

Documents Across Alphabets

Separate glyphs, direction, wrapping, copy and search in one small test set

01

02

03

SAMPLE EDITION

ENGLISH / 1.0 / 2026

How to use this guide

Authors exporting short multilingual text and checking its technical preservation.

Six original text cards and reproducible Unicode checks help diagnose independent font, direction, normalization and export problems without claiming universal language support.

- Preserve an authoritative UTF-8 source.
- Separate logical text from visual ordering.
- Check named normalization behavior without erasing distinctions.
- Build a reader-specific visual, copy and search matrix.

Scope and limitations

- The English book contains short multilingual technical fixtures, not certified translations.
- No third-party fonts or universal font/reader compatibility are supplied.
- Source checks do not replace rendering, PDF accessibility or qualified language 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
Separate six questions that look like one	4
Sources and tools	6

01

Separate six questions that look like one

The fictional Atlas Desk prepares short information cards for a mixed-language workshop. Its coordinator has English, Turkish, Arabic, Hebrew, Chinese, and Japanese sample strings, each paired with a small reference code. The task is to export a document in which the text remains readable, ordered sensibly, searchable where promised, and recoverable through copy and paste. A screenshot that looks right answers only part of that task. This workbook turns one vague claim, multilingual support, into several separate checks.

The six questions are whether the characters are preserved, whether the selected fonts contain usable glyphs, whether shaping is appropriate, whether direction and punctuation behave correctly, whether lines wrap without losing content, and whether copy or search preserves the intended text. These properties can fail independently. An image of correctly shaped Arabic may look good while containing no searchable text. A PDF may preserve the Unicode string yet render missing-glyph boxes. Record each property separately instead of awarding a global pass after one visual glance.

The supplied cards are short original technical fixtures, not a professionally localized document or a promise of fluent translation. The Arabic and Hebrew labels are simple document-related words, and the CJK examples are brief labels. Their role is to expose rendering behavior. For a real public document, qualified language review must examine meaning, tone, names, and local conventions. Correct glyphs do not establish that a sentence says what you intend, just as a correct translation does not prove that an export preserved it.

Begin with the UTF-8 JSON reference and the HTML card sheet. The JSON preserves logical text values, while the HTML declares language and direction for presentation. A standard-library checker verifies the fixture's structure and Unicode examples. It does not render fonts or certify a PDF. Your capstone will use an actual export and reader to complete those checks. Keeping source truth, presentation, and reader behavior distinct gives you a way to diagnose failures without rewriting the original text merely to make one screen look acceptable.

Sources and tools

[S1] Unicode Standard Annex 9: Bidirectional Algorithm

<https://www.unicode.org/reports/tr9/>

Checked: 2026-10-01

[S2] Unicode Standard Annex 15: normalization forms

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

Checked: 2026-10-01

[S3] W3C internationalization quick tips

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

Checked: 2026-10-01

Inside the complete guide

Keep the source in a stable text representation

Check font coverage before blaming the text

Distinguish logical order from visual order

Check shaping and punctuation with a readable reference

Test wrapping without changing the message

Use normalization as a named comparison policy

Export, reopen, copy and search

Build a small, honest compatibility matrix

Capstone and answer key