

# Exploring Emoji Directionality

**Author:** Sean Stewart and Jennifer Daniel on behalf of the Emoji Subcommittee (ESC)

**Date:** Oct 19, 2022

## Introduction

As the ESC reviews requests from the public for new emoji, we encounter increasing evidence that the existing corpus of emoji is hitting or is rapidly about to hit thresholds of saturation. In order for emoji to operate at the speed of language online, the ESC's time and energy has turned towards looking at the emoji experience more holistically—that is, to see which improvements can be made to the existing repertoire of emoji—to address fundamental needs.

While there are no plans to stop adding new emoji characters, an incredibly powerful aspect of written language is that it consists of a finite number of characters that can "do it all". In this spirit, we are here to challenge a presumption that the primary way for emoji to move forward is to add more of them to the Unicode Standard. As the ESC actively explores viable paths forward, one track has demonstrated great potential and is the focus of this document: Directionality.

## Background

Some of the world's languages—notably those belonging to or related to the Semitic language family—write their sentences from right-to-left, whereas other languages write their languages either left-to-right (e.g. English, French, Tamil, etc.) or sometimes top-to-bottom (e.g. Chinese or Japanese). Focusing on the left-to-right (LTR) and right-to-left (RTL) directionality of the world's languages, digital user interfaces often need to adapt to the writing directionality of the language in which they are localized.

For example, major platforms horizontally mirror much of the user interface when a user runs in an Arabic or Hebrew localization. The reasoning behind the choice to mirror the interface lies within how users of LTR and RTL languages associate the concepts of "begin" and "end" with how the user would write a sentence. **LTR speakers associate "begin" with left and RTL users associate "begin" with right.**

This tightly coupled association of "begin" and "end" with the writing directionality of one's native language also affects how user's parse sequences of things (e.g. characters that compose words, words that compose sentences, etc.).

Unicode and many higher-level technologies have affordances for these differences of language directionality in order to support the varying conceptualizations of “begin” and “end”. See [UAX#9 \(Unicode Bidirectional Algorithm\)](#) for more information.

Specifically concerning to the Emoji Subcommittee is how sequences of emoji can unintentionally communicate different meanings when the sequence is presented to the user in reverse order when running in an RTL language environment.

The semantics of sequences of emoji can differ when writing directionality changes because emoji themselves are full-color, icon-like pictures that communicate complex semantics unlike individual characters that compose words. Because some emoji characters have semantics that encode implicit directionality (e.g. 🏃 🚗), when the string is mirrored, some meaning may be unintentionally lost or changed.

The ESC continues to investigate if this directionality challenge also affects languages with differing syntactic structures in addition to how a language is written.

This document discusses several options to help prevent the loss or change of meaning when emoji are used in right-to-left environments.

## The Challenge

As discussed above, user interface designers and implementers recognize this association between language writing direction and a user’s expectation where things begin and end. For example, when running in an RTL language, many vendor operating systems will generally mirror its UI “chrome” such that sequences of buttons and controls start being drawn on screen from the right hand side of the screen).

Given that emoji characters depict concepts using full-color, often skeuomorphic iconography (as opposed to using words or text to convey these concepts), some of the design-related challenges associated with language directionality also affect emoji characters. For example, the 🚀 ROCKET SHIP emoji may look unnatural to a native RTL language speaker, because the rocket ship encodes movement in space and time left-to-right. If this rocket ship is strung together in a sequence of other emoji, when the sequence of emoji is reversed when, for example, when surrounded by Arabic text (strong directionality characters), the awkwardness becomes even more evident.

## Example of Semantic Changes

Directionality	Sequence	Meaning
LTR		<i>Quickly running away from a line of cars</i>
RTL		<i>Warning to not run behind car fumes</i>

This contrived yet nevertheless representative example above shows that sequences of emoji—especially sequences of emoji that have inherent “strong” directionalities like the cars or the runner or even the puff of smoke—may be interpreted differently when input left-to-right but read right-to-left (i.e. typed by an English speaker and read by an Arabic speaker).

## Identification Process

All 1874 base emoji characters/sequences present within Unicode 15.0 ([Full Emoji List, v15.0](#)) continue to be analyzed by the ESC. Each of these entries will be added to one of three buckets:

1. **Strong Directionality** (i.e. encodes semantic movement)
2. **Weak Directionality** (i.e. encodes semantics involving transitivity)
3. **Neutral Directionality** (i.e. directionality has no effect on meaning)

These buckets share common terminology (which may change in the future) w/ [UAX#9](#). An emoji categorized as having **strong** directionality encodes meaning that specifically involves movement (e.g. transporting, walking, rolling, walking a dog, hitting a golf ball, throwing something, etc.). An emoji categorized as having **weak** directionality encodes meaning involving a transitive action whose meaning changes depending on how the emoji is oriented in a sequence (e.g. blowing on something, punching something, video recording something, using chopsticks to eat something, etc.)

Today, the ESC has identified 241 emoji that fall under either the **strong** or **weak** directionality buckets. All other emoji fall under the **neutral** bucket.

## Example Bucket Membership

Bucket	Member Emoji
Strong	
Weak	
Neutral	

## Potential Solutions

### 1. Universally Mirror All Emoji Characters

**Overview:** When an emoji or any sequence of emoji characters are present within an environment consisting of strong right-to-left directional characters (or even directionality override characters), all emoji become mirrored horizontally when displayed to the user.

**Benefits:**

1. Most simple solution
2. Least disruptive and most easily implemented
3. Leverages existing Unicode technologies (see [UAX#9](#))

**Challenges:**

1. Not all emoji characters technically require mirroring, so this approach can be considered computationally wasteful
2. All-or-nothing approach, leaving vendors with no guidance to allow the user to override the mirroring behavior

### 2. Mirror Emoji with Inherent Semantic Directionality

**Overview:** As a result of manual review, only some emoji appear to possess inherent semantic directionality, meaning only some emoji encode either movement or whose designs implement a specific direction / perspective. If the ESC were to create, publish, and maintain a wholly bespoke, new data file (or add attributes elsewhere) that identifies which emoji require special treatment in RTL environments, vendors can adopt this data in order to instruct their text rendering systems to “do the right thing” in RTL environments.

**Benefits:**

1. Most explicitly accurate solution that identifies exactly which emoji possess inherent semantic directionality
2. Affords specific control over emoji mirroring behavior by means of modifying the data file, should co-ordinated vendor design changes occur in the future

**Challenges:**

1. Disruptive in that new data files need to be created, published, maintained, **and** consumed by a client
2. Requires new means of accessing this data (ICU API or vendor-specific mechanisms)

### 3. Implement Support for UTS#51 Recommendations on Directionality

**Overview:** [UTS#51](#) has already been published with a potential solution to glyph directionality involving an open-ended arrow-character-and-ZWJ-based grammar that enables a user to explicitly specify which direction an emoji faces.

**Benefits:**

1. Most open-ended, requiring no additional data files
2. Moderately straight-forward to implement in text rendering stacks

**Challenges:**

1. Requires new input method-based UI to explicitly flip an emoji
2. No recommendations on if this flipping should occur automatically for RTL users
3. Requires RTL users to explicitly flip their emoji
4. Offers a strange experience for LTR language speakers if they encounter a sequence of emoji input by an RTL-aware input method

---

Special Thanks to Trevor Cortez and for his feedback on this document