

# UniWorld – Global Unicode Showcase (TEST OUTPUT)

This document is a **Unicode stress-test and demo** for UniWorld.

It is intentionally long and visually dense. Treat it as both: - A **presentation** of what UniWorld enables. - A **test fixture** for rendering, fonts, and text handling across scripts.

---

## 1. What UniWorld Enables

- **Correct text behavior for every script:**
- Bidirectional layout for **Arabic, Hebrew, mixed RTL/LTR**.
- Reliable grapheme clusters for **emoji, Indic conjuncts, and combining marks**.
- Dictionary-based line breaking for **Thai, Lao, Khmer, Myanmar** (no-space scripts).
- Accurate display width for **CJK full-width** text and terminals.
- **Single, testable core:**
- Implemented in Rust with full conformance to:
  - **UAX #9** – Bidirectional Algorithm.
  - **UAX #14** – Line Breaking.
  - **UAX #29** – Text Segmentation (grapheme, word, sentence).
  - **UAX #15** – Normalization.
- **Bindings:**
- Python (uniworld), JavaScript/WASM, C, Go – same behavior, same tests.

Use this file to: - Inspect how your editor, renderer, or terminal handles complex Unicode.  
- Verify that UniWorld-based tools respect grapheme clusters, widths, and breaks. - Copy-paste samples into your own test harnesses.

---

## 2. Latin Scripts and Accents

Regular ASCII Latin:

- The quick brown fox jumps over the lazy dog.

Latin with diacritics and composed vs decomposed forms:

- Composed: é, ö, ñ, å, ç, û, ß, Ű
- Decomposed (base + combining marks):
- e + COMBINING ACUTE: é
- o + COMBINING DIAERESIS: ö
- n + COMBINING TILDE: ñ

NFKC/NFKD-sensitive examples (ligatures, compatibility forms):

- Ligatures: ff, fi, fl, ffi, ff1 (should normalize to ff, fi, fl, ffi, ff1).
- Fractions: ½ vs 1/2, ⅓ vs 1/3.

These samples exercise: - **Normalization** (NFC/NFD/NFKC/NFKD). - **Grapheme clustering** for combining marks.

---

### 3. Greek, Cyrillic, and Case Mapping

Greek:

- Lowercase: α β γ δ ε ζ η θ ι κ λ μ ν ξ ο π ρ σ τ υ φ χ ψ ω
- Uppercase: Α Β Γ Δ Ε Ζ Η Θ Ι Κ Λ Μ Ν Ξ Ο Π Ρ Σ Τ Υ Φ Χ Ψ Ω
- Final sigma test word: ΟΔΥΣΣΕΥΣ → lowercase should end with final sigma.

Cyrillic (Russian phrase):

- Привет, мир! Это тест юникода.  
(Hello, world! This is a Unicode test.)

These samples exercise: - **Case mapping** (upper/lower/title). - **Locale-neutral casing** and **Greek final sigma**.

---

### 4. Hebrew and Arabic – Bidirectional Text

Hebrew:

- Plain: שלום עולָם (shalom olam – “peace, world / hello, world”)

- With punctuation and numbers: שלום (שלום), 123, סוף.

Arabic:

- Plain: مرحبا بالعالم (marḥaban bil-‘ālam – “hello, world”)
- With Latin and digits:
- UniWorld النسخة 2.0 من
- العربية + 1234 + English: مع النص المختلط UniWorld تجربة

Mixed BiDi paragraph:

- RTL: مرحبا بالعالم, then LTR: Hello, and back to RTL: שלום.

These lines should be rendered in **correct visual order** under UAX #9, with appropriate cursor mapping if UniWorld’s bidi and cursor logic is used.

---

## 5. South Asian Scripts – Indic and Beyond

### 5.1 Devanagari (Hindi)

- नमस्ते दुनिया (namaste duniya – “hello, world”)
- Conjuncts and matras: क् + ष → क्ष, श् + र → श्र
- Mixed word: प्रोग्रामिंग (programming)

### 5.2 Bengali

- হ্যালো বিশ্ব (hyalo biśśo – “hello, world”)
- Combined consonant clusters: ক্ষ, ভ্র, স্প্র

### 5.3 Gurmukhi (Punjabi)

- ਸਤ ਸ੍ਰੀ ਅਕਾਲ (sat sri akal – traditional greeting)

### 5.4 Tamil

- வணக்கம் உலகம் (vaṇakkam ulagam – “hello, world”)

## 5.5 Sinhala

- ඔහු ලෝ වර්ලඩ් (hello world, approximated)

These samples exercise: - **Grapheme cluster boundaries** for Indic consonant + virama + consonant sequences. - **Line breaking** around scripts with complex clusters.

---

## 6. Southeast Asian Scripts – Dictionary-based Line Breaking

These scripts traditionally **do not use spaces between words**, so line breaking depends on dictionary-based segmentation.

### 6.1 Thai

- Sentence (no spaces between words):  
สวัสดีชาวโลกนี่คือการทดสอบการตัดคำด้วย UniWorld

### 6.2 Lao

- ສະບາຍດີໂລກນີ້ແມ່ນການທົດສອບການຕັດຄໍາ

### 6.3 Khmer

- សួស្តីពិភពលោកនេះគឺជាការធ្វើតេស្តការកាត់ពាក្យ

### 6.4 Myanmar

- မင်္ဂလာပါကမ္ဘာလောကကြိုသည်မှာUniWorld၏ဝမ်းသာပွင့်ခြင်းဖြစ်သည်

When viewed in a UniWorld-aware editor or test harness:

- Line breaks should respect **dictionary segmentation** (Thai/Lao/Khmer/Myanmar).
  - Grapheme boundaries should never split inside stacked consonants or complex clusters.
- 

## 7. CJK – Chinese, Japanese, Korean

## 7.1 Chinese

- Simplified: 你好, 世界
- Traditional: 你好, 世界 (same orthography here, different fonts/locale may differ)

## 7.2 Japanese

- Kanji + Hiragana + Katakana + Latin:  
こんにちは、UniWorld ライブラリへようこそ - Unicode テキスト処理のテストです。

## 7.3 Korean

- Hangul: 안녕하세요, 유니월드 라이브러리입니다.

Display width checks (should be full-width = 2 columns per CJK ideograph/Hangul syllable):

- Monospace grid (conceptual):

text ASCII: [H][e][l][l][o] -> width 5 CJK: [你][好][世][界] -> width 8  
(4 × 2) Mixed: [H][i][!][你][好] -> width 2 + 1 + 4 = 7

These samples exercise: - **East Asian Width** handling in `display_width`. - **Grapheme clustering** for Hangul and kana with diacritics.

---

## 8. African and Other Scripts

### 8.1 Ethiopic (Amharic)

- ሰላም ዓለም (selam ālem – “hello, world”)

### 8.2 Tifinagh (Amazigh/Berber)

- ⵝⵉⵙⵉⵏ ⵏ ⵉⵎⵎⵉ (zisin n Immi – sample text)

### 8.3 Cherokee

- ፊጅጅ TEፊ (osiyo igvyi – “hello there, friend” approximation)

## 8.4 Canadian Aboriginal Syllabics (Cree, Inuktitut, Ojibwe)

Unified Canadian Aboriginal Syllabics (UCAS), used for Cree, Inuktitut, Ojibwe, and other Indigenous languages:

- Cree (Plains Cree syllabics): ᐃᓄᐅᑦ ᐃᓄᐅᑦᐱᑦ (sample greeting)
- Inuktitut: ᐃᓄᐅᑦᐱᑦ (Inuktitut – the language name)
- Word with hyphen: ᐃᓄᐅᑦᐱᑦ-ᐃᓄᐅᑦᐱᑦ (Canadian syllabics hyphen U+1400)

These test: - **Line breaking**: UCAS block (U+1400–U+167F) is AL (alphabetic) in UAX #14: hyphen U+1400 allows break after. - **Grapheme clustering**: one cluster per syllabic; no special GCB rules beyond defaults. - LTR; no dictionary-based segmentation (word boundaries follow spaces).

Section 8 (African and Other) tests: - Unicode range coverage. - Grapheme clustering in less-common scripts.

## 9. Emoji, ZWJ Sequences, and Flags

## 9.1 Basic Emoji

- Faces: 😊 😏 😄 😬 😟 😴 😎 😇 🤖 🦋
- Animals: 🐶 🐱 🐻 🐼 🐸 🐙 🐧 🦄 🦊
- Food: 🍏 🍕 🍟 🍜 🍪 🍩 🍵

Each should be a **single grapheme cluster** when treated by UniWorld.

## 9.2 Skin Tone Modifiers

- 
- The image displays two rows of emojis. The first row contains seven 'thumbs up' emojis in different skin tones: yellow, light pink, light brown, medium brown, dark brown, and black. The second row contains seven 'clapping hands' emojis in the same sequence of skin tones. Each emoji is accompanied by a small black dot to its left, likely representing a bullet point or a list item.

Each base + modifier pair must be a single grapheme cluster (emoji base + Extend).

### 9.3 ZWJ Sequences (Emoji ZWJ)

ZWJ (Zero Width Joiner, U+200D) sequences combine multiple emoji into a single rendered glyph. UniWorld must treat each full ZWJ sequence as **one grapheme cluster**.

Examples (codepoint sequences):

- Family: U+1F468 ZWJ U+1F469 ZWJ U+1F467 ZWJ U+1F466 (man + woman + girl + boy)
- Profession: U+1F469 ZWJ U+1F4BB (woman technologist)
- Relationship: U+1F469 ZWJ U+2764 U+FE0F ZWJ U+1F468 (couple with heart)

These are the most complex grapheme clusters in Unicode: multiple ExtPict code points joined by ZWJ into a single user-perceived character. UniWorld handles them correctly; the rendered ZWJ emoji are omitted in this PDF only because headless browser print-to-PDF has layout issues with such sequences, not because of any Unicode limitation in the library.

## 9.4 Regional Indicator Flags

Regional Indicator pairs (U+1F1E6..U+1F1FF) form flag emoji:

- UN (UN)
- US (US)
- CA (Canada)
- BR (Brazil)
- JP (Japan)

Regional indicators must pair: RI + RI = one cluster. UniWorld's grapheme segmentation must not split inside a flag pair.

---

## 10. Combining Marks and Edge Cases

Layered combining marks:

- à (a + grave + grave)
- ô (o + tilde + circumflex)
- é (e with acute and macron when normalized)

Visually similar but canonically distinct sequences:

- Å vs Å (precomposed vs decomposed).

These cases stress: - **Canonical equivalence** under NFC/NFD. - **Cluster boundaries** in the presence of multiple combining marks.

---

## 11. Simple Pictorials and Box Drawing

Unicode box-drawing table (small preview):

UniWorld Box
[x] Graphemes
[x] Width
[x] Bidi
[x] Breaks

Monospace art with mixed characters:

Column ruler:

12345678901234567890

ASCII: Hello, world!

CJK: 你好世界

Mixed: Hi你好

These test: - **Display width** alignment in monospace contexts. - Handling of box drawing characters.

---

## 12. Mixed-Script Paragraph (Stress Test)

Hello, שלום, مرحبا, नमस्ते, สวัสดี, 你好, 안녕하세요 -- this single paragraph mixes Latin, Arabic, Hebrew, Devanagari, Thai, Han, and Hangul. 12345. UniWorld's job is to ensure that segmentation (grapheme/word/sentence), bidi reordering, and line breaking all behave as the Unicode Standard intends, regardless of which binding (Rust, Python, JS/WASM, C, Go) you use.



You can use this paragraph to: - Test cursor movement (logical and visual) across mixed scripts. - Check line wrapping with different terminal widths. - Verify case folding (HELLO vs hello vs Straße vs STRASSE).

---

## 13. How to Use This File with UniWorld

Some suggested experiments:

- **Grapheme boundaries:**
- Run `uniworld.segment.grapheme_boundaries()` (Python) or `grapheme_boundaries()` (Rust/JS) on each section.
- Verify that:
  - Emoji ZWJ sequences and flags are single clusters.
  - Indic conjuncts with virama/nukta are not split.
  - Combining marks stay attached to their base.
- **Word boundaries:**
- Apply `word_boundaries()` and inspect how mixed scripts and punctuation are grouped.
- Confirm that dictionary-based segmentation improves Thai/Lao/Khmer/Myanmar breaks.
- **Normalization:**
- Compare NFC vs NFD on accented Latin samples and compatibility forms (ligatures, fractions).
- Ensure canonical-equivalent strings compare equal after normalization.
- **Line breaking:**
- Use `line_break_opportunities_with_dictionary()` on the SE Asian sections.
- Render with and without dictionary support to see the difference.

- **Display width and truncation:**
  - Use `display_width()` and `truncate_display_width()` on CJK/emoji sections at various widths to confirm visual clamping.
- 

## 14. Invitation

UniWorld exists so that **every script** gets **first-class treatment** in modern software:

- Editors and terminals that respect the people using them.
- Backends and services that handle global text correctly, not just ASCII.
- Applications that can be confidently localized into **any writing system** supported by Unicode.

If this document renders cleanly in your environment – with legible, correctly ordered text across scripts – then UniWorld is doing its job underneath. If you find edge cases, mis-renderings, or missing coverage, they become **actionable test cases** we can add to the suite.

*This file is a test output, but it is also an invitation: build a world where every script works correctly by default.*