LanguageDataExporter

User Guide

Version 1.0.16

Language XML to Categorized Excel Converter

with VRS-based Story Ordering

Table of Contents

1. Quick Start

30-Second Workflow

1. Double-click `LanguageDataExporter.exe`

2. Click **Analyze Categories** to see distribution
3. Click **Generate Language Excels** to create files
4. Find output in `GeneratedExcel/` folder

What It Does

Feature	Description	Output
Language Export	Convert XML to categorized Excel	`LanguageData_{LANG}.xlsx`
Word Count Report	LQA scheduling metrics	`WordCountReport.xlsx`
VRS Ordering	Chronological story order	Sorted STORY rows
Two-Tier Clustering	STORY + GAME_DATA categories	Color-coded cells

2. Installation

Requirements

Requirement	Details
OS	Windows 10/11
Disk Space	~50 MB
Network	Access to game data folders
Drive	Perforce sync on D:, E:, or F:

Installation Steps

1. **Download** - Get `LanguageDataExporter-Setup.exe` from GitHub Releases
2. **Run Installer** - Double-click and follow the prompts
3. **Configure Drive Letter**

> **Drive Configuration**
>
> On first launch, select your Perforce drive letter:
>
> `F:` (Default) - Most common
> `D:` or `E:` - Alternative drives
>
> This sets paths to LOC, EXPORT, and VRS folders.

4. **Launch** - Double-click `LanguageDataExporter.exe`

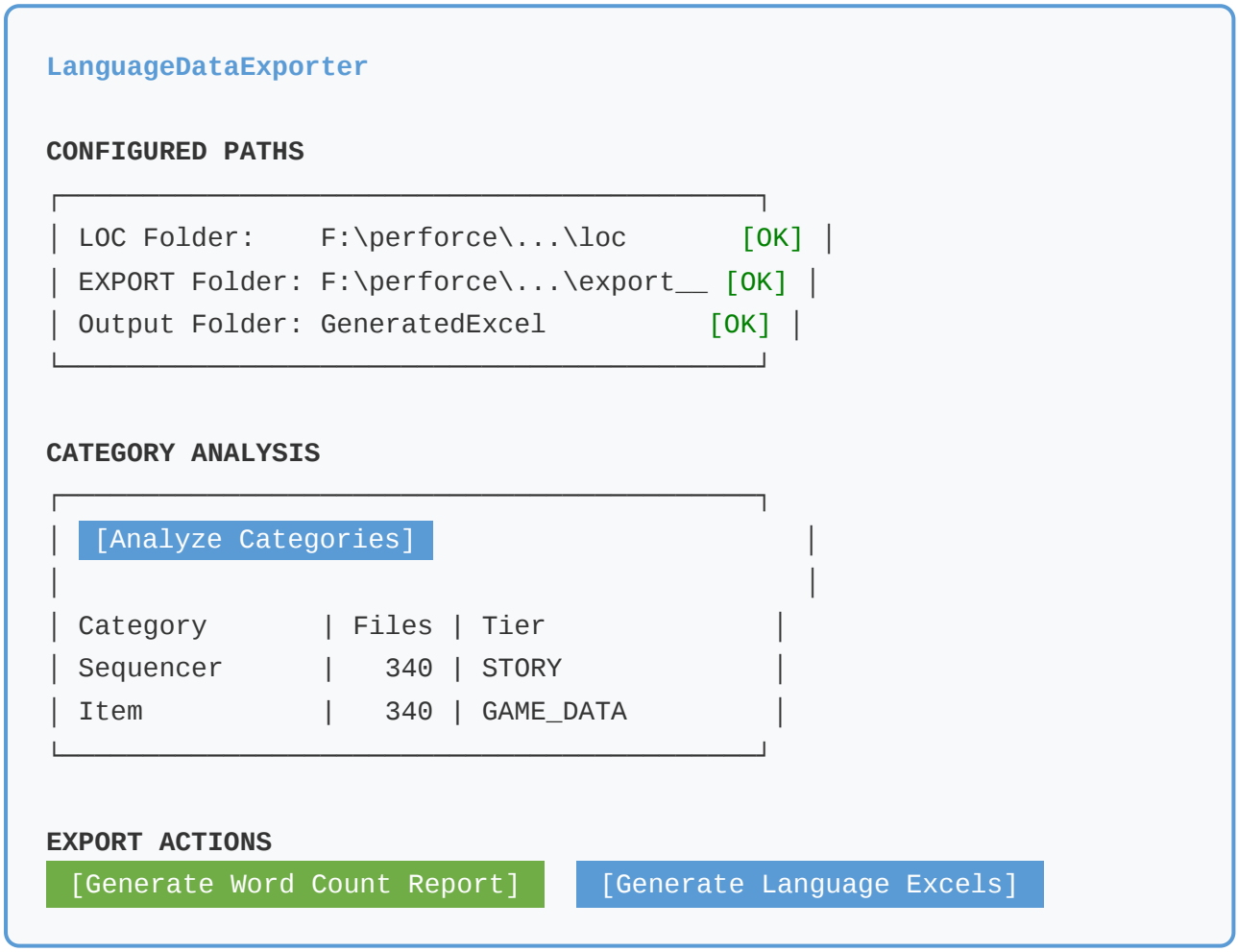
Folder Structure

```
LanguageDataExporter/
├── LanguageDataExporter.exe ← Main application
├── settings.json ← Your drive configuration
├── category_clusters.json ← Category colors/keywords
├── GeneratedExcel/ ← Output folder
│   ├── LanguageData_ENG.xlsx
│   ├── LanguageData_FRE.xlsx
│   ├── WordCountReport.xlsx
│   └── _Summary.xlsx
└── _internal/ ← Python runtime
```

3. GUI Mode

Launch by double-clicking `LanguageDataExporter.exe`

Interface Layout



GUI Actions

Button	What It Does	Output
Analyze Categories	Scans EXPORT folder, shows distribution	TreeView updated

Generate Word Count Report	Creates LQA metrics report	`WordCountReport.xlsx`
Generate Language Excels	Creates all language files	`LanguageData_*.xlsx`

4. CLI Mode

Basic Commands

```
# Run with GUI (default)
python main.py

# Run in CLI mode
python main.py --cli

# Process specific languages
python main.py --cli --lang eng,fre,ger

# Generate word count report
python main.py --cli --word-count

# Preview without writing files
python main.py --cli --dry-run

# Show category distribution
python main.py --list-categories
```

CLI Arguments Reference

Argument	Description	Example
`--cli`	Run in command-line mode	`--cli`
`--lang`	Process specific languages	`--lang eng,fre`
`--word-count`	Include word count report	`--word-count`
`--word-count-only`	Only generate word count report	`--word-count-only`
`--dry-run`	Preview without writing	`--dry-run`

`--list-categories`	Show category distribution	`--list-categories`
`--output`	Custom output folder	`--output D:\Out`
`-v`	Enable debug logging	`-v`

5. Category System

> **Two-Tier Architecture**
>
> LanguageDataExporter uses a two-tier category system:
> **TIER 1: STORY** - VRS-ordered chronological content
> **TIER 2: GAME_DATA** - Keyword-based categories

TIER 1: STORY Categories

STORY content is sorted chronologically using VoiceRecordingSheet ordering.

Category	Color	Source Folder	Description
Sequencer	Light Orange	`Sequencer/`	Story cutscenes, major moments
AIDialog	Light Green	`Dialog/AIDialog/`	NPC ambient conversation
QuestDialog	Light Green	`Dialog/QuestDialog/`	Quest dialogue trees
NarrationDialog	Light Green	`Dialog/NarrationDialog/`	Tutorial text, narration

TIER 2: GAME_DATA Categories

GAME_DATA uses a two-phase matching algorithm:
Phase 1: Priority Keywords (checked FIRST!)
Phase 2: Standard Patterns (folder + keywords)

Priority Keywords (Override Everything!)

> **Priority keywords in the filename OVERRIDE folder location!**
> Example: `KnowledgeInfo_Item.xml` in Knowledge/ folder → **Item** (not Knowledge!)

Priority	Keyword	Category	Note
1	`gimmick`	**Gimmick**	HIGHEST - wins over all
2	`item`	**Item**	Overrides Knowledge/Quest
3	`quest`	**Quest**	Overrides folder matching
4	`skill`	**Skill**	
5	`character`	**Character**	
6	`faction`	**Faction**	
7	`region`	**Region**	

Standard Patterns (Phase 2)

Category	Color	Folders	Keywords
Item	Light Purple	`LookAt/`, `PatternDescription/`	`weapon`, `armor`
Quest	Light Purple	`Quest/`	`schedule_`
Character	Light Peach	`Character/`, `Npc/`	`monster`, `animal`
Knowledge	Light Purple	`Knowledge/`	-
UI	Light Teal	`Ui/`	`localstringinfo`, `symboltext`

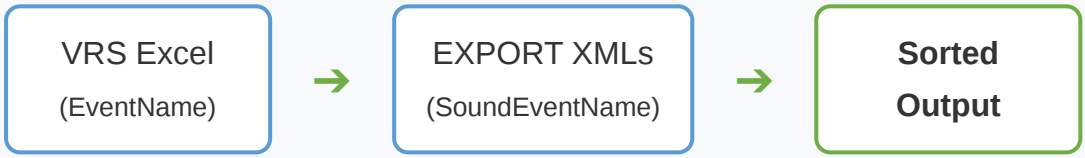
System_Misc	Light Grey	(default)	-

6. VRS Ordering

What is VRS?

VoiceRecordingSheet (VRS) is the master Excel file containing all voiced lines in chronological story order. LanguageDataExporter uses VRS to sort STORY content so LQA reviewers see dialogue in the order players experience it.

How It Works



Step	Action	Result
1	Load VoiceRecordingSheet.xlsx	Read EventName from Column W
2	Scan EXPORT XMLs	Extract SoundEventName attribute
3	Match StringID to EventName	Build ordering map
4	Sort STORY entries	Chronological story order!

Result: STORY strings appear in Excel in **chronological story order**. LQA reviewers see content as players experience it.

7. Word Count Reports

Report Purpose

Use Case	How It Helps
Schedule Work	Estimate LQA time based on word counts
Track Progress	Compare counts across languages
Find Untranslated	Identify strings still containing Korean

Counting Method

Language Type	Method	Languages
European/SEA	Word count	ENG, FRE, GER, SPA, POR, ITA, RUS, TUR, POL, THA, VIE, IND, MSA
CJK	Character count	JPN, ZHO-CN, ZHO-TW

> **Untranslated Detection**
>
> A string is marked **untranslated** if the translation still contains Korean characters (Unicode U+AC00-U+D7A3).
>
> Example: `"Hello 안녕"` → Marked as untranslated

8. Output Files

Language Excel Files

Filename: `LanguageData_{LANG}.xlsx`

Column	Width	Description
StrOrigin	45	Korean source text
Str	45	Translated text
StringID	15	Unique identifier
English	45	English reference (EU languages only)
Category	20	Color-coded category

> **Note:** CJK languages (JPN, ZHO-CN, ZHO-TW) don't include the English column.

9. Troubleshooting

Issue	Cause	Solution
Path NOT FOUND	Wrong drive letter	Edit `settings.json` or run `drive_replacer.py`
No language files	LOC folder empty	Check Perforce sync status
VRS not loaded	Missing VRS folder	Verify VRS path in settings
Empty output	No .loc.xml files	Check EXPORT folder exists
Wrong category	Priority keyword conflict	Check filename for keywords

Debug Mode

Run with `-v` flag for detailed logging:
`python main.py --cli -v`

LanguageDataExporter • Version 1.0.16 • January 2025
GitHub Repository