

A Translation Handoff That Preserves Meaning

Package context, terminology and protected variables before
integrating translated text

01

02

03

How to use this guide

Small product teams and editors handing interface or help text to a translator.

Work through thirty original interface strings, a deliberately faulty English return, a narrow Python checker and a full reconciliation answer. Build a handoff that separates technical checks from language and product review.

- Freeze and identify the source version.
- Explain message contexts and product terms.
- Check keys and protected token counts.
- Track corrections without claiming linguistic acceptance from a technical pass.

Scope and limitations

- All buyer prose and rehearsal messages are English; this is not a translated product.
- The checker handles only flat JSON and the documented simple token grammar.
- Plural behavior, actual rendering and target-language quality require separate review.

Published by KitForma. Edition 1.0. Sources checked 1 October 2026. People and businesses in worked examples are fictional. This is an independent publication, not endorsed by the platforms mentioned.

Copyright 2026 KitForma. Usage, support and refund terms are included in LICENSE.txt in the purchased package.

Contents

How to use this guide	2
Contents	3
Send context, not a bag of sentences	4
Sources and tools	6

01

Send context, not a bag of sentences

A translation handoff is a description of how language behaves in a product. A file containing English sentences is only one part of that description. The translator also needs to know who reads each sentence, when it appears, what action it names, and which pieces the application replaces at runtime. Without this context, a fluent translation can still describe the wrong action or break the interface that displays it.

This workbook uses a fictional file organizer called Harbor Notes. It has thirty original interface strings covering project selection, file import, preview, saving and recovery. The exercise does not claim to represent a released product or an existing translation. Its returned file is a deliberately altered English test file, clearly labelled as a technical rehearsal. That choice lets you practice key and placeholder checks without pretending that an unreviewed foreign-language text is ready for customers.

Your deliverable is a small package: a frozen source file, a context register, a limited glossary, instructions for protected tokens, a returned file, and a reconciliation report. Keep these artifacts separate. The source establishes what was requested. The context explains meaning. The return records what came back. The report identifies what needs correction or human review. Overwriting the source with the return destroys the comparison that makes the process useful.

Before sending real work, name the target language and regional variant with the responsible reviewer. Do not infer those choices from a customer's country or a team member's name. Also agree who can approve linguistic decisions and who owns application behavior. This English-only workbook does not supply a multilingual release or claim that

a token validator can replace a translator. It gives you a reproducible way to make the translator's work easier to interpret and safer to integrate.

Sources and tools

[S1] W3C Internationalization Quick Tips

<https://www.w3.org/International/quicktips/>

Checked: 2026-10-01

[S2] Unicode Standard Annex 15: Normalization Forms

<https://www.unicode.org/reports/tr15/>

Checked: 2026-10-01

Unicode Inspector

Optional character inspection after checking current tool availability and privacy behavior; not required for the supplied local checker.

<https://kitforma.com/en/unicode-inspector/>

Inside the complete guide

Freeze a source version with stable identifiers

Write context a translator can act on

Use a small glossary with decision ownership

Protect placeholders without freezing grammar

Preserve text while investigating Unicode differences

Run bounded technical checks

Reconcile corrections without losing translator decisions

Capstone: repair the thirty-string rehearsal

Answer key and a repeatable boundary