TAKEOVER TOOLS
UNITY
TRANSLATOR PRO DOCUMENTATION

GETTING STARTED

What is Translator Pro?

Translator Pro is an AI-powered all-in-one localization platform for Unity, built to automate the entire translation workflow from text and voice to subtitles, images, and store listings.

It combines Editor automation, runtime tracking, and multi-model quality control (QC) into a single, seamless system. With just a few clicks, developers can localize an entire project in minutes (depending on project size) and reach players in over 339+ languages.

Ask Anything, Anytime in Your Own Language Takeover GPT is your personal AI assistant. Curious how it works? Not sure how to set it up? Want to know if your project is compatible? Just ask in your own language, in your own words.

- 👉 Talk to "Translator Pro GPT" now

WHAT PROBLEMS DOES IT SOLVE?

Game localization has traditionally been one of the slowest, most expensive, and error-prone parts of game development. Developers often spend weeks or even months extracting texts, organizing spreadsheets, sending files to translators, reviewing results, fixing context issues, and repeating the process across dozens of languages. Even after all that work, translations frequently lose the original intent, tone, and meaning of the game. Translator Pro changes this completely. It replaces a fragmented, manual workflow with a fully automated AI localization pipeline inside Unity turning a process that normally takes months and thousands of dollars into something that can be completed in minutes. But speed is only part of the solution. Translator Pro also removes much of the hidden technical workload that usually comes with shipping a game globally. From RTL support and Smart Strings to Unicode compatibility and other localization-specific requirements, many of the issues developers typically have to solve manually are already handled as part of the workflow behind the scenes. At the same time, Translator Pro improves quality through a context-aware AI pipeline that understands meaning before translation and compares results across multiple leading AI models to produce smoother, more natural outcomes. The result is a localization process that is faster, more accurate, and significantly more practical for real production helping developers bring their games to players worldwide with far less time, cost, and technical overhead.

PRE-REQUISITES

==

IMPORTANT – API Keys Required

==

Translator Pro uses multiple AI providers for translation, quality control, voice cloning, and image localization.
You must configure the required API keys before using the system.

Add your keys inside the Translator Pro Settings window in Unity.

Required for Core Translation

At least ONE of the following AI providers must be configured in order to generate translations:

OpenAI API Key – Required for Quality Control (QC) and recommended as the main translation model

Google Gemini API Key – Used for image localization and multimodal translation

xAI Grok API Key – Optional model that can participate in translation or arbitration

DeepSeek API Key – Optional model used for translation comparison and referee system

⚠ Important:
To perform translations, at least ONE of these four AI models must be configured.

Voice Generation

VoicePack supports multiple providers.

- OpenAI
  Used for AI voice generation and dubbing.
- ElevenLabs *(Optional)*
  Used specifically for voice cloning.

If ElevenLabs is not configured, VoicePack will still work normally using OpenAI voices.

ElevenLabs is only required for advanced voice cloning workflows and for clips that contain multiple speakers.

===

IMPORTANT – FFmpeg Setup Required

===

To use the VoicePack features of Translator Pro, you must manually install FFmpeg.

Due to Unity Asset Store guidelines, we cannot ship ffmpeg.exe inside this package.

Follow these steps:

1. Download the latest **ffmpeg-git-full.7z** build from the official FFmpeg site:
   https://www.gyan.dev/ffmpeg/builds/

2. Extract the archive. Inside the "bin" folder you will find **ffmpeg.exe**.

3. Copy **ffmpeg.exe** into the following folder in your Unity project:

Assets/Takeover/Translator Pro/VoicePack/FFmpeg/

4. Restart Unity.

If FFmpeg is missing, the console will show:

```
Debug.LogError("ffmpeg.exe not found in package.");
```

Make sure you have copied the correct file to the correct folder.

THIRD PARTY SERVICES

Translator Pro uses multiple third-party AI and platform APIs.

This asset does NOT include any API credits.

To use Translator Pro you must provide your own API keys for one or more of the
Following services:

- OpenAI (GPT)
- Google Gemini
- Grok (xAI)
- DeepSeek
- ElevenLabs
- Google Play Developer API

Some of these services may require a paid subscription or usage-based billing.

Pricing, free tiers and limits are determined solely by the respective providers.

Translator Pro does not control or modify third-party pricing in any way.

HOW TO GET API KEYS

OpenAI:

https://platform.openai.com/account/api-keys

Google Gemini:

https://ai.google.dev

Grok (xAI):

https://x.ai

DeepSeek:

https://www.deepseek.com

ElevenLabs:

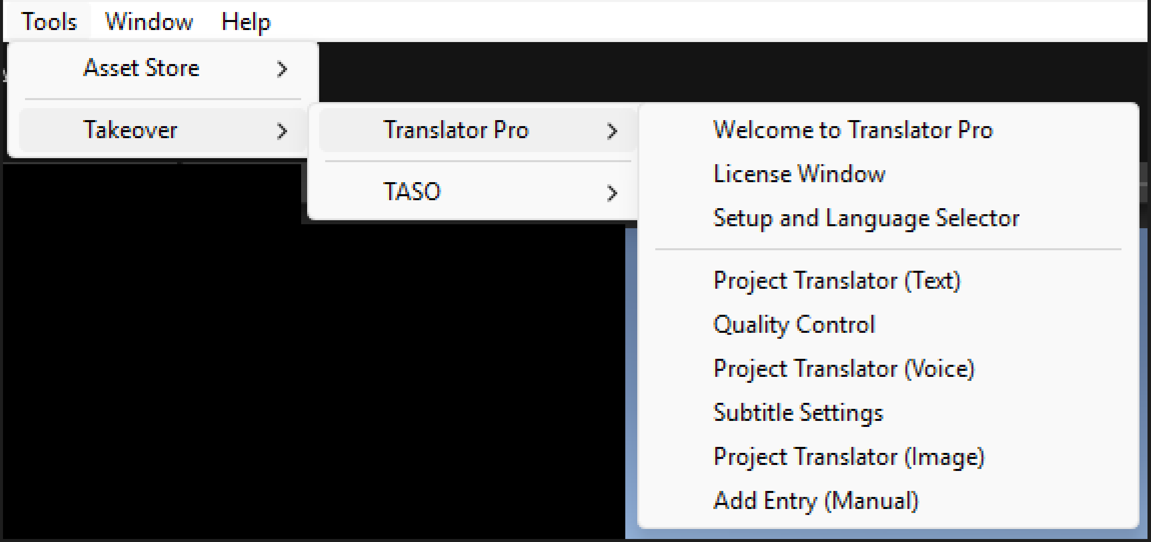
https://elevenlabs.io

Google Play Developer API:

https://developers.google.com/android-publisher/getting_started

---

CORE EDİTOR WİNDOWS


Tools
Window
Help
Asset Store
Takeover
Translator Pro
TASO
Welcome to Translator Pro
License Window
Setup and Language Selector
Project Translator (Text)
Quality Control
Project Translator (Voice)
Subtitle Settings
Project Translator (Image)
Add Entry (Manual)

(Translator Pro menu integrated into the Unity top bar)

This image shows the quick-access menu of Translator Pro integrated into the Unity top menu. You can think of it as a control center where you can access all steps from the

Welcome Screen (and additional tools) with a single click: license/setup, project translations (text/voice/image), quality control, subtitle settings, and single-file tools are all gathered here.

Takeover → Translator Pro Menu (Line-by-line brief explanations)

Welcome to Translator Pro
Opens the Translator Pro welcome / quick start window. This is where you begin the step-by-step setup and workflow.

License Window
Opens the window where you enter, verify, and activate your license.

Setup and Language Selector
Opens the section where you select the source and target languages for the project and configure the basic setup settings.

Project Translator (Text)
Runs the bulk translation workflow for all text content in the project into the selected languages.

Quality Control
Runs the generated translations through the quality control filter; detects issues and supports the correction workflow.

Project Translator (Voice)

Processes voice content in the project through the dubbing / voice localization pipeline.

Subtitle Settings

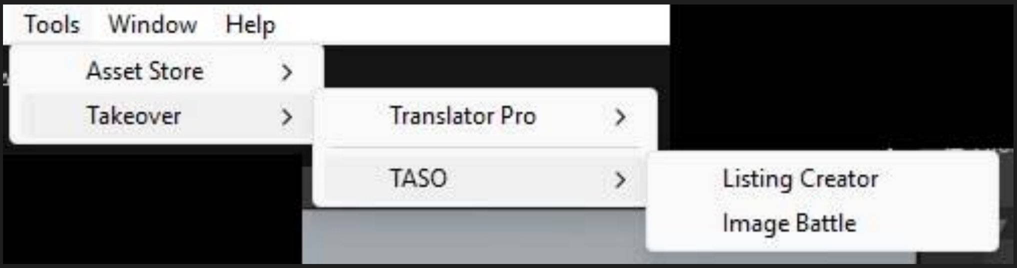
The window where you configure subtitle behavior such as font, size, display duration, and fade timing.

Project Translator (Image)

Runs the batch image localization workflow for visual content in the project (such as text inside images).

Add Entry (Manual)

This is the most accurate way to add new entries. It records the entry together with its meaning and usage context and saves it to JSON. As a result, every entry you add is fully included in the translation and QC processes.

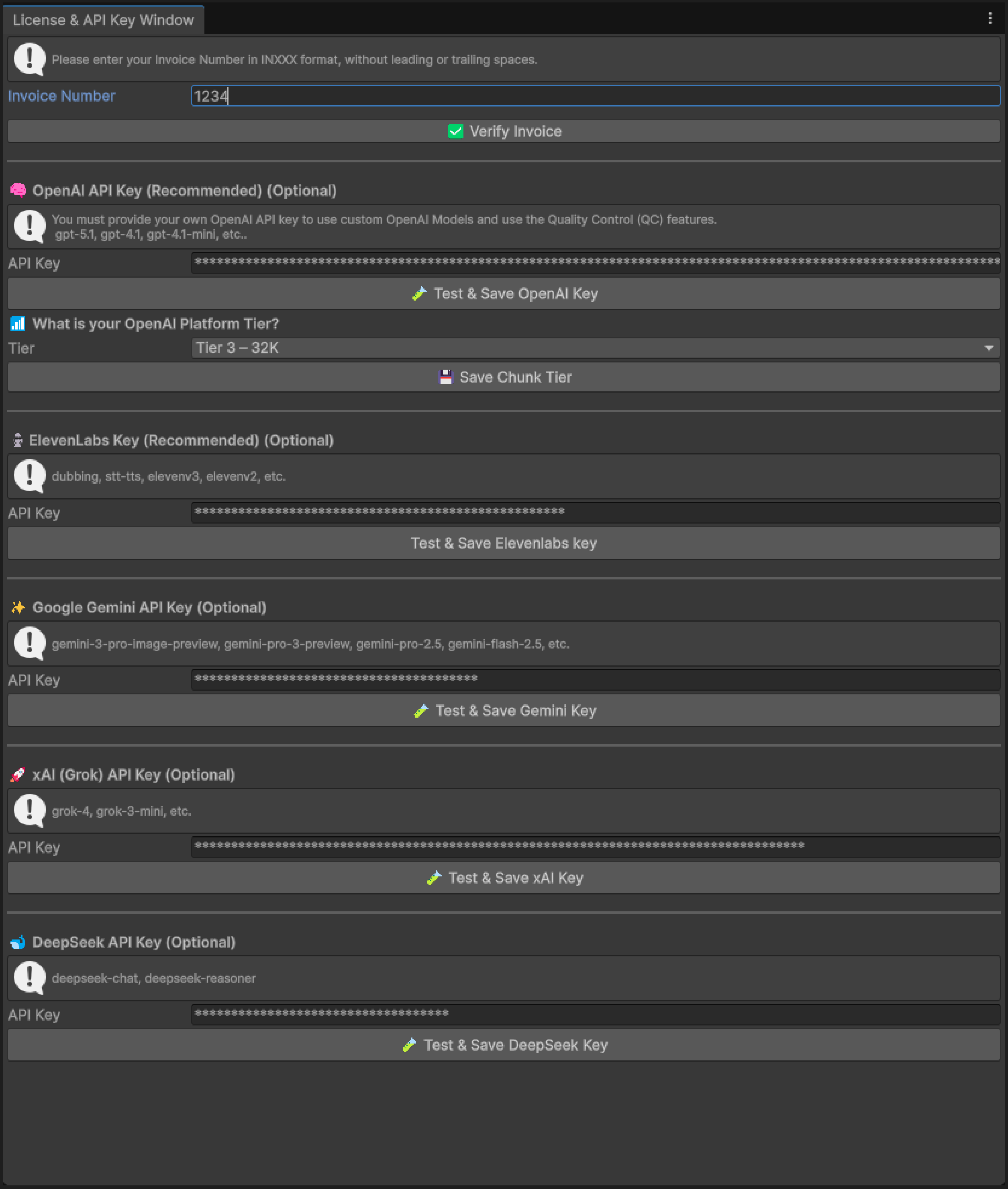


(TASO tools integrated into Unity, including Listing Creator for store optimization and Image Battle for generating and testing visual variations.)

The TASO (Takeover ASO) menu inside Unity, providing tools for store listing creation and visual A/B testing.

Listing Creator
The main hub of TASO. All workflows related to store listings (optimization, localization, content generation, etc.) are managed here through organized tabs.

Image Battle
Allows you to design icons, feature graphics, and similar assets by generating unlimited variations using carefully crafted prompts and an experiment-improve approach. It is especially useful as a fast and efficient production space for listing A/B testing.



License & API Key Window

Please enter your Invoice Number in INXXX format, without leading or trailing spaces.

Invoice Number: 1234

✅ Verify Invoice

🧠 OpenAI API Key (Recommended) (Optional)

You must provide your own OpenAI API key to use custom OpenAI Models and use the Quality Control (QC) features.
gpt-5.1, gpt-4.1, gpt-4.1-mini, etc..

API Key: **

🧪 Test & Save OpenAI Key

📊 What is your OpenAI Platform Tier?

Tier: Tier 3 – 32K

💾 Save Chunk Tier

ElevenLabs Key (Recommended) (Optional)

dubbing, stt-tts, elevenv3, elevenv2, etc.

API Key: ***

Test & Save Elevenlabs key

✨ Google Gemini API Key (Optional)

gemini-3-pro-image-preview, gemini-pro-3-preview, gemini-pro-2.5, gemini-flash-2.5, etc.

API Key: **

🧪 Test & Save Gemini Key

🚀 xAI (Grok) API Key (Optional)

grok-4, grok-3-mini, etc.

API Key: **

🧪 Test & Save xAI Key

🐳 DeepSeek API Key (Optional)

deepseek-chat, deepseek-reasoner

API Key: **************************************

🧪 Test & Save DeepSeek Key

What is this window? (Short overview)

This is the central window used to activate Translator Pro and configure the AI service API keys required for the tools you want to use.

From here you can:

- **Verify your license using your Invoice Number**
- **Enter, test, and save API keys for OpenAI / ElevenLabs / Gemini / xAI / DeepSeek**
- **Select your OpenAI Platform Tier / Chunk Tier to optimize Quality Control (QC) performance**

Fields and Buttons on the Screen (Line-by-Line)

1) Invoice Number (INXXX format)

Enter the Invoice Number from your purchase to verify your license.

---

2) Verify Invoice

Checks the entered Invoice Number and activates the license if it is valid.

OpenAI API Key (Recommended) *(Optional)*

API Key

Enter your OpenAI API Key to enable OpenAI models and QC features.

(Examples mentioned in the interface include models such as gpt-5.1, gpt-4.1, and gpt-4.1-mini.)

Test & Save OpenAI Key

Tests the API key and saves it if it is valid.

What is your OpenAI Platform Tier? (Tier)

Select the tier of your OpenAI account based on your usage limits and capacity.

This setting helps Translator Pro determine the best chunking and processing strategy, especially when working with large projects.

Save Chunk Tier

Saves the selected tier/chunk configuration so the pipeline can operate accordingly.

ElevenLabs Key (Recommended) *(Optional)*

API Key

Enter your ElevenLabs API Key for voice-related workflows, including dubbing and STT/TTS processes.

Test & Save ElevenLabs Key

Tests the API key and saves it if it is valid.

Google Gemini API Key *(Optional)*

API Key

Enter your Gemini API Key to use Gemini models, particularly for image and video workflows.

Test & Save Gemini Key

Tests the API key and saves it if it is valid.

---

xAI (Grok) API Key *(Optional)*

API Key

Enter your xAI API Key if you want to use Grok models.

Test & Save xAI Key

Tests the API key and saves it if it is valid.

DeepSeek API Key *(Optional)*

API Key

Enter your DeepSeek API Key to use DeepSeek models.

Test & Save DeepSeek Key

Tests the API key and saves it if it is valid.

What is your OpenAI Platform Tier? (Tier)

This is the tier you select based on your OpenAI account's **usage limits and capacity level**. It helps Translator Pro choose the right **chunking (splitting) and request management** strategy, especially for large projects.
In general, a **higher tier** means more available capacity, which typically translates to **faster processing** and, in many scenarios, **more cost-efficient** translations.

Save Chunk Tier

Saves your selected tier/chunk configuration so the Translator Pro pipeline can tune **request size, chunking behavior, and rate-limit handling** accordingly.

Important Notes (API Keys)

OpenAI API Key - Mandatory

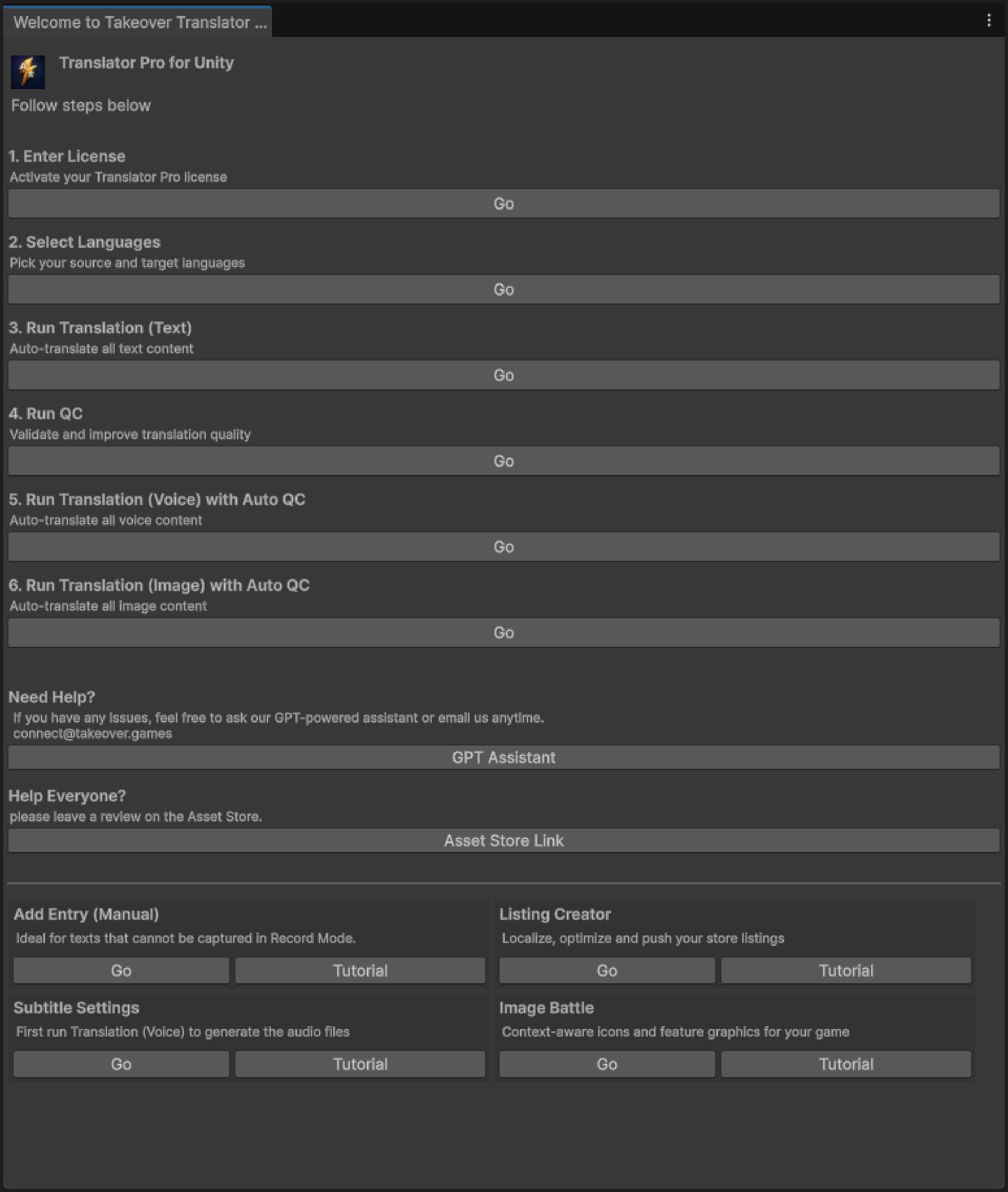
Even if the UI says "optional", it is **effectively mandatory**. Translator Pro's **Quality Control (QC)** stages rely on **OpenAI models only**; without this key, the core QC functionality is unavailable and the overall product becomes heavily limited.
Note: For initial "raw" translations (before QC), you may use other models (Gemini, Grok, DeepSeek, etc.). These keys can also be used to gather additional opinions as a **second/third reviewer** during QC. However, the **core QC pipeline is OpenAI-based**.

ElevenLabs Key - Required for Voice Cloning

Without an ElevenLabs key, you cannot use **voice cloning.**
When you enable **ElevenLabs + Dubbing**, Translator Pro can **clone the original voice** and generate dubbed speech across all target languages.
If you don’t provide an ElevenLabs key, the system falls back to an **OpenAI-based** voice workflow where you typically select a voice/model **per character**. This is often sufficient for games with one or two speaking characters, but if you want **true voice cloning**, ElevenLabs is **required**.

Google Gemini API Key Required for Image Localization (Google AI Studio)

The Gemini key is critical for **image localization / image translation** workflows. Translator Pro expects a **Google AI Studio API Key** (not Vertex AI).
Without a Gemini key, image localization features may not work properly. This can impact both **Project Translator (Image)** and certain **TASO** image workflows. If you plan to use image features, providing the Gemini key is **strongly recommended** (effectively required for image localization).



Translator Pro for Unity

Follow steps below

1. Enter License
Activate your Translator Pro license

2. Select Languages
Pick your source and target languages

3. Run Translation (Text)
Auto-translate all text content

4. Run QC
Validate and improve translation quality

5. Run Translation (Voice) with Auto QC
Auto-translate all voice content

6. Run Translation (Image) with Auto QC
Auto-translate all image content

Need Help?
If you have any issues, feel free to ask our GPT-powered assistant or email us anytime.
connect@takeover.games

Help Everyone?
please leave a review on the Asset Store.

Add Entry (Manual)
Ideal for texts that cannot be captured in Record Mode.

Listing Creator
Localize, optimize and push your store listings

Subtitle Settings
First run Translation (Voice) to generate the audio files

Image Battle
Context-aware icons and feature graphics for your game

This screen is the **Translator Pro Welcome / Quick Start window.**
It provides a recommended **step-by-step workflow** for setting up and running localization in your project.

Each step includes a **Go** button that opens the related tool, allowing you to start the entire localization pipeline

---

Steps

1. Enter License

Opens the window where you enter and verify your Translator Pro license. License activation is performed here.

2. Select Languages

Select the **source language** and **target languages** for your project. All translation operations will follow this language configuration.

3. Run Translation (Text)

Automatically translates all text content in your project into the selected target languages.

4. Run QC

Runs the **Quality Control (QC)** pipeline to validate translations and improve overall translation quality and accuracy.

5. Run Translation (Voice) with Auto QC

Automatically localizes voice and dialogue content into the selected languages using **voice dubbing**, followed by automatic quality control.

6. Run Translation (Image) with Auto QC

Localizes text found inside images and runs the results through the quality control process.

Additional Tools

Add Entry (Manual)

Used to **manually add localization entries to the table** for texts that cannot be captured automatically by Record Mode.

Subtitle Settings

Allows you to configure **subtitle appearance and behavior**, such as font, size, and display duration for voice localization.

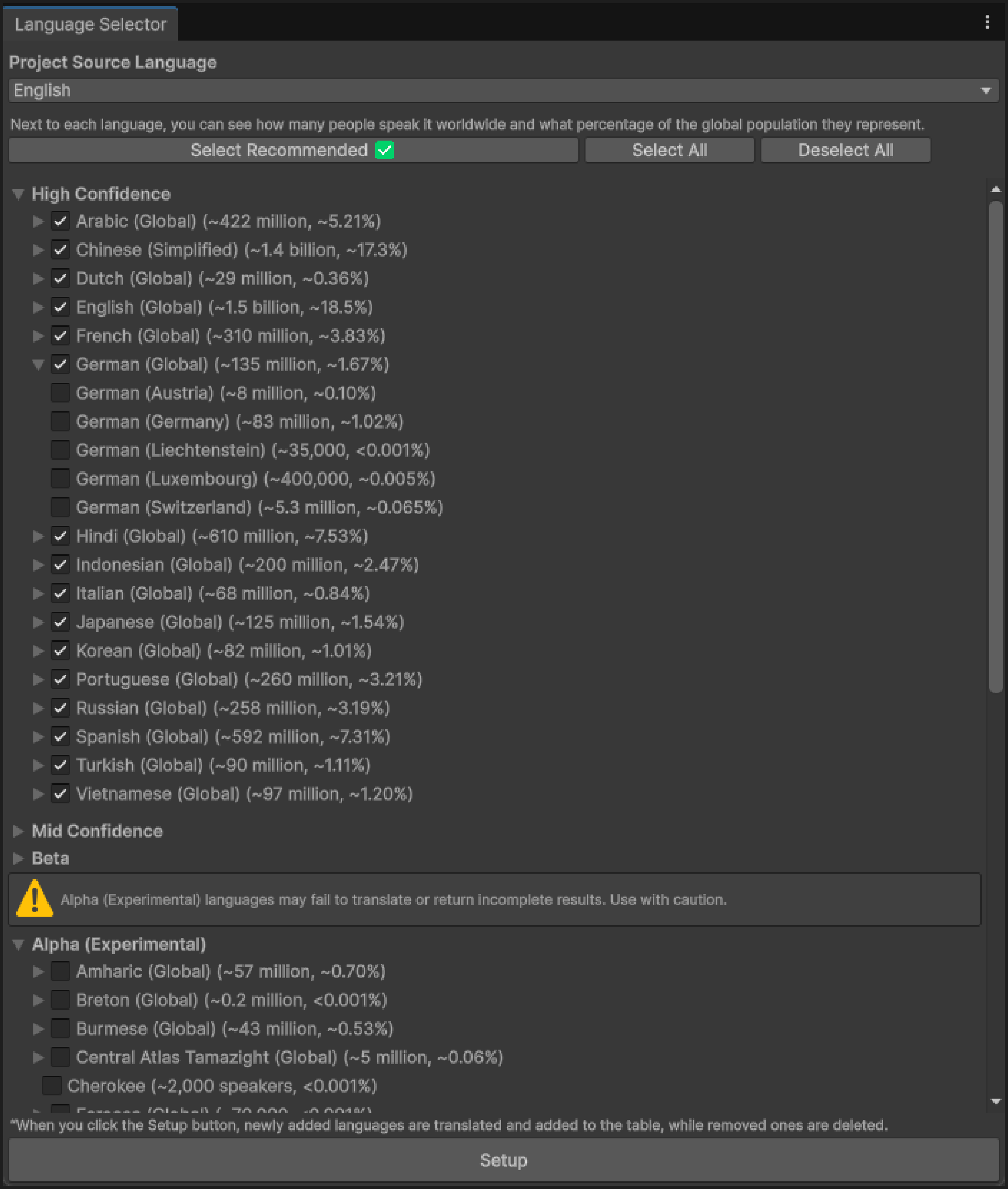
TASO Tools

Listing Creator

The main control center of the TASO system. Used to **localize, optimize, and manage your store listings**.

Image Battle

Generates and compares multiple visual variations for **game icons and store graphics**, helping you run better store listing experiments and optimizations.



Project Source Language

English

Next to each language, you can see how many people speak it worldwide and what percentage of the global population they represent.

High Confidence

- ☑ Arabic (Global) (~422 million, ~5.21%)
- ☑ Chinese (Simplified) (~1.4 billion, ~17.3%)
- ☑ Dutch (Global) (~29 million, ~0.36%)
- ☑ English (Global) (~1.5 billion, ~18.5%)
- ☑ French (Global) (~310 million, ~3.83%)
- ☑ German (Global) (~135 million, ~1.67%)
  - □ German (Austria) (~8 million, ~0.10%)
  - □ German (Germany) (~83 million, ~1.02%)
  - □ German (Liechtenstein) (~35,000, <0.001%)
  - □ German (Luxembourg) (~400,000, ~0.005%)
  - □ German (Switzerland) (~5.3 million, ~0.065%)
- ☑ Hindi (Global) (~610 million, ~7.53%)
- ☑ Indonesian (Global) (~200 million, ~2.47%)
- ☑ Italian (Global) (~68 million, ~0.84%)
- ☑ Japanese (Global) (~125 million, ~1.54%)
- ☑ Korean (Global) (~82 million, ~1.01%)
- ☑ Portuguese (Global) (~260 million, ~3.21%)
- ☑ Russian (Global) (~258 million, ~3.19%)
- ☑ Spanish (Global) (~592 million, ~7.31%)
- ☑ Turkish (Global) (~90 million, ~1.11%)
- ☑ Vietnamese (Global) (~97 million, ~1.20%)

Mid Confidence

Beta

⚠ Alpha (Experimental) languages may fail to translate or return incomplete results. Use with caution.

Alpha (Experimental)

- □ Amharic (Global) (~57 million, ~0.70%)
- □ Breton (Global) (~0.2 million, <0.001%)
- □ Burmese (Global) (~43 million, ~0.53%)
- □ Central Atlas Tamazight (Global) (~5 million, ~0.06%)
- □ Cherokee (~2,000 speakers, <0.001%)
- □ Faroese (Global) (~70,000, <0.001%)

*When you click the Setup button, newly added languages are translated and added to the table, while removed ones are deleted.

Language Selector is the window where you choose the **project source language** and your **target languages** for Translator Pro's project-based localization pipeline. The target languages you select here are added to your translation tables, and all subsequent steps (**Text / QC / Voice / Image**) run based on this language list.

Fields and buttons (line by line)

Project Source Language

Select your project's main language (e.g., English). This defines **the language you translate from**.

Info line (the description at the top)

Next to each language, Translator Pro shows the **estimated number of speakers** and the **percentage of the global population**. This helps you decide which languages to prioritize.

Target language selection tools

Select Recommended

Selects Translator Pro's recommended "most common / most useful" target languages with a single click. Ideal for a fast start.

Select All

Selects all target languages currently visible in the list.

Deselect All

Clears all selected languages (bulk reset).

Language list and groups

High Confidence

The most stable languages with the highest chance of smooth, reliable results. This is the primary recommended group for general use.

Mid Confidence

Usually works well, but quality/compatibility can be more variable depending on content type.

Beta

Supported languages that are still in an active improvement/testing stage and may require extra caution.

Warning box (yellow icon note)

Indicates that **Alpha (Experimental)** languages may fail to translate or return incomplete results, and should be used carefully.

Alpha (Experimental)

Experimental languages with a higher chance of errors depending on the model/workflow.

Note: Some languages appear as a single **“Global”** entry, while others are split into regional variants (e.g., German → Austria/Germany/Switzerland). This allows you to choose a **region-specific variant** when needed.

Bottom note (language add/remove behavior)

After making your selections and clicking **Setup**:

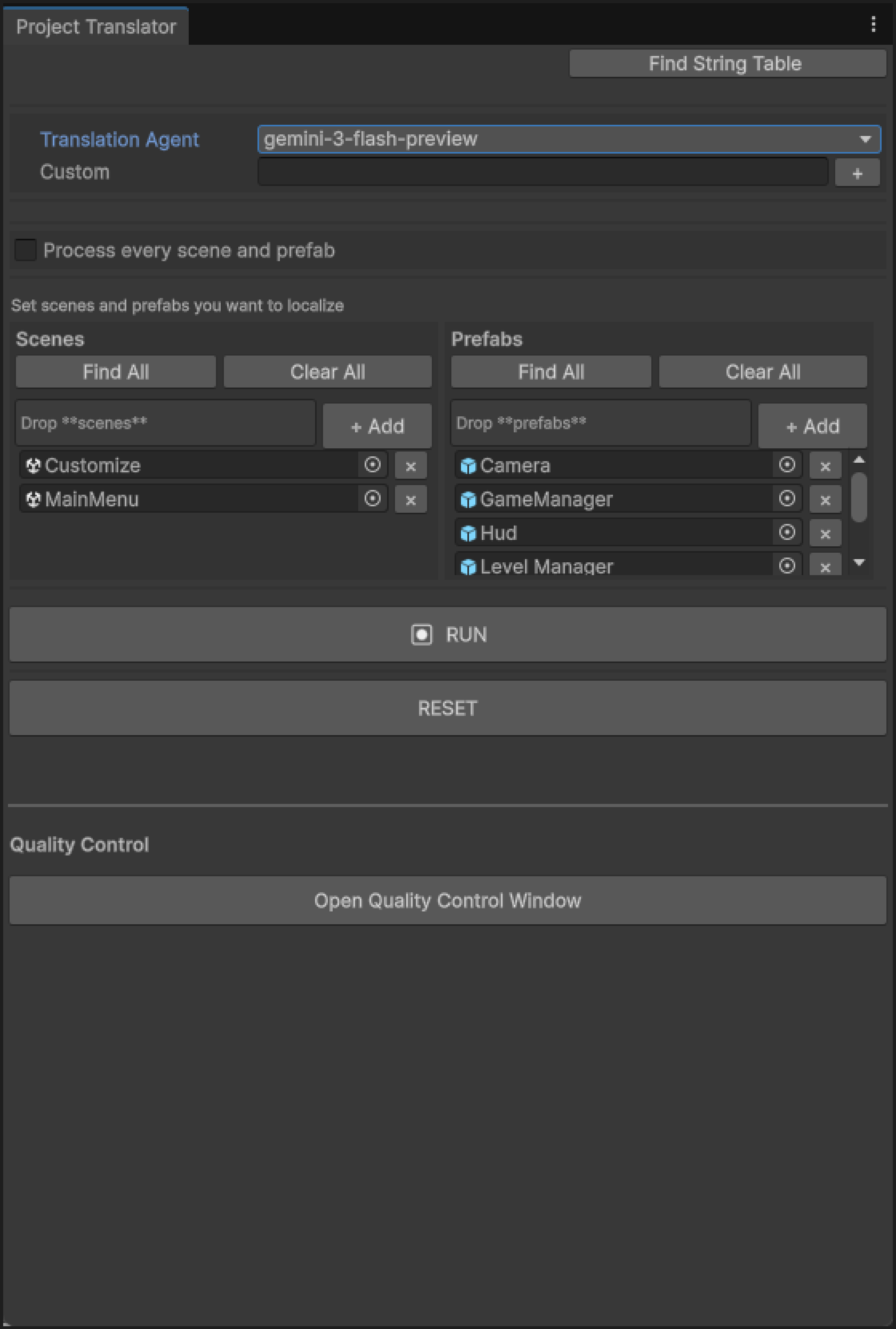
- Newly added languages are **added to your tables**.
- Languages you remove may be **deleted from the related tables/entries** (as noted in the documentation: “removed ones are deleted”).

Setup

Applies the selected target languages to your project: prepares the required table/entry fields and creates the **active language set** used by Translator Pro in all next steps.

When you select a new target language in **Language Selector** and click **Setup**, Translator Pro doesn’t only save the language list — it also **adds the required fields to your tables and automatically generates the initial translations** for that newly added language. This first pass is designed for fast onboarding and typically results in a **raw / word-to-word** style translation. To ensure accuracy, natural phrasing, and overall quality, it is strongly recommended to run **Quality Control (QC)** again after adding new languages.

This workflow has a major advantage: you can set up Translator Pro with **only a couple of languages first** (e.g., 2 languages), validate that the entire pipeline works correctly, and only then enable the remaining languages once everything is ready. This helps you **control costs** by avoiding unnecessary full-scale translation runs before your project and settings are finalized.



Translation Agent: gemini-3-flash-preview

□ Process every scene and prefab

What is the Project Translator (Text) window? (Short overview)

Project Translator (Text) is the main window used to scan the **scenes and prefabs in your project for text content** and translate them into the languages you select. From here you decide **which scenes and prefabs will be processed** and choose the **Translation Agent / model** used for the translation.

After the translation is completed, the window also provides a quick shortcut to the **Quality Control** window to further improve translation accuracy and consistency.

Fields and Buttons on the Screen (Line-by-Line)

Find String Table

Helps locate and connect the **localization String Table structure used in the project**.
This allows Translator Pro to quickly detect **where translation outputs should be written**.

Translation Agent (Model / Service Selection)

Translation Agent

Select the **model or service used for text translation** (for example, gemini-3-flash-preview).
This choice affects the **speed, quality, and cost profile** of the translation process.

Custom

Allows you to enter a **custom model or agent name** independently of the predefined list, or define custom configurations.

(+) Button

Adds and saves the **custom agent/model configuration** you entered in the Custom field.
In practice, this functions as **“add custom translation agent.”**

Scene & Prefab Selection (What content will be translated?)

Process every scene and prefab (Checkbox)

When enabled, **all scenes and prefabs in the project may be processed automatically**.
When disabled, **only the scenes and prefabs you manually select below will be processed**.

Scenes & Prefab Section

The area where you select **which** scene/prefab **in the project should be translated**.

Find All
Scans the project and adds all scene/prefab to the list.

Clear All
Clears the scene list.

Drop scenes area
Allows you to **drag and drop** scene/prefab into the list.

+ Add
Adds the selected scene to the list.

List rows
Each row represents a scene/prefab (for example: *Customize, MainMenu*).

Eye (👁) icon
Acts as a shortcut to **quickly locate, ping, or focus the asset in the project**.

X
Removes that scene/prefab from the list.

Execution / Reset

RUN (Checkbox + Large Button)

Scans the selected scenes and prefabs, collects all text content, and **starts the translation process using the selected agent/model**.

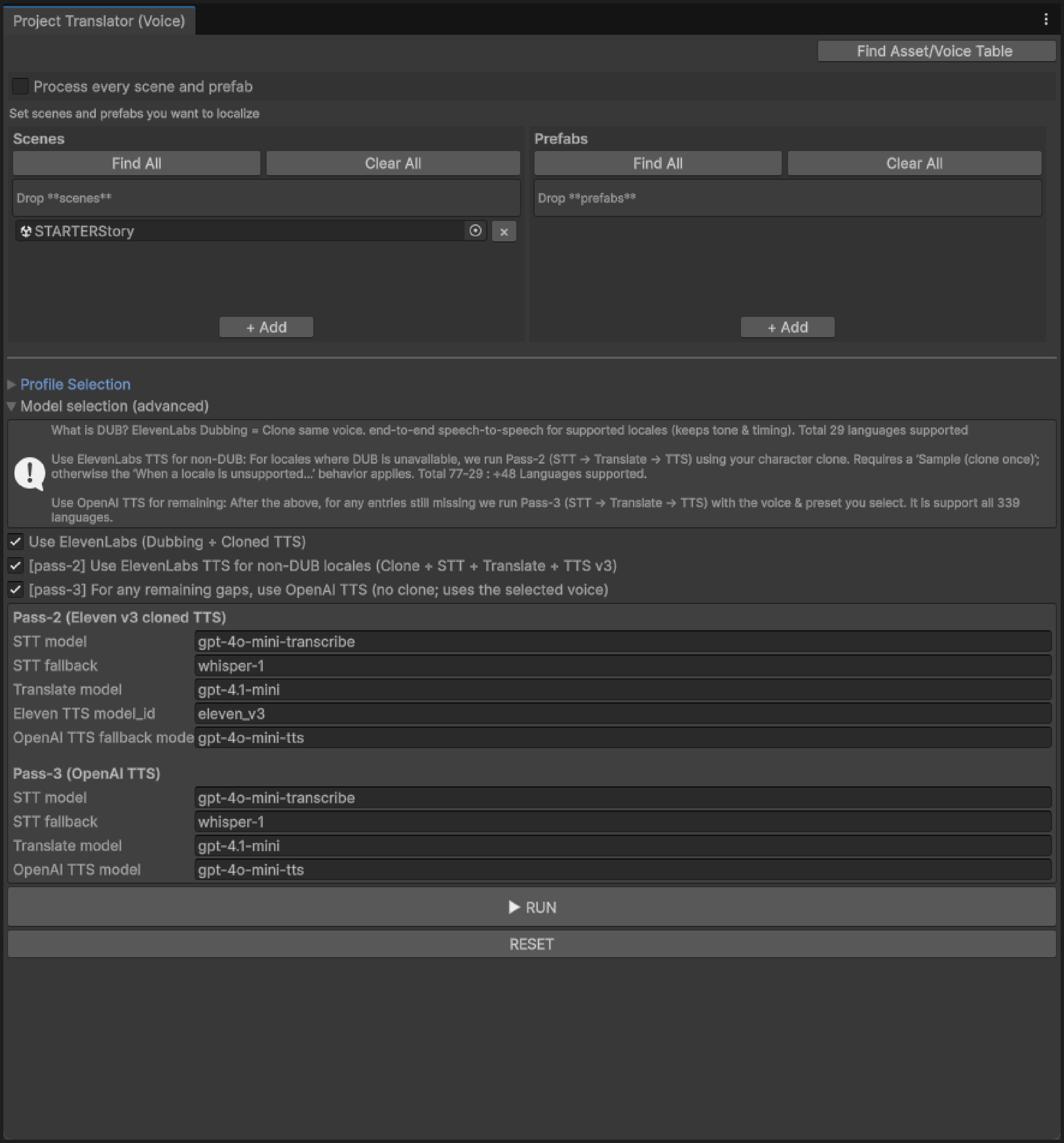
RESET

Resets the selections and settings in this window.
Useful for **clearing lists and temporary selections and returning to the default state**.

Quality Control Section

Open Quality Control Window

Quickly opens the **Quality Control window** after translation is complete to improve translation accuracy and consistency.



□ Process every scene and prefab

Set scenes and prefabs you want to localize

What is DUB? ElevenLabs Dubbing = Clone same voice. end-to-end speech-to-speech for supported locales (keeps tone & timing). Total 29 languages supported

Use ElevenLabs TTS for non-DUB: For locales where DUB is unavailable, we run Pass-2 (STT → Translate → TTS) using your character clone. Requires a 'Sample (clone once)'; otherwise the 'When a locale is unsupported...' behavior applies. Total 77-29 : +48 Languages supported.

Use OpenAI TTS for remaining: After the above, for any entries still missing we run Pass-3 (STT → Translate → TTS) with the voice & preset you select. It is support all 339 languages.

☑ Use ElevenLabs (Dubbing + Cloned TTS)

☑ [pass-2] Use ElevenLabs TTS for non-DUB locales (Clone + STT + Translate + TTS v3)

☑ [pass-3] For any remaining gaps, use OpenAI TTS (no clone; uses the selected voice)

Pass-2 (Eleven v3 cloned TTS)

STT model: gpt-4o-mini-transcribe
STT fallback: whisper-1
Translate model: gpt-4.1-mini
Eleven TTS model_id: eleven_v3
OpenAI TTS fallback mode: gpt-4o-mini-tts

Pass-3 (OpenAI TTS)

STT model: gpt-4o-mini-transcribe
STT fallback: whisper-1
Translate model: gpt-4.1-mini
OpenAI TTS model: gpt-4o-mini-tts

What is the Translation Quality Checker (QC) window? (Short overview)

Translation Quality Checker (QC) is the quality assurance center of Translator Pro. It automatically reviews generated translations to verify **meaning accuracy, consistency, and template/placeholder safety**.

From this window you can:

- Select which **QC steps** will run
- Choose **which models are used for each step**
- Optionally include additional models such as **Grok, Gemini, or DeepSeek** as a second or third review layer
- Apply suggested corrections to the translation table with a single click after QC is completed

---

Fields and Buttons on the Screen (Line-by-Line)

Quality Control Settings

This is the top section where the **general QC configuration** is managed.

Restart QC from scratch (delete previous results)

If enabled, all previously generated QC reports and results are deleted and the QC process starts **from a clean state**.
Useful when you want to **re-evaluate the project from scratch**.

Steps to run (Start button)

(QC steps to execute)

Here you select **which QC stages will run**.

Meaning *(mandatory)*

The main stage that checks whether the **translation is semantically correct**.

Prefilter Deletion

Removes **noisy, irrelevant, or low-quality candidates** using a pre-filtering approach.
This improves quality while **reducing processing workload**.

Dedupe

Detects and removes **duplicate or extremely similar entries**, helping maintain **consistency and cleaner datasets**.

Template Probe (Smart Strings)

Its main purpose is to **automatically detect entries that could be Smart Strings and convert them into the Smart String format**. During this conversion process, **unnecessary or outdated template structures are cleaned up and the Smart String structure is standardized**.

As a result:

- The **entire Smart String workflow becomes automated**, reducing manual effort.
- Cleaner entries reduce noise, which **lowers processing cost and speeds up the QC and translation pipeline**.
- Because context and structure are preserved more accurately, **translation quality reaches its maximum potential**.

Note: Template Probe produces better results when used with **stronger models and higher reasoning effort**. Running QC **multiple times in sequence** can also help detect Smart String candidates that may have been missed initially, increasing overall coverage with each pass.

Quality Control *(mandatory)*

The **final QC evaluation layer**, which aggregates the results and determines the final quality decisions.

Models (per step)

(Model selection for each QC step)

Here you define **which model is used at each QC stage**.
This allows you to optimize the pipeline—for example:

- using **fast/low-cost models** for Prefilter and Dedupe
- using **stronger reasoning models** for Meaning and final QC

Meaning / Translator Agent Model

Model Selection

Choose the model used for the **Meaning evaluation stage** (example: gpt-5.2).

Reasoning Effort

Controls the **reasoning intensity** of the model (example: none). Higher reasoning effort may improve quality in some scenarios but may increase **cost and processing time**.

Custom API Key (optional)

Allows you to use a **separate API key specifically for this step** (advanced usage).

Report & Apply

Generates the QC report for this step and prepares applicable corrections. Depending on the pipeline configuration, this may trigger **both reporting and applying suggestions**.

Pre-Filter Model

Pre-Filter Model

Select the model used during the **Prefilter stage** (example: gpt-4.1-mini).

Pre-Filter Reasoning Effort

Reasoning configuration for the Prefilter model.

Pre-Filter Custom API Key (optional)

Optional API key specifically for the Prefilter stage.

Last Prefilter Report

Displays the file path of the **most recently generated Prefilter report**.

Ping

Helps locate the report file **in the operating system file explorer**.

Apply

Applies the Prefilter results (cleanup and filtering effects) to the project.

Report Only

Generates the report **without applying any changes**.

Dedupe Model

Dedupe Model

Model used during the **duplicate detection and cleanup stage**
(example: gpt-4.1-mini).

Dedupe Reasoning Effort

Reasoning configuration for the deduplication step.

Dedupe Custom API Key (optional)

Optional API key for the Dedupe stage.

Last Dedupe Report + Ping / Apply

Displays the last dedupe report path and provides buttons to **locate or apply the report results**.

Report Only

Generates a report without applying the changes.

Template Probe Model

Template Probe Model

Model used to verify **smart strings**
(example: gpt-5.2).

Template Probe Reasoning Effort

Reasoning configuration for Template Probe (example: medium).

Template Probe Custom API Key (optional)

Optional API key specifically for this step.

Last Template Probe Report + Ping / Apply

Displays the most recent Template Probe report path and provides quick actions to **locate or apply results**.

Report Only

Generates only the report without applying changes.

Comparative QC (Optional) ***(Recommended)***

This section allows additional models to participate as **comparative reviewers** to strengthen QC results.

The goal is to **reduce blind spots of a single model** and reach more reliable decisions.

Enable Comparative QC ***(Optional – Recommended)***

When enabled, the selected additional models participate in QC evaluation.

Include Grok → Grok Model

Adds **Grok** to the QC comparison process
(example: grok-4).

Include Gemini → Gemini Model

Adds **Gemini** to the evaluation process
(example: gemini-3-pro-preview, gemini-2.5-pro, etc.).

Include DeepSeek → DeepSeek Model

Adds **DeepSeek** as an additional QC reviewer
(example: deepseek-chat).

Bottom Buttons (Execution and Applying Results)

Start Quality Control

Starts the QC process using the **selected steps and models**.

Apply Suggestions & Save

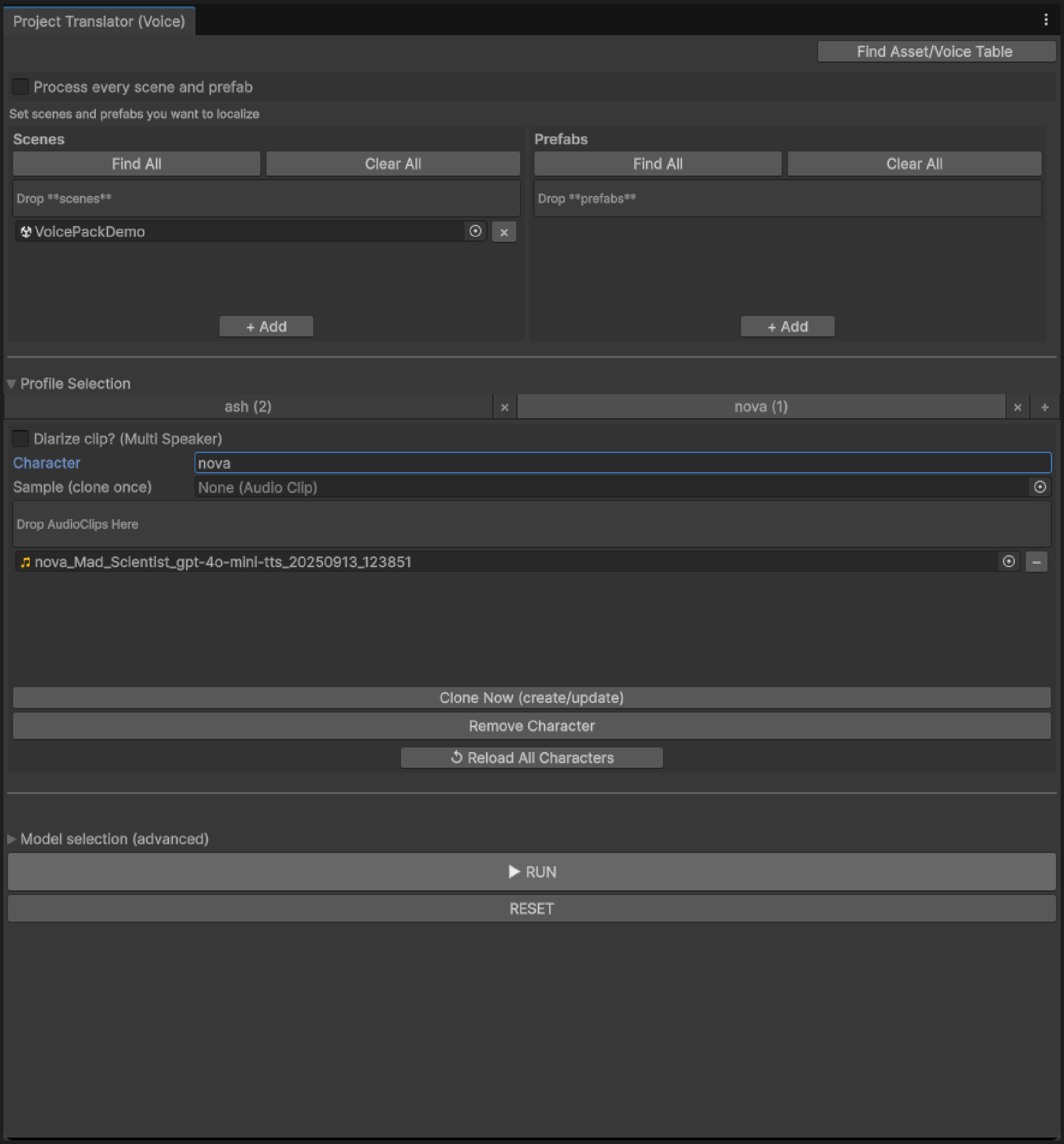
Applies the corrections suggested by QC to the translation table and **saves the results**.
This is typically the **final step after QC is completed**.

Send to Hugging Face

Exports QC data or dataset outputs for **external sharing or dataset publishing workflows**.

Explore the open dataset on Hugging Face

Acts as a link or gateway to **browse the open dataset on Hugging Face**, similar to a documentation or dataset reference page.



☐ Process every scene and prefab

Scenes: VoicePackDemo

☐ Diarize clip? (Multi Speaker)

Character: nova

Sample (clone once): None (Audio Clip)

nova_Mad_Scientist_gpt-4o-mini-tts_20250913_123851

Project Translator (Voice)

What is this window?

The **Project Translator (Voice)** window automatically localizes dialogue and voice content in your project into selected languages using **AI dubbing and voice localization**.

The system works **per character**, allowing you to assign voice clips to characters, optionally clone voices, or use predefined TTS voices.

The window also supports **Multi Speaker (Diarization)**, which can automatically detect and separate multiple speakers inside a single audio clip.

Scene and Prefab Selection

Process every scene and prefab

When enabled, all scenes and prefabs in the project will be processed automatically.

Scenes

Select the scenes that contain voice content.

Find All
Scans the project and adds all scenes automatically.

Clear All
Clears the scene list.

Drop scenes
Scenes can also be added via drag and drop.

Prefabs

Select prefabs that contain voice elements.

Find All
Scans and adds all prefabs in the project.

Clear All
Clears the prefab list.

Drop prefabs
Prefabs can be added via drag and drop.

Profile Selection

This section manages **character voice profiles**.

Each character appears as a separate **tab**.

Character

Defines the character name used for voice generation.

Sample (clone once)

Select a sample voice clip that represents the character.

This clip can be used to **create a voice clone** with ElevenLabs.

Diarize clip? (Multi Speaker)

When enabled, the system automatically **detects and separates multiple speakers** within a single audio clip.

This feature is particularly useful for:

- Dialogue recordings
- Podcast-style recordings
- Clips where multiple characters speak in the same audio file

---

Drop AudioClips Here

Drag and drop voice clips for the selected character here.

Clone Now (create/update)

Creates or updates the **voice clone** for the selected character using ElevenLabs.

Remove Character

Removes the selected character profile.

Reload All Characters

Reloads all character profiles.

Model Selection (Advanced)

Allows advanced users to manually configure the **STT, translation, and TTS models** used in the voice pipeline.

RUN

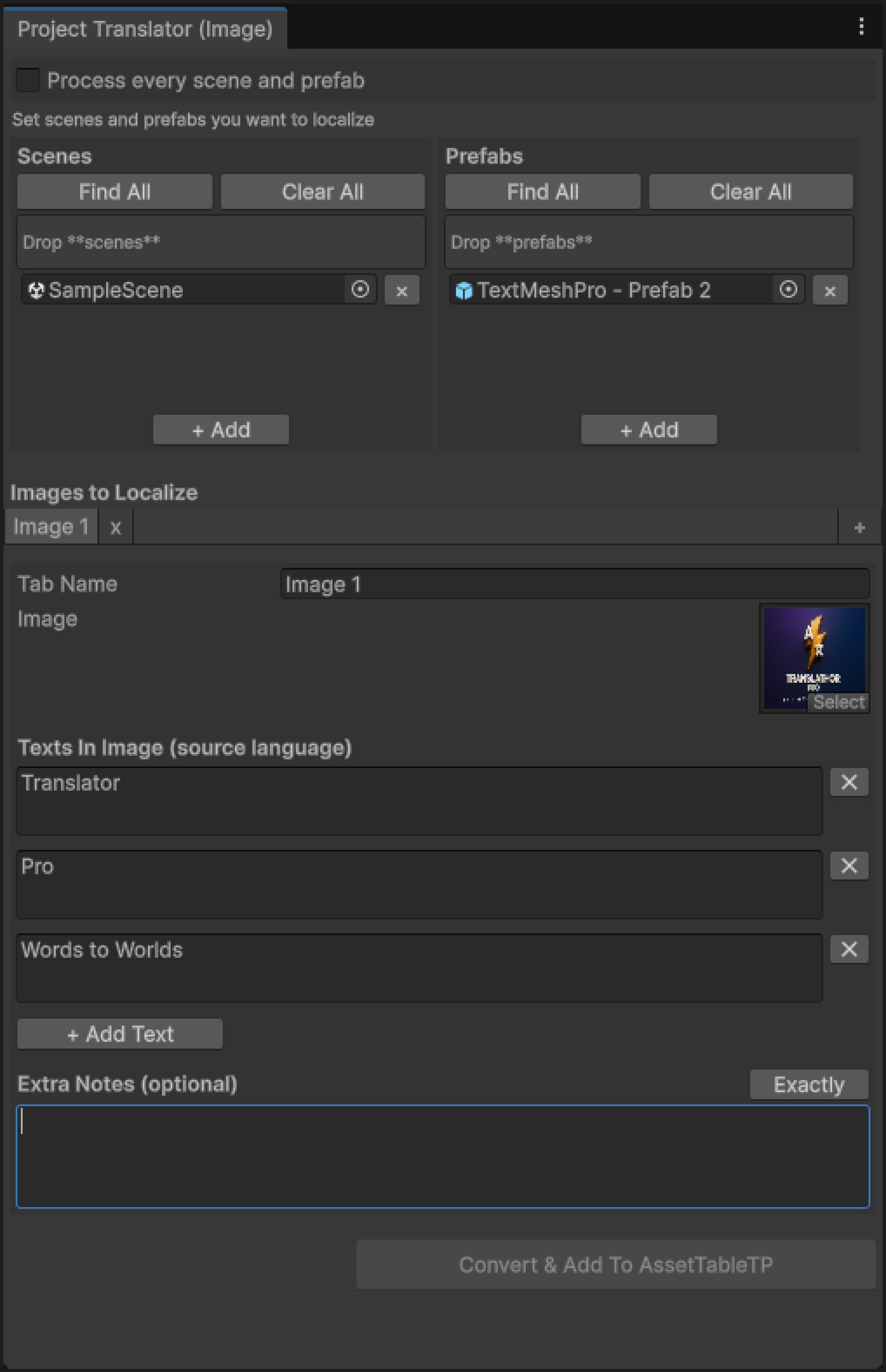
Starts the **voice localization pipeline** using the selected scenes, characters, and models.

RESET

Removes the Translator Pro voice localization components and generated table data from the project.
This clears the voice pipeline by removing the components that were added to scenes and prefabs by Translator Pro.

Notes:

- **Audio clip assets are not deleted.**
- **Only the Translator Pro components and generated table data are reset.**
- **This is useful when you want to restart the voice localization process from a clean state.**



□ Process every scene and prefab

Set scenes and prefabs you want to localize

Scenes

Drop **scenes**

SampleScene

Prefabs

Drop **prefabs**

TextMeshPro - Prefab 2

Images to Localize

Image 1

Tab Name: Image 1

Image

Texts In Image (source language)

Translator

Pro

Words to Worlds

Extra Notes (optional)

Project Translator (Image)

What is this window?

The **Project Translator (Image)** window provides a fully automated **image localization pipeline**.

It scans images inside selected scenes and prefabs, translates the specified text elements, generates localized versions of the image for each target language, and integrates them into the project.

The generated images are added to the **Asset Table**, allowing the correct localized image to be loaded automatically at runtime.

Scene and Prefab Selection

Process every scene and prefab

When enabled, **all scenes and prefabs** in the project will be processed automatically.

Scenes

Select the scenes that contain images you want to localize.

Find All
Scans the project and adds all scenes automatically.

Clear All
Clears the scene list.

Drop scenes
Scenes can also be added via drag and drop.

Prefabs

Select prefabs that contain images you want to localize.

Find All
Scans and adds all prefabs in the project.

Clear All
Clears the prefab list.

Drop prefabs
Prefabs can be added via drag and drop.

Images to Localize

This section defines the images that will be localized.

Each image can be added as a separate **tab**.

Tab Name

Defines the internal name used for this image entry.

Image

Select the image you want to localize.

Texts in Image (source language)

In this section you specify the **text elements that appear in the image**.

Example:

Translator
Pro
Words to Worlds

Only the texts listed here will be translated.

Important:

- Other visual elements of the image are **not modified**
- Only the specified text elements are translated
- A localized image is generated for **each target language**

+ Add Text

Adds a new text entry.

Extra Notes (optional)

Here you can provide **additional instructions to the model** for image generation.

Examples:

- keep text centered
- preserve the original font style
- keep the logo unchanged

Convert & Add To AssetTableTP

Starts the image localization pipeline.

This process:

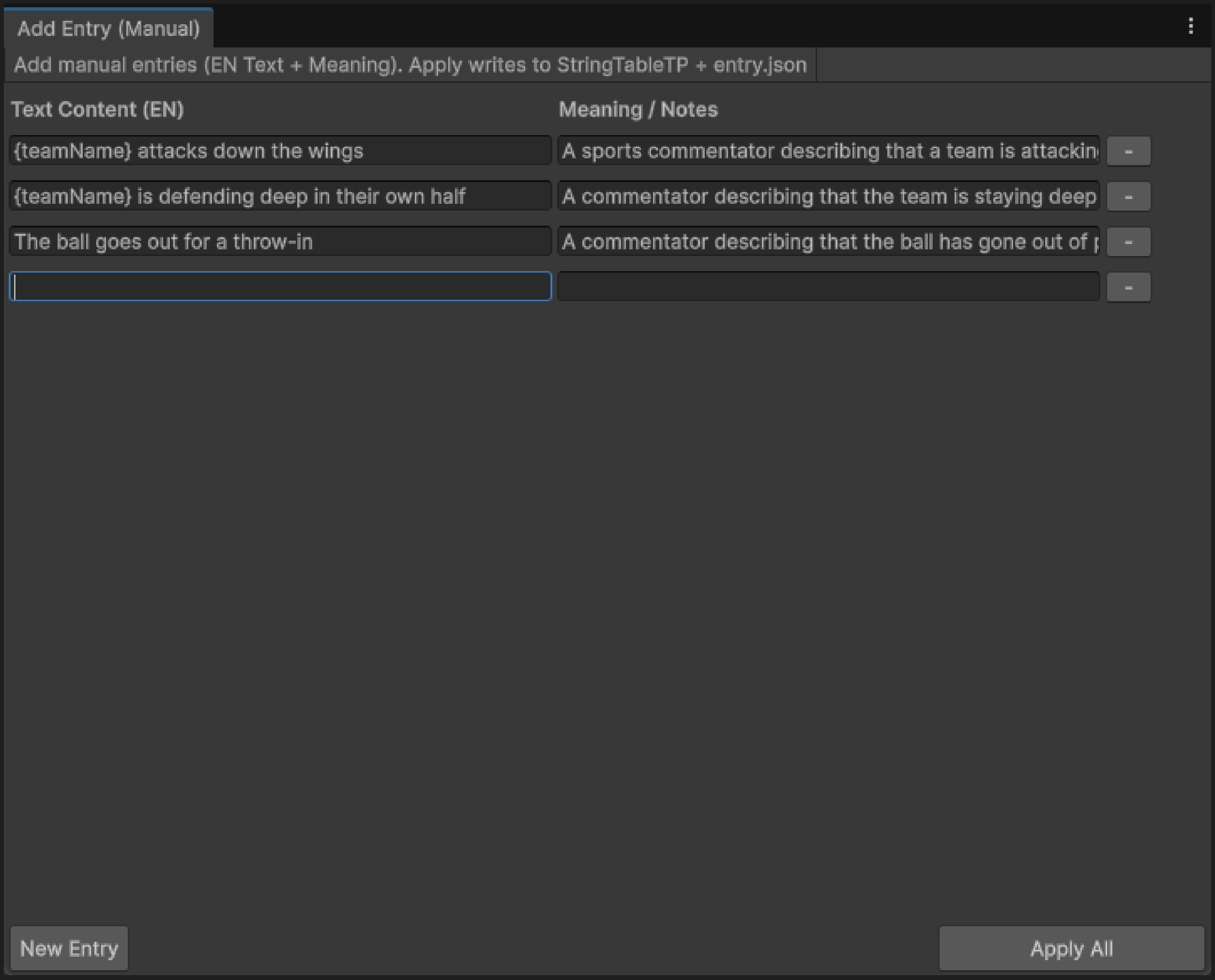
1. Analyzes the image
2. Translates the specified texts
3. Generates localized images
4. Adds them to the **AssetTableTP**

This allows the correct localized image to be automatically loaded at runtime.

Important Note

The pipeline requires **at least one target language** in the project.

If only the source language exists, the pipeline will not run because there are **no target languages to generate localized images for**.



Add Entry (Manual)

Add manual entries (EN Text + Meaning). Apply writes to StringTableTP + entry.json

Text Content (EN)	Meaning / Notes
{teamName} attacks down the wings	A sports commentator describing that a team is attackin
{teamName} is defending deep in their own half	A commentator describing that the team is staying deep
The ball goes out for a throw-in	A commentator describing that the ball has gone out of

Add Entry (Manual)

What is this window?

The **Add Entry (Manual)** window is used to **manually add localization entries** to your project. It is useful for texts that cannot be captured automatically through Record Mode.

Entries added through this window are written to both **StringTableTP** and the **entry.json** file, ensuring they are included in the full Translator Pro translation and quality control pipeline.

Text Content (EN)

This field contains the **source language text** (usually English).

You can use placeholders or Smart String formatting inside the text.

Example:

{teamName} attacks down the wings

These placeholders are preserved during translation and replaced with actual values at runtime.

Meaning / Notes

This field explains the **context or intended meaning** of the text.

Providing meaning is particularly important when:

- a sentence could have multiple interpretations
- gameplay context must be clarified
- the text belongs to commentary, dialogue, or narrative systems

The meaning field allows the AI translation system to generate **more accurate and context-aware translations**.

Example:

A commentator describing that a team is attacking down the wings. {teamName} is the team name placeholder.

Adding a New Entry

New Entry

Adds a new manual entry row.

Each row contains:

- Text Content (EN)
- Meaning / Notes

Apply All

Applies all entries to the project.

This action:

1. Adds the entries to **StringTableTP**
2. Updates the **entry.json** file

3. Includes the entries in the Translator Pro **translation and QC pipeline**

Smart String Support

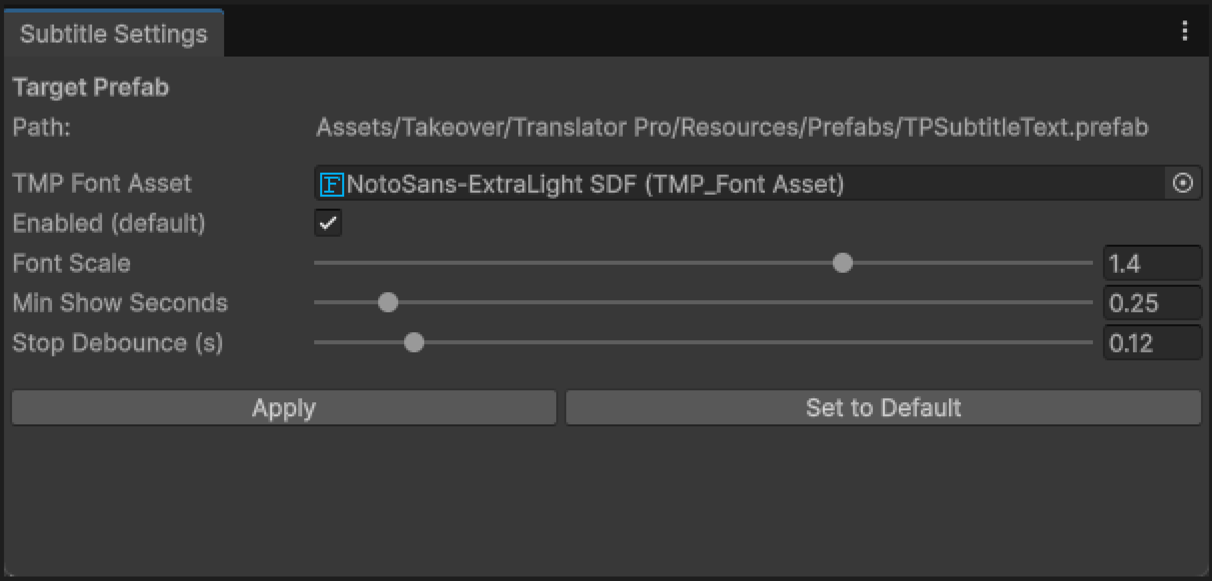
Manual entries support **Smart String formatting**.

Example:

{teamName} attacks down the wings

This allows:

- dynamic values to remain intact
- placeholders to stay safe during translation
- runtime systems to inject correct values automatically



Target Prefab

Path: Assets/Takeover/Translator Pro/Resources/Prefabs/TPSubtitleText.prefab

TMP Font Asset: NotoSans-ExtraLight SDF (TMP_Font Asset)

Enabled (default): ☑

Font Scale: 1.4

Min Show Seconds: 0.25

Stop Debounce (s): 0.12

Subtitle Settings

What is this window?

The **Subtitle Settings** window allows you to configure the **appearance and behavior of subtitles** generated during voice localization.

These settings are applied to the subtitle prefab used by Translator Pro and control how subtitles appear and how long they remain visible at runtime.

Target Prefab

Path

Displays the project path of the subtitle prefab used by the system.

This prefab is a **TextMeshPro-based UI component** responsible for rendering subtitles in the game.

TMP Font Asset

Specifies the **TextMeshPro font asset** used for subtitle text.

It is recommended to use a font that supports multiple languages (for example Noto Sans).

Enabled (default)

Determines whether subtitles are **enabled by default**.

If disabled, subtitles will not be generated automatically.

Font Scale

Controls the **visual size of subtitle text** on the screen.

This helps adjust readability across different screen sizes.

Min Show Seconds

Defines the **minimum duration** that a subtitle will remain visible on screen.

This ensures that even short dialogue lines remain readable.

Stop Debounce (s)

Defines a short delay after audio playback stops before the subtitle disappears.

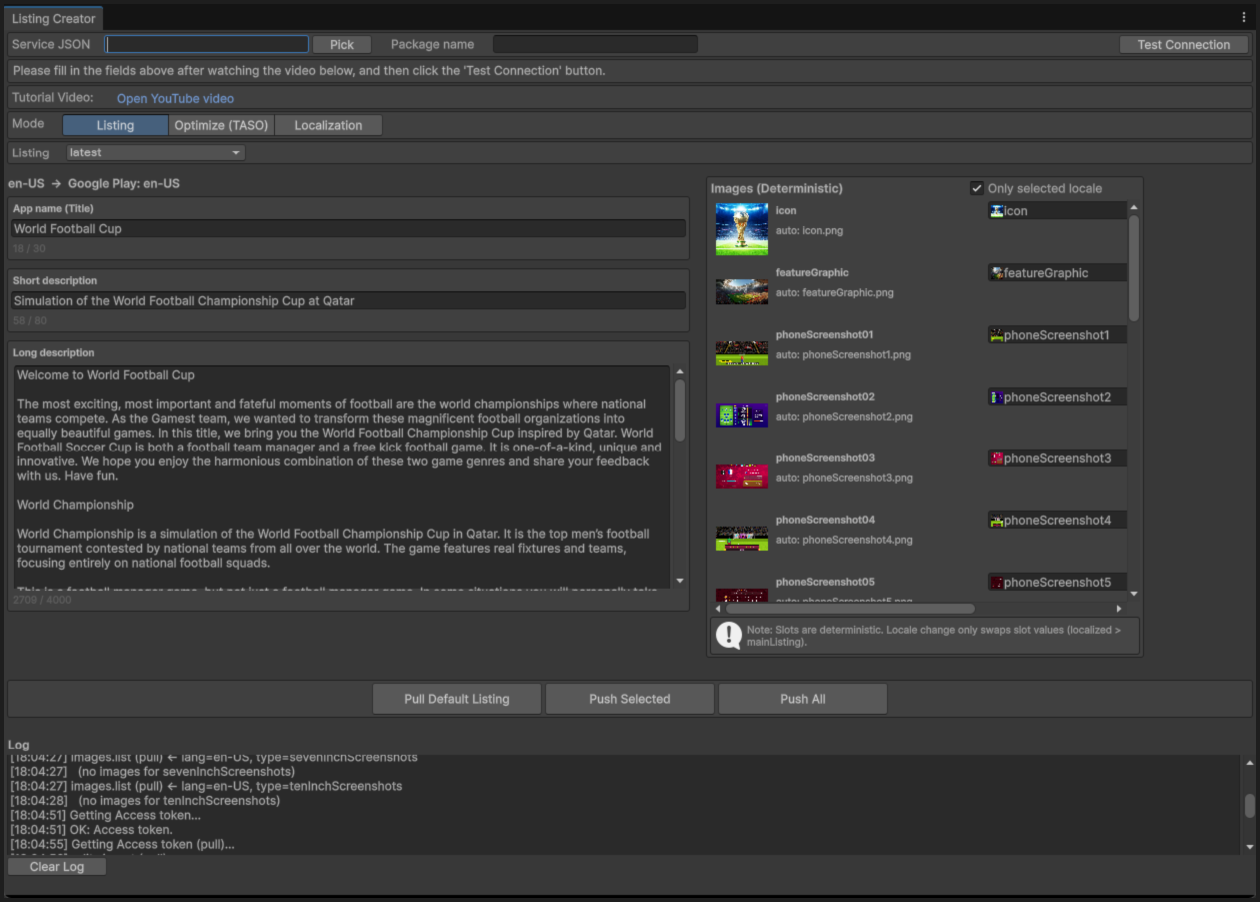
This makes subtitle timing feel more natural when dialogue ends.

Apply

Applies the current settings to the subtitle prefab.

Set to Default

Resets all subtitle settings back to their **default values**.



Please fill in the fields above after watching the video below, and then click the 'Test Connection' button.

Listing: latest

en-US → Google Play: en-US

App name (Title)

World Football Cup

18 / 30

Short description

Simulation of the World Football Championship Cup at Qatar

58 / 80

Long description

Welcome to World Football Cup

The most exciting, most important and fateful moments of football are the world championships where national teams compete. As the Gamest team, we wanted to transform these magnificent football organizations into equally beautiful games. In this title, we bring you the World Football Championship Cup inspired by Qatar. World Football Soccer Cup is both a football team manager and a free kick football game. It is one-of-a-kind, unique and innovative. We hope you enjoy the harmonious combination of these two game genres and share your feedback with us. Have fun.

World Championship

World Championship is a simulation of the World Football Championship Cup in Qatar. It is the top men’s football tournament contested by national teams from all over the world. The game features real fixtures and teams, focusing entirely on national football squads.

2709 / 4000

Images (Deterministic) ☑ Only selected locale

icon auto: icon.png	icon
featureGraphic auto: featureGraphic.png	featureGraphic
phoneScreenshot01 auto: phoneScreenshot1.png	phoneScreenshot1
phoneScreenshot02 auto: phoneScreenshot2.png	phoneScreenshot2
phoneScreenshot03 auto: phoneScreenshot3.png	phoneScreenshot3
phoneScreenshot04 auto: phoneScreenshot4.png	phoneScreenshot4
phoneScreenshot05	phoneScreenshot5

Note: Slots are deterministic. Locale change only swaps slot values (localized > mainListing).

```
[18:04:27] images.list (pull) ← lang=en-US, type=sevenInchScreenshots
[18:04:27]   (no images for sevenInchScreenshots)
[18:04:27] images.list (pull) ← lang=en-US, type=tenInchScreenshots
[18:04:28]   (no images for tenInchScreenshots)
[18:04:51] Getting Access token...
[18:04:51] OK: Access token.
[18:04:55] Getting Access token (pull)...
```

Listing Creator

What is this window?

Listing Creator allows you to manage your **Google Play store listing directly from Unity.**

With this tool you can edit:

- app title
- short description
- long description
- icon
- feature graphic
- screenshots

and **push them directly to Google Play**.

This system is designed for **fast store updates and multilingual ASO workflows**.

Top Settings

Service JSON

Select the **service account JSON file** used for Google Play Developer API access.

This file is:

- generated in Google Cloud
- required for Google Play API authentication

Use the **Pick** button to select the file.

Package name

The **Android package name** of the application.

Example:

com.company.gamename

This identifies your app in Google Play.

Test Connection

Tests the connection to the Google Play API.

If successful, the system confirms that the API connection works.

Tutorial Video

Opens the **YouTube tutorial video** explaining the setup process.

Mode Tabs

Listing

Used to edit the **main store listing content**.

Editable fields include:

- App name
- Short description

- Long description

Optimize (TASO)

Contains **Takeover ASO (TASO)** optimization tools.

This section helps:

- analyze store text
- generate alternative versions
- experiment with different listing variants

The goal is to **increase install conversion rates.**

Localization

Used to **automatically translate store listing text into multiple languages.**

This allows the same store listing to be published in many languages.

Listing

Listing Version

Select the **store listing version** you want to edit.

Usually:

latest

App name (Title)

The **application title** displayed in the Google Play store.

Character limits are defined by Google Play.

Short Description

The **short description** shown under the app title in the store.

This field should:

- attract user attention
- quickly explain the game concept.

Long Description

The detailed description of the application.

Typically includes:

- game features
- gameplay explanation
- content overview

Images (Deterministic)

This section manages **Google Play store images**.

Slots follow Google Play’s deterministic slot system.

Examples:

- icon
- featureGraphic
- phoneScreenshot01
- phoneScreenshot02
- phoneScreenshot03

Images can vary by locale.

Only selected locale

Edits images only for the **currently selected language**.

Pull Default Listing

Downloads the **current store listing data from Google Play**.

This allows editing the existing store listing inside Unity.

Push Selected

Uploads the **selected locale listing** to Google Play.

Push All

Uploads **all localized listings** to Google Play.

Log

Displays **API operation logs**.

Useful for debugging and connection diagnostics.

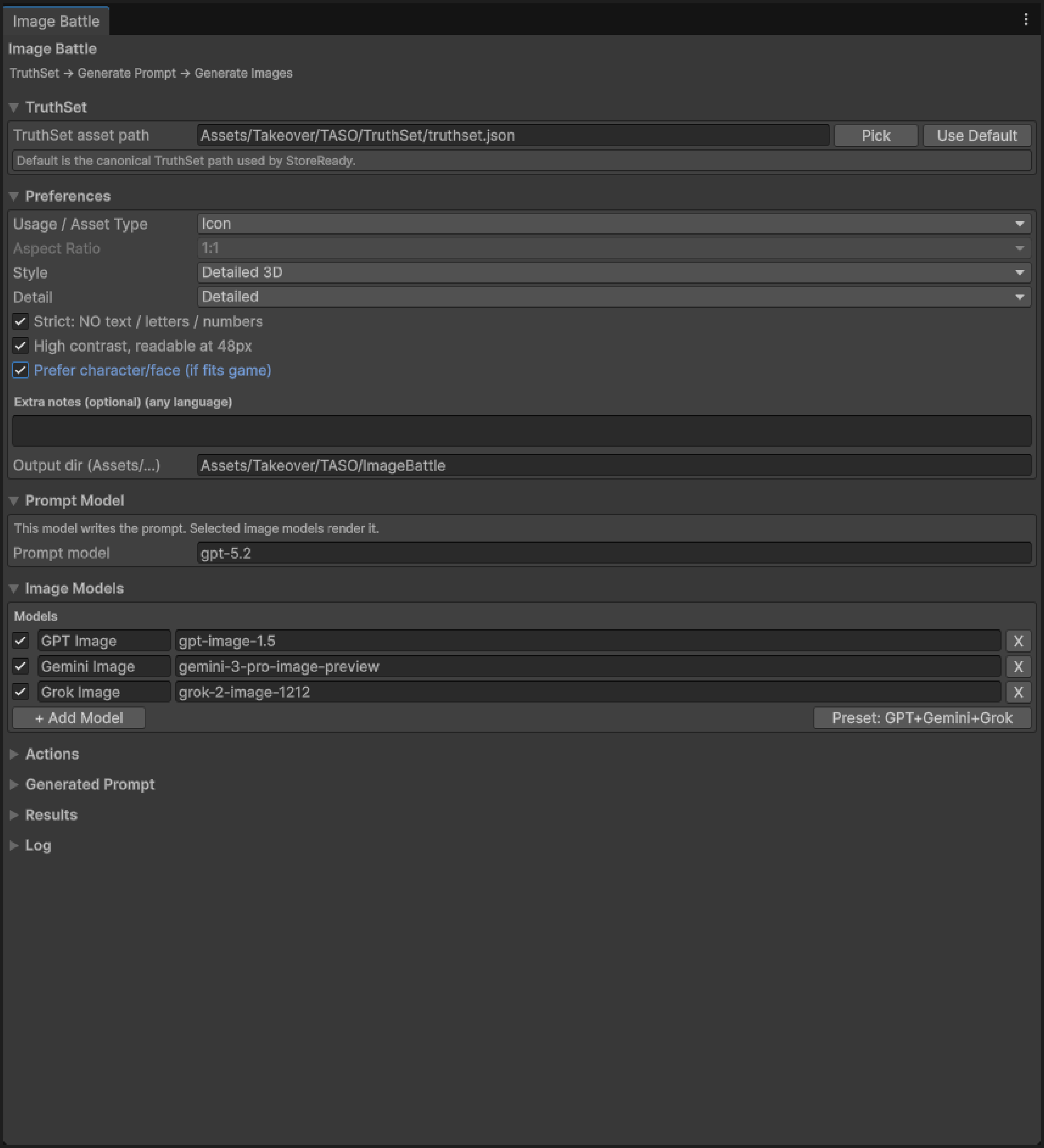


Image Battle

TruthSet → Generate Prompt → Generate Images

▼ TruthSet

TruthSet asset path: Assets/Takeover/TASO/TruthSet/truthset.json

Default is the canonical TruthSet path used by StoreReady.

▼ Preferences

Usage / Asset Type: Icon

Aspect Ratio: 1:1

Style: Detailed 3D

Detail: Detailed

☑ Strict: NO text / letters / numbers

☑ High contrast, readable at 48px

☑ Prefer character/face (if fits game)

Extra notes (optional) (any language)

Output dir (Assets/...): Assets/Takeover/TASO/ImageBattle

▼ Prompt Model

This model writes the prompt. Selected image models render it.

Prompt model: gpt-5.2

▼ Image Models

Models

☑	GPT Image	gpt-image-1.5	X
☑	Gemini Image	gemini-3-pro-image-preview	X
☑	Grok Image	grok-2-image-1212	X

▶ Actions

▶ Generated Prompt

▶ Results

▶ Log

Image Battle

What is this window?

Image Battle is an AI-powered tool designed to **generate store images** such as icons, feature graphics, and promotional visuals for your game.

The system works by:

1. generating an optimized prompt using your game data and preferences
2. sending that prompt to multiple AI image models
3. producing multiple image variations

This allows you to **compare results from different models and select the best visuals.**

The tool is primarily designed for **store icon creation, feature graphics, and ASO visual optimization.**

TruthSet

TruthSet asset path

Defines the **TruthSet data file** used by Image Battle.

TruthSet contains structured information about your game such as:

- genre
- theme
- characters
- visual style
- target audience

This information helps the AI generate **more accurate prompts.**

Pick

Select a different **TruthSet file**.

Use Default

Uses the default TruthSet configuration.

Preferences

This section defines the **visual style and constraints** for generated images.

Usage / Asset Type

Defines the **purpose of the image** being generated.

Examples:

- Icon
- Feature Graphic
- Screenshot
- Promotional Art

Aspect Ratio

Defines the **aspect ratio** of the generated image.

Examples:

1:1
16:9
9:16

Style

Defines the **art style** of the generated image.

Examples:

- Detailed 3D
- Cartoon
- Minimal
- Stylized

Detail

Controls the **level of visual detail**.

Examples:

- Simple
- Balanced

- Detailed

Strict: NO text / letters / numbers

Prevents AI models from generating **text or letters** in the image.

This is recommended for **store icons**.

High contrast, readable at 48px

Ensures the image remains **readable at small sizes**, such as app icons.

Prefer character/face (if fits game)

Encourages compositions focused on **characters or faces**, if appropriate for the game.

Extra notes (optional)

Allows you to provide **additional prompt instructions**.

Supports any language.

Output Directory

Defines where generated images will be saved.

Default location:

Assets/Takeover/TASO/ImageBattle

Prompt Model

Prompt model

The AI model used to **generate the image prompt.**

This model analyzes:

- TruthSet data
- preferences
- asset usage

and produces an optimized prompt for image generation.

Image Models

Defines the **AI image generation models** used to render images.
The same prompt is sent to multiple models.
Examples:

- GPT Image
- Gemini Image
- Grok Image

This allows you to **compare outputs from different models.**

Preset

Quickly select predefined **model combinations**.

Actions

Controls the image generation process.

Generated Prompt

Displays the **final prompt** generated by the system.

Results

Shows the **generated images** from each model.

Log

Displays system logs and generation details.

QA (COMMON QUESTIONS)

Q: Can I use Translator Pro without an OpenAI API key?
A: You can use Translator Pro in a very limited way without an OpenAI API key.

To unlock the full power of Translator Pro:

- You must enter your own OpenAI API key.
- This gives you the freedom to choose the latest and most accurate models.
- It also enables Quality Control, which checks the accuracy of every translation using scene context and AI.

Without an API key, you won’t be able to verify your translations, and the results may not match your game’s true meaning.
We strongly recommend entering your OpenAI API key to use Translator Pro properly.

Q: Localization table looks empty what should I do?
A: Try typing and clearing the search box in the Localization tab. Unity sometimes glitches and hides entries.

Q: I don’t want certain texts to be translated (e.g., brand names or counters). How?
A: No worries. Translator Pro automatically detects many elements that should not be translated, such as **proper names, brand names, and similar identifiers**, and skips them during localization.

If there is a specific text you want to guarantee will never be translated, simply add the **DontTranslate.cs** component to that GameObject. Translator Pro will completely exclude it from the localization setup.

Q: What happens if I use a weak model?
A: You'll likely get incomplete or wrong translations, especially if your entries are long or context-heavy.

Q: Can I translate my existing CSV translation table with Translator Pro?

A: Not really necessary anymore.

Translator Pro automatically scans your project and captures all localization entries directly from your game. Because of this, importing an external CSV or Excel translation table is usually redundant.

Technically, you *can* still import your existing table and press Setup in the language selection panel. Translator Pro will scan the table entries and translate them into the languages you choose.

However, this approach has an important limitation:
When translations come from a manually prepared CSV table, they tend to be word-level, similar to traditional localization workflows. These often miss context and can lead to awkward or incorrect results in many languages.

Translator Pro is designed to work differently.
By analyzing entries directly from the project and processing them through its context-aware pipeline, it produces translations that are much closer to natural, native-level language quality rather than simple word-by-word conversions.

For that reason, the recommended workflow is simple:
Let Translator Pro detect and translate the entries automatically, instead of relying on an old CSV localization table.