` file containing the project name, the original text, the parsed and normalized structure, the voice mapping and project settings (model, output format, processing mode…).

Open project (Ctrl + O) and **Open Recent** restore a previously saved file. **New Project** (Ctrl + N) clears the current workspace (script, dialogue, review). If there is unsaved work, the app asks whether to save first (Save / Don't Save / Cancel).

> **API keys are never stored in a project**
> You can share the project file safely — it never contains your keys.

Older files created under the application's previous name can also be opened.

13. Menu & shortcuts

Menu	Contents
File	New Project (Ctrl + N), Open Project (Ctrl + O), Open Recent, Save Project (Ctrl + S), Load TXT (Ctrl + Shift + O), Translate transcription..., Quit
Edit	Undo, Redo, Cut, Copy, Paste, Select All
View	Reload, Developer Tools, zoom, full screen
Help	Documentation & manual, License, Check for Updates, About, website link

F1 opens this documentation, F12 the developer tools, Ctrl + C / Ctrl + V copy and paste, and Enter in the search field starts the search.

Codex

Codex can drive the Replica 11 window that is already open. It does not start a second copy and nothing is entered in Settings. The connection stays on this computer. API keys and license secrets are not part of the answers. A cloud Codex session on another machine cannot reach this window.

1. Install Node.js so the `node` command works in a terminal.
2. Leave Replica 11 open. While it runs, it writes the local address to `%APPDATA%\d3arts\Replica 11\control.json` and deletes that file on exit.
3. Open the project folder `D:\Python\11ka` in Codex and trust the project if Codex asks. Codex then reads `.codex/config.toml` and starts the local adapter `scripts\replica-mcp.mjs`.
4. If Codex is not opened from that folder, copy the same block into `%USERPROFILE%\.codex\config.toml`:

```
[mcp_servers.replica11]
command = "node"
args = ["D:/Python/11ka/scripts/replica-mcp.mjs"]
startup_timeout_sec = 20
```

Then ask Codex to use the `replica11` tools. For example: read the state, set the script, and analyze it locally. The same actions as the buttons change the open window: script text, Analyze locally, Script to Speech or Dubbing, voice IDs, and saving or opening a project.

Paid generation is a separate tool. Codex first receives the scope (lines, characters, batches) and calls ElevenLabs only after an explicit confirmation. The license is still checked inside generation, the same way as the Generate button. If the window is closed, the tools report that Replica 11 is not running.

14. Privacy & keys

- API keys are stored **locally only**, encrypted via Electron safeStorage.
- Keys never reach the application interface — all API calls happen in the main process.
- Keys are not part of exports or project files.
- Unless you explicitly trigger a feature, your text never leaves your computer.

15. Troubleshooting

Problem	Solution
OpenAI error 429 “exceeded your current quota”	Your OpenAI account is out of credit or over its limit. Check billing at platform.openai.com — this is not an application bug.
A line was not generated	The `voice_id` is missing. Fill it in for the character, or set a default narrator voice.
Generation fails with WAV output	HQ (WAV 48 kHz) may require a higher ElevenLabs plan. Switch to LQ (MP3 128 kbps) in Settings.
Lines detected incorrectly	Check the formatting against Script format, run **Clean text**, or use ChatGPT processing.
A character appears multiple times	Use **Merge characters** below the characters table.
English text sounds Czech	Set the line language to `en` (the **English** button) and set Language override to `English` in Settings.
API keys are missing after a reinstall	Keys are tied to the application's user profile. Enter them again in Settings.

Replica 11 — developed by Petr Chroustovsky, d3arts.cz. The application version is shown in **Help** → **About**.