QuickTranslate User Guide

Version 2.0.0 | February 2026 | LocaNext Project

> *"Translate Smarter, Not Harder"*

Table of Contents

1. Introduction

1.1 What is QuickTranslate?

QuickTranslate is a desktop application for finding translations of Korean text by matching against game stringtables. It searches through thousands of localized strings across 17 languages to find existing translations instantly.

Core Function

QuickTranslate matches Korean source text (StrOrigin) against the stringtable database and returns all available translations. Instead of manually searching through XML files, you can:

- Find translations for hundreds of Korean strings in seconds
- Look up any StringID to see all language versions
- Reverse-lookup: find a StringID from text in any language
- Export results to Excel for review and handoff

1.2 Who is it for?

Role	Use Case
Localization Coordinators	Find existing translations for reuse
QA Testers	Verify translation consistency across languages
Translators	Look up reference translations
Developers	Find StringIDs from in-game text

1.3 Key Benefits

Benefit	Description
Speed	Process hundreds of strings in seconds
Accuracy	Multiple matching strategies for different needs
Completeness	Access all 17 languages at once
Flexibility	Excel and XML input/output support
Offline	Works entirely on local data

2. Installation

2.1 System Requirements

Requirement	Specification
Operating System	Windows 10 / Windows 11
Perforce Access	Sync access to stringtable folders
Drive	F: drive mapped (or custom path configured)
Python	3.11+ (portable version only)

2.2 Installation Methods

2.2.1 Setup Installer (Recommended)

The installer provides the easiest setup experience with automatic configuration.

Steps:

1. Download `QuickTranslate_vX.X.X_Setup.exe` from the releases page
2. Run the installer
3. Select installation drive (C:, D:, F:, etc.)
4. Choose whether to create a desktop shortcut
5. Click **Install**
6. Application launches automatically after installation

>
> **PRO TIP**
>
> Install on the same drive as your Perforce workspace for faster file access.

2.2.2 Portable Version

The portable version requires no installation and can run from any location.

Steps:

1. Download `QuickTranslate_vX.X.X_Portable.zip`
2. Extract to any folder
3. Run `QuickTranslate.exe`

> **i**
> **NOTE**
>
> The portable version is ideal for USB drives or environments with installation restrictions.

2.3 First-Time Configuration

On first launch, QuickTranslate creates `settings.json` with default paths:

```
{
  "loc_folder": "F:\\perforce\\cd\\mainline\\resource\\GameData\\stringtable\
  "export_folder": "F:\\perforce\\cd\\mainline\\resource\\GameData\\stringtal
}
```

To change paths:

1. Close QuickTranslate
2. Edit `settings.json` in the application folder
3. Update paths to match your Perforce workspace
4. Restart QuickTranslate

---

3. Quick Start

Get up and running in 5 minutes with these quick tutorials.

3.1 Your First Translation Lookup

Goal: Find translations for a list of Korean strings

Step 1: Prepare Your Input

Create an Excel file with Korean text in Column A:

A
안녕하세요
감사합니다
시작하기

Save as `input.xlsx`

Step 2: Configure QuickTranslate

1. Launch QuickTranslate
2. Set **Format**: `Excel`
3. Set **Mode**: `File`
4. Set **Match Type**: `Substring Match`

Step 3: Select Input File

1. Click **Browse** next to Source
2. Navigate to `input.xlsx`
3. Click **Open**

Step 4: Generate Output

1. Verify Branch is correct (mainline/cd_lambda)
2. Click **Generate**
3. Wait for processing (progress bar shows status)

Step 5: View Results

Output saved to: `Output/QuickTranslate_YYYYMMDD_HHMMSS.xlsx`

KOR (Input)	ENG	FRE	GER	...
안녕하세요	Hello	Bonjour	Hallo	...
감사합니다	Thank you	Merci	Danke	...

>
> **SUCCESS!**
>
> You've completed your first translation lookup!

3.2 Quick StringID Lookup

Goal: Find all translations for a specific StringID

1. Enter StringID in the **Quick Actions** section (e.g., `UI_MainMenu_Title`)
2. Click **Lookup**
3. Output: Excel file with all 16 language translations

3.3 Reverse Lookup

Goal: Find StringID from English (or any language) text

1. Create a text file with one string per line:

```
Start Game
Options
Exit
```

2. In **Quick Actions** → **Reverse**, click **Browse**
3. Select your text file
4. Click **Find All**
5. Output shows: Input | KOR | ENG | FRE | ...

4. Core Concepts

Understanding these concepts will help you use QuickTranslate effectively.

4.1 StringID, StrOrigin, and Translations

4.1.1 StringID

A **StringID** is the unique identifier for each localized string.

```
Examples:
  UI_MainMenu_Title_001
  Quest_Chapter1_Dialog_042
  Item_Weapon_Sword_Name
```

- Links source text to all translations
- Format varies by category (UI, Quest, Item, etc.)
- Same StringID = same meaning across all languages

4.1.2 StrOrigin

StrOrigin is the original Korean source text.

```
<LocStr StringId="UI_Button_OK" StrOrigin="확인" Str="OK" />
```

- Used for matching in substring and strict modes
- For SCRIPT strings: StrOrigin = raw Korean dialogue

- For UI strings: StrOrigin may be formatted/tagged

4.1.3 Translations

Translations are stored in `languagedata_*.xml` files:

```
languagedata_eng.xml  → English translations
languagedata_fre.xml  → French translations
languagedata_ger.xml  → German translations
... (17 languages total)
```

4.2 Branches

QuickTranslate supports multiple development branches:

Branch	Description
mainline	Main development branch (default)
cd_lambda	Alternative branch for specific builds

> ⚠
> **WARNING**
>
> Selecting a different branch reloads all data. This may take 1-2 minutes on first load.

Cross-branch comparison:

- Set **Source Branch** and **Target Branch** differently
- Compare mainline vs cd_lambda translations

4.3 SCRIPT Categories

SCRIPT categories are special: their StrOrigin equals the raw Korean text.

Category	Content Type
Sequencer	Cutscene dialogue
AIDialog	NPC AI dialogue
QuestDialog	Quest conversations
NarrationDialog	Narrator/voiceover

>
> **PRO TIP**
>
> For SCRIPT strings, use **STRINGID-ONLY** match type. StrOrigin matching isn't needed since StrOrigin = KOR text.

4.4 File Structure

LOC Folder

Contains all translations by language:

```
loc/
├── languagedata_eng.xml
├── languagedata_fre.xml
```

```
├── languagedata_ger.xml
└── ... (17 files)
```

Export Folder

Contains categorized source files:

```
export__/
├── Sequencer/
│   ├── Chapter1/*.loc.xml
│   └── Chapter2/*.loc.xml
├── UI/
├── Items/
└── Quest/
```

---

5. Features Deep Dive

5.1 Format Modes

5.1.1 Excel Format

Best for: Batch processing, handoff sheets

Input requirements by match type:

Match Type	Column A	Column B	Column C
Substring	Korean text	-	-
StringID-Only	StringID	StrOrigin	Correction
Strict	StringID	StrOrigin	Correction

Supported extensions: `.xlsx`, `.xls`

5.1.2 XML Format

Best for: Direct processing of game data files

Input format:

```
<LocStr StringId="UI_001" StrOrigin="한국어 텍스트" Str="Translation" />
```

Case-insensitive attributes: StringId, StringID, stringid all work

Supported extensions: `.xml`, `.loc.xml`

5.2 Input Modes

5.2.1 File Mode (Single File)

- Process one file at a time
- Select specific file via Browse dialog
- Good for focused tasks

5.2.2 Folder Mode (Recursive)

- Process all matching files in folder and subfolders
- Automatically finds all `.xlsx` / `.xls` or `.xml` files
- Good for batch processing entire directories

> **i**
> **NOTE**
>
> Folder mode shows progress: "Parsing XML files... 1/N"

5.3 Match Types

5.3.1 Substring Match (Original)

How it works:

```
Input: "시작"
Searches: All StrOrigin values
Finds: Any string containing "시작"
  - "게임 시작하기" ✓
  - "시작 버튼" ✓
  - "새로 시작" ✓
```

Pros	Cons
Flexible	May return multiple matches
Finds partial matches	Less precise
Works with fragments	Needs review of results

Best for: Finding strings when you only have partial Korean text

5.3.2 StringID-Only (SCRIPT)

How it works:

```
1. Reads StringIDs from input
2. Filters to SCRIPT categories ONLY
3. Returns translations for matching StringIDs
```

Automatic filtering includes:

- Sequencer
- AIDialog
- QuestDialog
- NarrationDialog

Status output: "SCRIPT filter: 150 kept, 23 skipped"

Best for: Processing Sequencer/Dialog corrections where StrOrigin = raw Korean

5.3.3 StringID + StrOrigin (STRICT)

How it works:

```
Input: StringID="UI_001", StrOrigin="확인"
Matches: ONLY if both StringID AND StrOrigin match
```

Requires:

- XML input format (has both StringID and StrOrigin)
- Target folder for comparison

Pros	Cons
Most precise	Requires more input data
Handles reused StringIDs	XML format only
No false positives	Needs target folder

Best for: Verifying corrections with 100% certainty

5.3.4 Special Key Match

How it works:

- Custom composite key from multiple fields
- Currently defaults to StringID matching
- Future: Configurable key patterns

Best for: Advanced matching scenarios

5.4 Quick Actions

5.4.1 StringID Lookup

Input: Single StringID (e.g., `UI_MainMenu_Title`)

Output: `StringID_<ID>_YYYYMMDD_HHMMSS.xlsx`

StringID	ENG	FRE	GER	SPA	...
UI_MainMenu_Title	Main Menu	Menu Principal	Hauptmenü	Menú Principal	...

5.4.2 Reverse Lookup

Input: Text file with strings in ANY language

```
Start Game
Optionen
Commencer
```

Output: `ReverseLookup_YYYYMMDD_HHMMSS.xlsx`

Input	KOR	ENG	FRE	GER
Start Game	게임 시작	Start Game	Démarrer	Spiel starten
Optionen	옵션	Options	Options	Optionen
Commencer	시작	Start	Commencer	Starten

Detection: Shows which language each input was detected as

5.5 ToSubmit Integration

Checkbox: "ToSubmit Folder Integration"

Location: `<app_folder>/ToSubmit/`

Purpose: Process correction files staged for submission

Expected structure: Files with StrOrigin, Correction, StringID columns

6. Workflows

6.1 Find Translations for Korean Text

Scenario: You have a list of Korean strings and need all language translations

Steps:

1. **Create Excel** with Korean text in Column A
2. **Launch** QuickTranslate
3. **Set** Format: Excel, Mode: File, Match Type: Substring Match
4. **Browse** to your Excel file
5. **Verify** Branch selection
6. **Click** Generate
7. **Open** output Excel

6.2 Process SCRIPT Corrections

Scenario: You have XML corrections for Sequencer dialogue

Steps:

1. Set Format: **XML**
2. Set Mode: **File**

3. Set Match Type: **StringID-Only (SCRIPT)**
4. Browse to your `.xml` file
5. Click **Generate**

Output: Only SCRIPT categories included

Status: "SCRIPT filter: X kept, Y skipped"

6.3 Verify Translations with Strict Matching

Scenario: Ensure corrections match exact StringID + StrOrigin pair

Steps:

1. Set Format: **XML**
2. Set Mode: **File**
3. Set Match Type: **StringID + StrOrigin (STRICT)**
4. Browse to **Source** XML file
5. Browse to **Target** folder (comparison data)
6. Click **Generate**

Output: Only verified matches included

6.4 Batch Process a Folder

Scenario: Process all XML files in a directory

Steps:

1. Set Format: **XML**
2. Set Mode: **Folder** (recursive)
3. Set Match Type: **Substring Match**

4. Browse to folder containing XML files
5. Click **Generate**

Progress: "Scanning folder... 1/N" → "Parsing XML files... 1/M"

6.5 Find StringID from English Text

Scenario: You have English text, need to find the StringID

Steps:

1. Create text file:

```
Main Menu
Start Game
Options
```

2. In Quick Actions → Reverse, click **Browse**
3. Select your text file
4. Click **Find All**

Result: Shows StringID and all translations

7. Output Files

All output files are saved to: `<app_folder>/Output/`

7.1 Standard Translation Output

Filename: `QuickTranslate_YYYYMMDD_HHMMSS.xlsx`

Columns:

Column	Content
A	KOR (Input)
B	ENG
C	FRE
D	GER
E	SPA
...	(17 languages total)

Multiple matches: Formatted as numbered list

```
1. Translation option 1
2. Translation option 2
3. Translation option 3
```

7.2 StringID Lookup Output

Filename: `StringID_<ID>_YYYYMMDD_HHMMSS.xlsx`

Format: Single row with StringID and all translations

7.3 Reverse Lookup Output

Filename: `ReverseLookup_YYYYMMDD_HHMMSS.xlsx`

Special values:

- `NOT FOUND` - No matching StringID found
- `NO TRANSLATION` - Translation empty or contains Korean (untranslated)

7.4 Folder Translation Output

Filename: `FolderTranslate_<foldername>_YYYYMMDD_HHMMSS.xlsx`

Format: One sheet per language

Columns per sheet:

StrOrigin	English	[Language]	StringID

8. Troubleshooting

8.1 Common Issues

"LOC folder not found"

Cause: Perforce not synced or path incorrect

Solution:

1. Ensure F: drive is mapped to Perforce
2. Run `p4 sync` on stringtable folder
3. Or update `settings.json` with correct path

"Sequencer folder not found"

Cause: Export folder not synced

Solution:

```
p4 sync //depot/cd/mainline/resource/GameData/stringtable/export__/...
```

"No input data found"

Cause: Empty file or wrong format

Solution:

- **Excel:** Ensure Korean text is in Column A (no header row by default)
- **XML:** Ensure file has `<LocStr>` elements

"StringID not found"

Cause: StringID doesn't exist in current branch

Solution:

1. Check spelling (case-sensitive)
2. Try different branch (mainline vs cd_lambda)
3. Verify StringID exists in source files

"Strict mode requires Target folder"

Cause: Strict matching needs comparison folder

Solution: Browse and select Target folder with XML files to compare against

Progress stuck at "Indexing Sequencer..."

Cause: Large number of files to scan (first run)

Solution: Wait patiently - first run builds index (1-2 minutes). Subsequent runs use cache.

8.2 Performance Tips

Scenario	Tip
First run slow	Normal - building index. Subsequent runs faster
Branch change slow	Triggers re-index. Consider staying on one branch
Large files	XML format processes faster than Excel
Memory usage	Close other applications if processing 1000+ strings

8.3 Data Caching

What's cached:

- StrOrigin index
- Translation lookup
- Category mapping

Cache invalidated when:

- Branch selection changes
- Application restarts

9. Reference

9.1 Supported Languages

Code	Display	Language
`kor`	KOR	Korean (Source)
`eng`	ENG	English
`fre`	FRE	French
`ger`	GER	German
`spa`	SPA	Spanish
`por`	POR	Portuguese
`ita`	ITA	Italian
`rus`	RUS	Russian
`tur`	TUR	Turkish
`pol`	POL	Polish
`zho-cn`	ZHO-CN	Chinese (Simplified)
`zho-tw`	ZHO-TW	Chinese (Traditional)
`jpn`	JPN	Japanese
`tha`	THA	Thai

`vie`	VIE	Vietnamese
`ind`	IND	Indonesian
`msa`	MSA	Malay

9.2 Supported File Formats

Format	Extensions	Library
Excel	`.xlsx`, `.xls`	openpyxl
XML	`.xml`, `.loc.xml`	lxml
Text	`.txt`	built-in

9.3 SCRIPT Categories

Category	Description
Sequencer	Cutscene/cinematic dialogue
AIDialog	NPC AI-triggered dialogue
QuestDialog	Quest conversation text
NarrationDialog	Narrator/voiceover text

9.4 Default Paths

Path	Default Value
LOC Folder	`F:\perforce\cd\mainline\resource\GameData\stringtable\loc`
Export Folder	`F:\perforce\cd\mainline\resource\GameData\stringtable\export__`
Output Folder	`<app_folder>\Output`
ToSubmit Folder	`<app_folder>\ToSubmit`
Settings File	`<app_folder>\settings.json`

9.5 Command Line Options

```
python main.py              # Launch GUI
python main.py --verbose    # Launch with debug logging
python main.py --version    # Show version
python main.py --help       # Show help
```

9.6 Keyboard Shortcuts

Shortcut	Action
`Alt+G`	Generate
`Alt+C`	Clear fields
`Alt+X`	Exit
`Enter`	Activate focused button

9.7 Settings File Format

```
{
  "loc_folder": "F:\\perforce\\cd\\mainline\\resource\\GameData\\stringtable\
  "export_folder": "F:\\perforce\\cd\\mainline\\resource\\GameData\\stringtal
}
```

> ⚠
> **WARNING**
>
> Use double backslashes (`\\`) in JSON paths on Windows.

10. Appendix

10.1 Glossary

Term	Definition
StringID	Unique identifier for a localized string
StrOrigin	Original Korean source text
Str	Translated text for a language
LocStr	XML element containing string data
SCRIPT	Categories with raw Korean StrOrigin
LOC folder	Contains `languagedata_*.xml` files
Export folder	Contains categorized `.loc.xml` files
Substring match	Find text contained within StrOrigin
Strict match	Match requiring both StringID and StrOrigin

10.2 XML Element Structure

```
<LocStr
    StringId="UI_Button_001"
    StrOrigin="확인 버튼"
    Str="OK Button"
```

```
    Category="UI"
/>
```

Attribute	Required	Description
StringId	Yes	Unique identifier
StrOrigin	No	Korean source text
Str	Yes	Translation text
Category	No	String category

10.3 Changelog

Version 2.0.0 (February 2026)

New Features:

- Added StringID-Only match type for SCRIPT strings
- Added Strict match type (StringID + StrOrigin)
- Added Special Key match type
- Added Folder mode (recursive processing)
- Added XML format input support
- Added Reverse Lookup feature
- Added branch selection (mainline/cd_lambda)
- Added ToSubmit folder integration
- Redesigned GUI with spacious 850x750 layout

Improvements:

- Modular codebase (main.py + core/ + gui/ + utils/)

- Better XML parsing with lxml recovery mode
- Progress bar and detailed status updates
- Improved error messages

Technical:

- Migrated from monolith to modular structure
- Added GitHub Actions CI/CD workflow
- PyInstaller + Inno Setup for distribution

Version 1.0.0 (Initial Release)

- Basic substring matching
- Excel input/output
- StringID lookup
- Multi-language support (17 languages)

10.4 Support

Issues & Feedback:

- GitHub Issues: LocalizationTools Repository

Documentation:

- This User Guide: `docs/USER_GUIDE.md`
- API Reference: See source code docstrings

QuickTranslate | Version 2.0.0 | LocaNext Project

Translate Smarter, Not Harder