Technical Reports

L2/26-200

Proposed Update Unicode® Technical Standard #51

UNICODE EMOJI

Version	18.0 (draft 2)
Editors	Mark Davis (Google LLC), Ned Holbrook (Apple Inc.)
Date	2026-07-23
This Version	https://www.unicode.org/reports/tr51/tr51-30.html
Previous Version	https://www.unicode.org/reports/tr51/tr51-29.html
Latest Version	https://www.unicode.org/reports/tr51/
Latest Proposed Update	https://www.unicode.org/reports/tr51/proposed.html
Revision	30

Summary

This document defines the structure of Unicode emoji characters and sequences, and provides data to support that structure, such as which characters are considered to be emoji, which emoji should be displayed by default with a text style versus an emoji style, and which can be displayed with a variety of skin tones. It also provides design guidelines for improving the interoperability of emoji characters across platforms and implementations.

Starting with Version 11.0 of this specification, the repertoire of emoji characters is synchronized with the Unicode Standard, and has the same version numbering system. For details, see Section 1.5.2, Versioning.

Status

*This is a **draft** document which may be updated, replaced, or superseded by other documents at any time. Publication does not imply endorsement by the Unicode Consortium. This is not a stable document; it is inappropriate to cite this document as other than a work in progress.*

> ***A Unicode Technical Standard (UTS)** is an independent specification. Conformance to the Unicode Standard does not imply conformance to any UTS.*

Please submit corrigenda and other comments with the online reporting form [Feedback]. Related information that is useful in understanding this document is found in the References. For the latest version of the Unicode Standard, see [Unicode]. For a list of current Unicode Technical Reports, see [Reports]. For more information about versions of the Unicode Standard, see [Versions].

Contents

1 Introduction
- Table: Emoji Proposals
- Table: Major Sources
- 1.1 Emoticons and Emoji
- 1.2 Encoding Considerations

1.3 Goals
1.4 Definitions
1.4.1 Emoji Characters
1.4.2 Emoji Presentation
1.4.3 Emoji and Text Presentation Sequences
1.4.4 Emoji Modifiers
1.4.5 Emoji Sequences
1.4.6 Emoji Sets
1.4.7 Notation
1.4.8 Property Stability
1.4.9 EBNF and Regex
1.5 Conformance
Table: Emoji Capabilities
1.5.1 Collation Conformance
1.5.2 Versioning
Table: Emoji Versions
2 Design Guidelines
2.1 Names
2.2 Display
2.3 Gender
Table: Emoji With Explicit Gender Appearance
Table: Emoji Changed to Gender-Neutral in Emoji 13.0+
2.3.1 Gender-Neutral Emoji
2.3.2 Marking Gender in Emoji Input
2.4 Diversity
Table: Emoji Modifiers
2.4.1 Implementations
Table: Sample Emoji Modifier Bases
Table: Expected Emoji Modifiers Display
2.4.2 Emoji Modifiers in Text
Table: Minipalettes
2.5 Emoji ZWJ Sequences
Table: ZWJ Sequence Display
2.6 Multi-Person Groupings
Table: Multi-Person Groupings
2.6.1 Multi-Person Gender
Table: Non-Gendered Family Sequences
Table: Examples of Gender with Multi-Person Groupings
2.6.2 Multi-Person Skin Tones
Table: Examples of Skin Tones for Multi-Person Groupings Using RGI Sequences
Table: Examples of Skin Tones for Multi-Person Groupings Using Single Characters
2.7 Emoji Implementation Notes
2.7.1 Emoji and Text Presentation Selectors
2.7.2 Handling Tag Characters
2.8 Hair Components
2.9 Color
Table: Emoji Glyph Color Examples
2.10 Emoji Glyph Facing Direction
Table: Emoji Glyph Direction Examples
2.11 Order of Emoji ZWJ Sequences
3 Which Characters are Emoji
4 Presentation Style
Table: Emoji versus Text Display
4.1 Emoji and Text Presentation Selectors
4.2 Emoji Locale Extension
4.3 Emoji Script Codes
4.4 Other Approaches for Control of Emoji Presentation

5 Ordering and Grouping
6 Input
Table: Palette Input
7 Searching
8 Longer Term Solutions
Annex A: Emoji Properties and Data Files
Table: Emoji Character Properties
A.1 Data Files
Table: Data Files
Annex B: Valid Emoji Flag Sequences
B.1 Presentation
B.2 Ordering
Annex C: Valid Emoji Tag Sequences
C.1 Flag Emoji Tag Sequences
C.1.1 Sample Valid Emoji Tag Sequences
Table: Display of Valid Emoji Tag Sequences
C.1.2 Sample Invalid Emoji Tag Sequences
Table: Display of Invalid Emoji Tag Sequences
C.1.3 Sample Ill-formed Emoji Tag Sequences
Table: Display of Ill-formed Emoji Tag Sequences
Acknowledgments
Rights to Emoji Images
References
Modifications

---

1 Introduction

Emoji are pictographs (pictorial symbols) that are typically presented in a colorful cartoon form and used inline in text. They represent things such as faces, weather, vehicles and buildings, food and drink, animals and plants, or icons that represent emotions, feelings, or activities.

Emoji on smartphones and in chat and email applications have become extremely popular worldwide. As of March 2015, for example, Instagram reported that “nearly half of text [on Instagram] contained emoji.” Individual emoji also vary greatly in popularity (and even by country), as described in the SwiftKey Emoji Report. See emoji press page for details about these reports and others.

Emoji are most often used in quick, short social media messages, where they connect with the reader and add flavor, color, and emotion. Emoji do not have the grammar or vocabulary to substitute for written language. In social media, emoji make up for the lack of gestures, facial expressions, and intonation that are found in speech. They also add useful ambiguity to messages, allowing the writer to convey many different possible concepts at the same time. Many people are also attracted by the challenge of composing messages in emoji, and puzzling out emoji messages.

The word *emoji* comes from Japanese:

絵 (e ≅ picture) 文字 (moji ≅ written character).

Emoji may be represented internally as graphics or they may be represented by normal glyphs encoded in fonts like other characters. These latter are called *emoji characters* for clarity. Some Unicode characters are normally displayed as emoji; some are normally displayed as ordinary text, and some can be displayed both ways.

There’s been considerable media attention to emoji since they appeared in the Unicode Standard, with increased attention starting in late 2013. For example, there were some 6,000 articles on the emoji appearing in Unicode 7.0, according to Google News. See the emoji press page for many samples of such articles, and also the Keynote from the 38th Internationalization & Unicode Conference.

Emoji became available in 1999 on Japanese mobile phones. There was an early proposal in 2000 to encode DoCoMo emoji in the Unicode standard. At that time, it was unclear whether these characters would come into widespread use—and there was not support from the Japanese mobile phone carriers to add them to Unicode—so no action was taken.

The emoji turned out to be quite popular in Japan, but each mobile phone carrier developed different (but partially overlapping) sets, and each mobile phone vendor used their own text encoding extensions, which were incompatible with one another. The vendors developed cross-mapping tables to allow limited interchange of emoji characters with phones from other vendors, including email. Characters from other platforms that could not be displayed were represented with 〓 (U+3013 GETA MARK), but it was all too easy for the characters to get corrupted or dropped.

When non-Japanese email and mobile phone vendors started to support email exchange with the Japanese carriers, they ran into those problems. Moreover, there was no way to represent these characters in Unicode, which was the basis for text in all modern programs. In 2006, Google started work on converting Japanese emoji to Unicode private-use codes, leading to the development of internal mapping tables for supporting the carrier emoji via Unicode characters in 2007.

There are, however, many problems with a private-use approach, and thus a proposal was made to the Unicode Consortium to expand the scope of symbols to encompass emoji. This proposal was approved in May 2007, leading to the formation of a symbols subcommittee, and in August 2007 the technical committee agreed to support the encoding of emoji in Unicode based on a set of principles developed by the subcommittee. The following are a few of the documents tracking the progression of Unicode emoji characters.

Emoji Proposals

Date	Doc No.	Title	Authors
2000-04-26	L2/00-152	NTT DoCoMo Pictographs	Graham Asher (Symbian)
2006-11-01	L2/06-369	Symbols (scope extension)	Mark Davis (Google)
2007-08-03	L2/07-257	Working Draft Proposal for Encoding Emoji Symbols	Kat Momoi, Mark Davis, Markus Scherer (Google)
2007-08-09	L2/07-274R	Symbols draft resolution	Mark Davis (Google)
2007-09-18	L2/07-391	Japanese TV Symbols (ARIB)	Michel Suignard (Microsoft)
2009-01-30	L2/09-026	Emoji Symbols Proposed for New Encoding	Markus Scherer, Mark Davis, Kat Momoi, Darick Tong (Google); Yasuo Kida, Peter Edberg (Apple)
2009-03-05	L2/09-025R2	Proposal for Encoding Emoji Symbols	
2010-04-27	L2/10-132	Emoji Symbols: Background Data	
2011-02-15	L2/11-052R	Wingdings and Webdings Symbols	Michel Suignard

In 2009, the first Unicode characters explicitly intended as emoji were added to Unicode 5.2 for interoperability with the ARIB (Association of Radio Industries and Businesses) set. A set of 722 characters was defined as the union of emoji characters used by Japanese mobile phone carriers: 114 of these characters were already in Unicode 5.2. In 2010, the remaining 608 emoji characters were added to Unicode 6.0, along with some other emoji characters. In 2012, a few more emoji were added to Unicode 6.1, and in 2014 a larger number were added to Unicode 7.0. Additional characters have been added since then, based on the *Selection Factors* found in Guidelines for Submitting Unicode Emoji Proposals.

Here is a summary of when some of the major sources of pictographs used as emoji were encoded in Unicode. Each source may include other characters in addition to emoji, and Unicode characters can correspond to multiple sources. The L column contains single-letter abbreviations of the various

sources for use in charts [emoji-charts] and data files [emoji-data]. Characters that do not correspond to any of these sources can be marked with Other (x).

Major Sources

Source	Abbr	L	Dev. Starts	Released	Unicode Version	Sample Character			
						B&W	Color	Code	CLDR Short Name
Zapf Dingbats	ZDings	z	1989	1991-10	1.0	✏	✏️	U+270F	*pencil*
ARIB	ARIB	a	2007	2008-10-01	5.2	☂	☔	U+2614	*umbrella with rain drops*
Japanese carriers	JCarrier	j	2007	2010-10-11	6.0	😎	😎	U+1F60E	*smiling face with sunglasses*
Wingdings & Webdings	WDings	w	2010	2014-06-16	7.0	🌶	🌶️	U+1F336	*hot pepper*

For a detailed view of when various source sets of emoji were added to Unicode, see **Emoji Version Sources** [emoji-charts]. The data file [JSources] shows the correspondence to the original Japanese carrier symbols.

People often ask how many emoji are in the Unicode Standard. This question does not have a simple answer, because there is no clear line separating which pictographic characters should be displayed with a typical emoji style. For a complete picture, see *Which Characters are Emoji*.

The colored images used in this document and associated charts [emoji-charts] are for illustration only. They do not appear in the Unicode Standard, which has only black and white images. They are either made available by the respective vendors for use in this document, or are believed to be available for non-commercial reuse. Inquiries for permission to use vendor images should be directed to those vendors, not to the Unicode Consortium. For more information, see *Rights to Emoji Images*.

1.1 Emoticons and Emoji

The term *emoticon* refers to a series of text characters (typically punctuation or symbols) that is meant to represent a facial expression or gesture (sometimes when viewed sideways), such as the following.

;-)

Emoticons predate Unicode and emoji, but were later adapted to include Unicode characters. The following examples use not only ASCII characters, but also U+203F (‿), U+FE35 (︵), U+25C9 (◉), and U+0CA0 (ಠ).

^‿^

◉︵◉

ಠ_ಠ

Often implementations allow emoticons to be used to input emoji. For example, the emoticon ;-) can be mapped to 😉 in a chat window. The term *emoticon* is sometimes used in a broader sense, to also include the emoji for facial expressions and gestures. That broad sense is used in the Unicode block name *Emoticons*, covering the code points from U+1F600 to U+1F64F.

1.2 Encoding Considerations

Unicode is the foundation for text in all modern software: it's how all mobile phones, desktops, and other computers represent the text of every language. People are using Unicode every time they type a key on their phone or desktop computer, and every time they look at a web page or text in an application. It is very important that the standard be stable, and that every character that goes into it be scrutinized carefully. This requires a formal process with a long development cycle. For example, the 🕶 *dark sunglasses* character was first proposed years before it was released in Unicode 7.0.

Characters considered for encoding must normally be in widespread use as elements of text. The emoji and various symbols were added to Unicode because of their use as characters for text-messaging in a number of Japanese manufacturers' corporate standards, and other places, or in long-standing use in widely distributed fonts such as Wingdings and Webdings. In many cases, the characters were added for complete round-tripping to and from a source set, *not* because they were inherently of more importance than other characters. For example, the 🕿 *clamshell phone* character was included because it was in Wingdings and Webdings, not because it is more important than, say, a "skunk" character.

In some cases, a character was added to complete a set: for example, a 🏉 *rugby football* character was added to Unicode 6.0 to complement the 🏈 *american football* character (the ⚽ *soccer ball* had been added back in Unicode 5.2). Similarly, a mechanism was added that could be used to represent all country flags (those corresponding to a two-letter unicode_region_subtag), such as the *flag for Canada*, even though the Japanese carrier set only had 10 country flags.

The data does not include non-pictographs, except for those in Unicode that are used to represent characters from emoji sources, for compatibility, such as:

🈹 or 🆔

Game pieces, such as the dominos (🁓 🁋 🁌 ... 🁢 🁣), are currently not included as emoji, with the exceptions of U+1F0CF (🃏) PLAYING CARD BLACK JOKER and U+1F004 (🀄) MAHJONG TILE RED DRAGON. These are included because they correspond each to an emoji character from one of the carrier sets.

The selection factors used to weigh the encoding of prospective candidates are found in *Selection Factors* in Guidelines for Submitting Unicode Emoji Proposals. That document also provides instructions for submitting proposals for new emoji.

For a list of frequently asked questions on emoji, see the Unicode Emoji FAQ.

1.3 Goals

This document provides:

- design guidelines for improving interoperability across platforms and implementations
- background information about emoji characters, and long-term alternatives
- data indicating:
  - which characters normally can be considered to be emoji
  - which emoji characters should be displayed by default in text style versus emoji style
  - which emoji characters may be displayed using a variety of skin tones, with implementation details
- pointers to [CLDR] data for
  - sorting emoji characters more naturally
  - annotations for searching and grouping emoji characters

It also provides background information about emoji, and discusses longer-term approaches to emoji.

As new Unicode characters are added or the "common practice" for emoji usage changes, the data and recommendations supplied by this document may change in accordance. Thus the recommendations and data will change across versions of this document.

1.4 Definitions

The following provide more formal definitions of some of the terms used in this document. Readers who are more interested in other features of the document may choose to continue from *Section 2, Design Guidelines*.

ED-1. emoji — A colorful pictograph that can be used inline in text. Internally the representation is either (a) an image, (b) an encoded character, or (c) a sequence of encoded characters.

- For (a) the term *emoji image* is used in this document. The term *sticker* may also be used.
- For (b) the term *emoji character* is used where necessary for clarity.
- For (c) the term *emoji sequence* is used for clarity.

ED-2. emoticon — (1) A series of text characters (typically punctuation or symbols) that is meant to represent a facial expression or gesture such as ;-) and (2) in a broader sense, also includes emoji for facial expressions and gestures.

1.4.1 Emoji Characters

ED-3. emoji character — A character that has the **Emoji** property.

```
emoji_character := \p{Emoji}
```

- These characters are recommended for use as emoji.

ED-4. extended pictographic character — a character that has the **Extended_Pictographic** property.

- These characters are pictographic, or otherwise similar in kind to characters with the Emoji property.
- The **Extended_Pictographic** property is used to customize segmentation (as described in [UAX29] and [UAX14]) so that possible future emoji ZWJ sequences will not break grapheme clusters, words, or lines. Unassigned codepoints with Line_Break=ID in some blocks are also assigned the **Extended_Pictographic** property. Those blocks are intended for future allocation of emoji characters.

ED-5. emoji component — A character that has the **Emoji_Component** property.

- These characters are used in emoji sequences but normally do not appear on emoji keyboards as separate choices, such as keycap base characters or Regional_Indicator characters.
- Some **emoji components** are **emoji characters**, and others (such as tag characters and **ZWJ**) are not.

For more information, see *Section 3, Which Characters are Emoji*. For information on data files which define emoji properties, see Annex A: Emoji Properties and Data Files.

1.4.2 Emoji Presentation

ED-6. default emoji presentation character — A character that, by default, should appear with an emoji presentation in well-formed sequences, rather than a text presentation.

```
default_emoji_presentation_character := \p{Emoji_Presentation}
```

- These characters have the **Emoji_Presentation** property. See Annex A: Emoji Properties and Data Files.

ED-7. default text presentation character — A character that, by default, should appear with a text presentation, rather than an emoji presentation.

```
default_text_presentation_character := \P{Emoji_Presentation}
```

- These characters do not have the **Emoji_Presentation** property; that is, their **Emoji_Presentation** property value is **No**. See Annex A: Emoji Properties and Data Files.

For more details about emoji and text presentation, see *Section 2, Design Guidelines* and *Section 4, Presentation Style*.

1.4.3 Emoji and Text Presentation Sequences

ED-8. text presentation selector — The character U+FE0E VARIATION SELECTOR-15 (VS15), used to request a text presentation for an emoji character. (Also known as *text variation selector* in prior versions of this specification.)

```
text_presentation_selector := \x{FE0E}
```

ED-8a. text presentation sequence — A variation sequence consisting of an **emoji character** followed by a **text presentation selector**.

```
text_presentation_sequence := emoji_character text_presentation_selector
```

- The only valid **text presentation sequences** are those listed in the **emoji-variation-sequences.txt** file [emoji-data].

ED-9. emoji presentation selector — The character U+FE0F VARIATION SELECTOR-16 (VS16), used to request an emoji presentation for an emoji character. (Also known as *emoji variation selector* in prior versions of this specification.)

```
emoji_presentation_selector := \x{FE0F}
```

ED-9a. emoji presentation sequence — A variation sequence consisting of an **emoji character** followed by a **emoji presentation selector**.

```
emoji_presentation_sequence := emoji_character emoji_presentation_selector
```

- The only valid **emoji presentation sequences** are those listed in the **emoji-variation-sequences.txt** file [emoji-data].

ED-10. (This definition has been removed.)

1.4.4 Emoji Modifiers

ED-11. emoji modifier — A character that can be used to modify the appearance of a preceding emoji in an *emoji modifier sequence*.

```
emoji_modifier := \p{Emoji_Modifier}
```

- These characters have the **Emoji_Modifier** property. See Annex A: Emoji Properties and Data Files.

ED-12. emoji modifier base — A character whose appearance can be modified by a subsequent emoji modifier in an *emoji modifier sequence*.

```
emoji_modifier_base := \p{Emoji_Modifier_Base}
```

- These characters have the **Emoji_Modifier_Base** property. See Annex A: Emoji Properties and Data Files.
- They are also listed in Characters Subject to Emoji Modifiers.

ED-13. emoji modifier sequence — A sequence of the following form:

```
emoji_modifier_sequence :=
  emoji_modifier_base emoji_modifier
```

For more details about emoji modifiers, see *Section 2.4, Diversity.*

1.4.5 Emoji Sequences

ED-14. emoji flag sequence — A sequence of two Regional Indicator characters.

```
emoji_flag_sequence :=
  regional_indicator regional_indicator

regional_indicator := \p{Regional_Indicator}
```

- The only valid **emoji flag sequences** are those listed in the **emoji-sequences.txt** file [emoji-data]. See also Annex B: Valid Emoji Flag Sequences.
- A singleton Regional Indicator character is not a well-formed **emoji flag sequence**.

ED-14a. emoji tag sequence (ETS) — A sequence of the following form:

```
emoji_tag_sequence := tag_base tag_spec tag_end
tag_base           := emoji_character
                    | emoji_modifier_sequence
                    | emoji_presentation_sequence
tag_spec           := [\x{E0020}-\x{E007E}]+
tag_end            := \x{E007F}
```

- The `tag_spec` consists of all characters from U+E0020 TAG SPACE to U+E007E TAG TILDE. Each `tag_spec` defines a particular visual variant to be applied to the `tag_base` character(s). Though `tag_spec` includes the values U+E0041 TAG LATIN CAPITAL LETTER A .. U+E005A TAG LATIN CAPITAL LETTER Z, they are not used currently and are reserved for future extensions.
- The `tag_end` consists of the character U+E007F CANCEL TAG, and must be used to terminate the sequence.
- A sequence of tag characters that is not part of an `emoji_tag_sequence` is not a well-formed **emoji tag sequence**.

The meaning and validity criteria for an **emoji tag sequence** and expected visual variants for a `tag_spec` are determined by Annex C: Valid Emoji Tag Sequences.

ED-14b. (This definition has been removed.)

ED-14c. emoji keycap sequence — A sequence of the following form:

```
emoji_keycap_sequence := [0-9#*] \x{FE0F 20E3}
```

- These sequences are in the **emoji-sequences.txt** file [emoji-data] listed under the type_field **Emoji_Keycap_Sequence**

ED-15. emoji core sequence — A sequence of the following form:

```
emoji_core_sequence :=
  emoji_character
| emoji_presentation_sequence
| emoji_keycap_sequence
| emoji_modifier_sequence
| emoji_flag_sequence
```

ED-15a. emoji ZWJ element — An element that can be used in an emoji ZWJ sequence, as follows:

```
emoji_zwj_element :=
  emoji_core_sequence
| emoji_tag_sequence
```

ED-16. emoji ZWJ sequence — An emoji sequence with at least one joiner character.

```
emoji_zwj_sequence :=
  emoji_zwj_element ( ZWJ emoji_zwj_element )+

ZWJ := \x{200d}
```

ED-17. emoji sequence — A core sequence, tag sequence, or ZWJ sequence, as follows:

```
emoji_sequence :=
  emoji_core_sequence
| emoji_zwj_sequence
| emoji_tag_sequence
```

Note that all emoji sequences are single grapheme clusters: there is never a grapheme cluster boundary within an emoji sequence. This affects editing operations, such as cursor movement or deletion, as well as word break, line break, and so on. For more information, see [UAX29].

ED-17a. qualified emoji character — An emoji character in a string that (a) has default emoji presentation or (b) is the first character in an emoji modifier sequence or (c) is not a default emoji presentation character, but is the first character in an emoji presentation sequence.

ED-18. fully-qualified emoji — A qualified emoji character, or an emoji sequence in which each emoji character is qualified.

ED-18a. minimally-qualified emoji — An emoji sequence in which the first character is qualified but the sequence is not fully qualified.

ED-19. unqualified emoji — An emoji that is neither fully-qualified nor minimally qualified.

For recommendations on the use of variation selectors in emoji sequences, see *Section 2.7, Emoji Implementation Notes*.

1.4.6 Emoji Sets

The following sets are defined based on the data files and properties described in Annex A: Emoji Properties and Data Files. The composition of these sets may change from one release to the next.

Each of these sets can be conceived of as a binary property; they are properties of strings. See *UTS #18: Unicode Regular Expressions* [UTS18] and *UTR #23: The Unicode Character Property Model* [UTR23] for more discussion.

ED-20. basic emoji set — The set of emoji characters and emoji presentation sequences listed in the **emoji-sequences.txt** file [emoji-data] under the type_field **Basic_Emoji**.

- This is the set of emoji intended for general-purpose input.
- This set excludes all those instances of an **emoji component** that are not intended for independent, direct input. Implementations should support independent display of **emoji components** in this set even if they are not made available for direct input.
  - Skin tone modifiers and hair components should be displayed even in isolation, but they should not (typically) be on the keyboard palette. These are included in Basic_Emoji.
  - Other components (U+20E3 COMBINING ENCLOSING KEYCAP, Regional Indicators, tag characters, ZWJ, and VS16) should never have an emoji presentation in isolation, but do occur as part of emoji sequences. These are not included in Basic_Emoji.
- This set otherwise includes all instances of an **emoji character** with the property value **Emoji_Presentation** = Yes and all instances of a valid **emoji presentation sequence** whose base character has the property value **Emoji_Presentation** = No.

ED-21. emoji keycap sequence set — The specific set of emoji sequences listed in the **emoji-sequences.txt** file [emoji-data] under the type_field **Emoji_Keycap_Sequence**.

- This is the set of all valid **emoji keycap sequences.**

Note: The following definitions use the acronym "RGI" to mean "recommended for general interchange", referring to that subset of some larger set that is intended to be widely supported across multiple platforms.

ED-22. RGI emoji modifier sequence set — The specific set of emoji sequences listed in the **emoji-sequences.txt** file [emoji-data] under the type_field **RGI_Emoji_Modifier_Sequence**.

- This is the *subset* of all valid **emoji modifier sequences** recommended for general interchange.

ED-23. RGI emoji flag sequence set — The specific set of emoji sequences listed in the **emoji-sequences.txt** file [emoji-data] under the type_field **RGI_Emoji_Flag_Sequence**.

- This is the *subset* of all valid **emoji flag sequences** recommended for general interchange. See Annex B: Valid Emoji Flag Sequences

ED-24. RGI emoji tag sequence set — The specific set of emoji sequences listed in the **emoji-sequences.txt** file [emoji-data] under the type_field **RGI_Emoji_Tag_Sequence**.

- This is the *subset* of all valid **emoji tag sequences** recommended for general interchange. See Annex C: Valid Emoji Tag Sequences.

ED-25. RGI emoji ZWJ sequence set — The specific set of emoji sequences listed in the **emoji-zwj-sequences.txt** file [emoji-data] under the type_field **RGI_Emoji_ZWJ_Sequence**.

- This is the *subset* of all valid **emoji ZWJ sequences** recommended for general interchange.

ED-26. (This definition has been removed.)

ED-27. RGI emoji set — The set of all emoji (characters and sequences) covered by ***ED-20***, ***ED-21***, ***ED-22***, ***ED-23***, ***ED-24***, and ***ED-25***.

- This is the subset of all valid emoji (characters and sequences) recommended for general interchange.
- This corresponds to the **RGI_Emoji** property. This property is not subject to any stability policies at this time.

ED-28. RGI_Emoji_Qualification — the status of emoji sequences
This is an enumerated property of strings, defined by the **emoji-test.txt** file [emoji-data]. It assigns one of four values to each emoji in ED-27 RGI emoji set and related sequences with missing variation selectors. The property value names and short aliases are:

Value Name	Alias	Description
Standalone_Component	component	an **emoji component**
Fully_Qualified	FQE	ED-18, excluding **emoji components**
Minimally_Qualified	MQE	ED-18a
Unqualified	UQE	ED-19

The property values are defined by the corresponding status values in [emoji-data]: component, fully-qualified, minimally-qualified, and unqualified.

1.4.7 Notation

Character names in all capitals are the formal Unicode Name property values, such as U+1F473 MAN WITH TURBAN. The formal names are immutable internal identifiers, but often do not reflect the current practice for interpretation of the character.

Lowercase character names for existing characters or sequences are CLDR short names, such as U+1F473 *person wearing turban*.

1.4.8 Property Stability

The emoji properties are stable for each version of the data—they will not change for that version. They may, however, change between that version and a subsequent version. For example, isEmoji(🕴)=false for Emoji Version 5.0, but true for Version 11.0.

Some emoji properties are not closed over certain string operations. For example:

isEmoji(toLowercase(X)) ≠ isEmoji(X) for the case of X= Ⓜ️ , because:

isEmoji(Ⓜ️) = true
toLowercase(Ⓜ️) = ⓜ
isEmoji(ⓜ) = false

Casing operations may produce invalid variation sequences. While the following strings form a case pair, the ***emoji presentation selector*** is not defined for ⓜ, and thus has no effect on its rendering:

Ⓜ️ = <U+24C2 CIRCLED LATIN CAPITAL LETTER M, U+FE0F VS16>	valid variation sequence
ⓜ = <U+24DC CIRCLED LATIN SMALL LETTER M, U+FE0F VS16>	invalid variation sequence

1.4.9 EBNF and Regex

The following EBNF can be used to quickly scan for *possible emoji*. Those possible emoji can then be verified where necessary by performing validity tests according to the definitions, or checking against the ***RGI emoji set***. It is much simpler than the expressions currently in the definitions. It includes a superset of emoji as a by-product of that simplicity, but the extras can be weeded out by validity tests.

EBNF	Notes
`possible_emoji := zwj_element (\x{200D} zwj_element)*`	\x{200D} = zero-width joiner
`flag_sequence := \p{RI} \p{RI}`	\p{RI} = Regional_Indicator
`zwj_element := \p{Emoji} emoji_modification? | flag_sequence`	
`emoji_modification := \p{EMod} | \x{FE0F} \x{20E3}? | tag_modifier`	\p{EMod} = Emoji_Modifier \x{FE0F} = emoji VS \x{20E3} = enclosing keycap
`tag_modifier := [\x{E0020}-\x{E007E}]+ \x{E007F}`	\x{E00xx} are tags \x{E007F} = TERM tag

From these EBNF rules a regex can be generated, as below. While this regex may seem complex, it is far simpler than what would result from the definitions. Direct use of the definitions would result in regex expressions which are many times more complicated, and yet still require verification with validity tests.

Regex

```
\p{RI} \p{RI}
| \p{Emoji}
  ( \p{EMod}
  | \x{FE0F} \x{20E3}?
  | [\x{E0020}-\x{E007E}]+ \x{E007F}
  )?
  (\x{200D}
    ( \p{RI} \p{RI}
    | \p{Emoji}
      ( \p{EMod}
      | \x{FE0F} \x{20E3}?
      | [\x{E0020}-\x{E007E}]+ \x{E007F}
      )?
    )
  )*
```

1.5 Conformance

Conformance to this specification is specified by the following clauses.

C1. An implementation claiming conformance to this specification shall identify the version of this specification to which conformance is claimed.

- Each version of this specification has a minimum version of the Unicode Standard, which contains all the characters with **Emoji=Yes**. For example, an implementation that claims conformance to Emoji 5.0 must also have support for the Unicode 9.0 repertoire.

C2. An implementation claiming conformance to this specification shall identify which of the capabilities specified below are supported for which emoji sets **ED-20** through **ED-25**. This must include at least the **C2a display** capability for set **ED-20 basic emoji set**. For example, an implementation can declare that it supports the **display**, **editing** and **input** capabilities for the **basic emoji set**, and the **display** and **editing** capabilities for the **emoji modifier sequence set**, and may make no claim of capabilities for any other sets.

Emoji Capabilities

C2a display	The implementation is capable of displaying each of the characters and sequences in the specified set as a single glyph with emoji presentation.
C2b editing	The implementation treats each of the characters and sequences in the specified set as an indivisible unit for editing purposes (cursor movement, deletion, line breaking, and so on).
C2c input	The implementation provides a mechanism for inputting each of the characters and sequences in the specified set as a single glyph with emoji presentation.

An implementation may claim *partial conformance* to C2, specifying the set of characters that it does not support. For example, an implementation could claim conformance to C2 for all emoji sets and capabilities except for the set [⏏ {UN}], that is:

- U+23CF *eject button*
- U+1F1FA U+1F1F3 *United Nations*

C3. An implementation claiming conformance to this specification must not support an invalid emoji_flag_sequence or invalid or ill-formed emoji_tag_sequence for **display** or **input**, except for a fallback **display** depiction indicating the presence of an invalid sequence, such as 🏴‍☠️.

- A singleton emoji Regional Indicator may be displayed as a capital A..Z character with a special display

An implementation *may* support any of the following for **display**, **editing**, or **input**:

- a single code point outside of the **basic emoji set**
- an emoji sequence that would be in one of the emoji sets **ED-20** through **ED-25** except that it is missing one or more **emoji presentation selectors**
- an emoji ZWJ sequence that is not in **ED-25**

1.5.1 Collation Conformance

Implementations can claim conformance for emoji collation or short names by conforming to a particular version of CLDR.

1.5.2 Versioning

Starting with Version 11.0 of this specification, the repertoire of emoji characters is synchronized with the Unicode Standard, and has the same version numbering system.

As of version 13.0, data file comments use the labeling convention “Ex.x”. This label corresponds to the Emoji version when the emoji character or emoji sequence was first defined in associated data files. For example, the label “E5.0” is associated with Unicode Emoji, Version 5.0. There are three special values used primarily for emoji characters before the official release of Emoji 1.0 in 2015:

Label	Intended Coverage
E0.0	This label is used for special characters, including: • Most emoji component characters, regardless of when they were first encoded. • Other non-emoji characters in the data files.
E0.6	Emoji characters added to Unicode 6.0. This includes the emoji characters deriving from Japanese carrier sets, as well as some characters from the ARIB Japanese television standard.
E0.7	Emoji characters added to Unicode 7.0. This consists largely of emoji deriving from the Windows Wingding and Webdings sets, but also includes more characters from the ARIB Japanese television standard.

The following table shows the corresponding Emoji version and Unicode Standard version, up through Version 18.0, including the labels used in data file comments.

Emoji Versions

Emoji Version	Date	Unicode Version	Data File Comment
N/A	various	various	E0.0
N/A	2010-10-11	Unicode 6.0	E0.6
N/A	2014-06-16	Unicode 7.0	E0.7
Emoji 1.0	2015-06-09	Unicode 8.0	E1.0
Emoji 2.0	2015-11-12	Unicode 8.0	E2.0
Emoji 3.0	2016-06-03	Unicode 9.0	E3.0
Emoji 4.0	2016-11-22	Unicode 9.0	E4.0
Emoji 5.0	2017-06-20	Unicode 10.0	E5.0
Emoji 11.0	2018-05-21	Unicode 11.0	E11.0
Emoji 12.0	2019-03-05	Unicode 12.0	E12.0
Emoji 12.1	2019-10-21	Unicode 12.1	E12.1
Emoji 13.0	2020-03-10	Unicode 13.0	E13.0

Emoji 13.1	2020-09-15	Unicode 13.0	E13.1
Emoji 14.0	2021-09-14	Unicode 14.0	E14.0
Emoji 15.0	2022-09-13	Unicode 15.0	E15.0
Emoji 15.1	2023-09-12	Unicode 15.1	E15.1
Emoji 16.0	2024-09-10	Unicode 16.0	E16.0
Emoji 17.0	2025-09-09	Unicode 17.0	E17.0
Emoji 18.0	2026-09-15	Unicode 18.0	E18.0

2 Design Guidelines

Unicode characters can have many different presentations as text. An “a” for example, can look quite different depending on the font. Emoji characters can have two main kinds of presentation:

- the primary *emoji presentation,* with colorful and perhaps whimsical shapes, even animated
- a *text presentation,* such as black & white

More precisely, a text presentation is a simple foreground shape whose color is determined by other information, such as setting a color on the text, while an emoji presentation determines the color(s) of the character, and is typically multicolored. In other words, when someone changes the text color in a word processor, a character with an emoji presentation will not change color.

Any Unicode character can be presented with a text presentation, as in the Unicode charts. For the emoji presentation, both the name and the representative glyph in the Unicode chart should be taken into account when designing the appearance of the emoji, along with the images used by other vendors. The shape of the character can vary significantly. For example, here are just a few of the possible images for U+1F36D LOLLIPOP, U+1F36E CUSTARD, U+1F36F HONEY POT, and U+1F370 SHORTCAKE:

While the shape of the character can vary significantly, designers should maintain the same “core” shape, based on the shapes used mostly commonly in industry practice. For example, a U+1F36F HONEY POT encodes for a pictorial representation of a pot of honey, not for some semantic like “sweet”. It would be unexpected to represent U+1F36F HONEY POT as a sugar cube, for example. Deviating too far from that core shape can cause interoperability problems: see accidentally-sending-friends-a-hairy-heart-emoji. Direction (whether a person or object faces to the right or left, up or down) should also be maintained where possible, because a change in direction can change the meaning: when sending 🏊‍♂️🐊 “person escaping from crocodile”, people expect any recipient to see the person swimming in the same direction as when they composed it. See *Section 2.10, Emoji Glyph Facing Direction*.

Emoji should have a generic appearance. For certain emoji, specific modifiers for skin tone and gender can be applied. See Section 2.4 Diversity.

Flag emoji characters are discussed in Annex B: Valid Emoji Flag Sequences.

2.1 Names

Every emoji has a CLDR short name, which may change over time. Every emoji character also has a formal Unicode name, like every other Unicode character; this is a permanent identifier which cannot be changed.

The formal Unicode name of a Unicode character does not determine its appearance. Formal names of symbols such as BLACK MEDIUM SQUARE or WHITE MEDIUM SQUARE are not meant to indicate that the corresponding character must be presented in black or white, respectively; rather, the use of “black” and “white” in the names is generally just to contrast **filled** versus **outline** shapes, or a darker color fill versus a lighter color fill. Similarly, in other symbols such as the hands U+261A BLACK LEFT POINTING INDEX and U+261C WHITE LEFT POINTING INDEX, the words “white” and “black” also refer to outlined versus filled, and do not indicate skin color.

However, other color words in the name, such as YELLOW, typically provide a recommendation as to the emoji presentation, which should be followed to avoid interoperability problems.

In many cases the consensus for the best depiction has evolved in the time since the original formal name was standardized, and the preferred depiction is now better reflected by the CLDR short name. For example, U+1F483 DANCER should be designed in accordance with the CLDR short name *woman dancing* (an additional character was added for *man dancing*). In addition, only emoji characters have formal Unicode names; the emoji sequences just have CLDR short names.

The formal Unicode name of each character must be unique, and sometimes distinguishing words are included in the name to maintain that uniqueness when two contrasting characters are added, such as:

🐶 U+1F436 DOG FACE
🐕 U+1F415 DOG

🐮 U+1F42E COW FACE
🐄 U+1F404 COW

In cases such as these, the images must also contrast. However, in some cases additional terms like FACE were added to the name when they were not needed for uniqueness. There is no requirement that an image contrast be maintained where there are not contrasting emoji. Consider the following emoji:

🦌 U+1F98C DEER

🦓 U+1F993 ZEBRA FACE

Because there are no other contrasting DEER or ZEBRA emoji, each of these two could be depicted with a face only, face and shoulders, full body, or other choices.

2.2 Display

Emoji characters may not always be displayed on a white background. They are often best given a faint, narrow contrasting border to keep the character visually distinct from a similarly colored background. Thus a Japanese flag would have a border so that it would be visible on a white background, and a Swiss flag have a border so that it is visible on a red background.

Current practice is for emoji to have a square aspect ratio, deriving from their origin in Japanese. For interoperability, it is recommended that this practice be continued with current and future emoji. They will typically have about the same vertical placement and advance width as CJK ideographs. For example:

They should use transparency for proper display for selection and with colored backgrounds:

The set of supported emoji sequences may vary by platform. For example, take the following emoji ZWJ sequence:

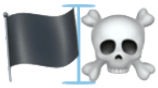
On a particular platform, it can be shown as a single image:

However, if that combination is not supported as a single unit, it may show up as a sequence like the following, and the user sees no indication that it was meant to be composed into a single image:

Implementations could provide an indication of the composed nature of an unsupported emoji sequence where possible. This gives users the additional information that that sequence was intended to have a composed form. It also explains why the sequence will not behave as separate elements: The arrow key will not move between the flag and the skull & crossbones, and line breaks will not occur between apparently separate emoji.

The following is an example of an approach that implementations can use. There are other approaches that could have a more intuitive appearance, but that could be difficult to implement with current text display mechanisms.

Display the ZWJ as a visible “glue” character, with zero or very narrow width.



2.3 Gender

The following human-form emoji are currently considered to have explicit gender appearance based on the name and/or practice. They intentionally contrast with other characters. This list may change in the future if new explicit-gender characters are added, or if some of these are changed to be gender-neutral. The names below are the CLDR short names, followed by the formal Unicode name in capital letters if it differs.

Emoji With Explicit Gender Appearance

Female		Male	
U+1F467	girl	U+1F466	boy
U+1F469	woman	U+1F468	man
U+1F475	old woman OLDER WOMAN	U+1F474	old man OLDER MAN
U+1F46D	women holding hands TWO WOMEN HOLDING HANDS	U+1F46C	men holding hands TWO MEN HOLDING HANDS

U+1F936	Mrs. Claus MOTHER CHRISTMAS	U+1F385	Santa Claus FATHER CHRISTMAS
U+1F478	princess	U+1F934	prince
U+1F483	woman dancing DANCER	U+1F57A	man dancing
U+1F930	pregnant woman	U+1FAC3	pregnant man
U+1F931	breast-feeding		
U+1F9D5	woman with headscarf PERSON WITH HEADSCARF		
Explicit Gender Combination			
U+1F46B	woman and man holding hands MAN AND WOMAN HOLDING HANDS		

The emoji in the table Emoji Changed to Gender-Neutral in Emoji 13.0+ below have been removed from the table Emoji With Explicit Gender Appearance, and the CLDR names for most have been changed to use *person* (along with some other changes). The *person with veil* and *person in tuxedo* emoji also have RGI man and woman gender variants. The others do not; for *person in suit levitating* and *person with skullcap*, the visual distinctions would be unclear at emoji sizes.

Emoji Changed to Gender-Neutral in Emoji 13.0+

Gender-Neutral		
E13.0	U+1F470	person with veil BRIDE WITH VEIL
	U+1F935	person in tuxedo MAN IN TUXEDO
	U+1F574	person in suit levitating MAN IN BUSINESS SUIT LEVITATING
	U+1F472	person with skullcap MAN WITH GUA PI MAO
E13.1	U+1F9D4	person: beard BEARDED PERSON

2.3.1 Gender-Neutral Emoji

It is often the case that gender is unknown or irrelevant, as in the usage “Is there a doctor on the plane?,” or a gendered appearance may not be desired. Such cases are known as “gender-neutral,” “gender-inclusive,” “unspecified-gender,” or many other terms. Except for the emoji shown in the table Emoji With Explicit Gender Appearance, human-form emoji should normally be depicted in a gender-neutral way unless gender appearance is explicitly specified using an emoji ZWJ sequence in one of the ways shown in the following table.

Gender Appearance Mechanisms

Type	Description	Examples
Sign Format	A human-form emoji can be given explicit gender using a ZWJ sequence. The sequence contains the base	man runner = RUNNER + ZWJ + MALE SIGN woman runner = RUNNER + ZWJ + FEMALE SIGN runner = RUNNER

	emoji followed by ZWJ and either FEMALE SIGN or MALE SIGN. The human-form emoji alone should be gender-neutral in form.	
Object Format	A profession or role emoji can be formed using a ZWJ sequence. The sequence starts with MAN or WOMAN followed by ZWJ and ending with an object. The ADULT character can be used for a gender-neutral version.	man astronaut = MAN + ZWJ + ROCKET SHIP woman astronaut = WOMAN + ZWJ + ROCKET SHIP astronaut = ADULT + ZWJ + ROCKET SHIP

Although the human-form emoji used in sign format type ZWJ sequences are supposed to have gender-neutral appearance by themselves (when not used in a sign format type ZWJ sequence), many vendors previously depicted these human-form emoji as a man or woman. As a result, they had the same appearance as one of the sign format type ZWJ sequences. For example, most vendors depicted *detective* as *man detective* and *person getting haircut* as *woman getting haircut*, but some vendors depicted *police officer* as *man police officer* while others depicted it as *woman police officer*.

Gender-neutral versions of the profession or role emoji using object format type ZWJ sequences are promulgated by adding them to the ***RGI emoji ZWJ sequence set***.

2.3.2 Marking Gender in Emoji Input

Emoji input systems such as keyboards or palettes typically provide for input of some emoji whose appearance is explicitly gendered—for example, emoji that appear specifically as a woman or man. When such emoji are not included in the table ***Emoji With Explicit Gender Appearance***, the input system should generate a sequence for them that explicitly indicates the gendered appearance, rather than relying on a particular system's default appearance. This principle is shown with the following example:

Assume on some system that the default appearance of *detective* is as *man detective*. On that system, when entering *man detective*, an input system should still use the explicit sequence

`U+1F575 U+FE0F U+200D U+2642 U+FE0F` (*man detective*)

rather than just

`U+1F575 U+FE0F` (*detective*)

2.4 Diversity

Five symbol modifier characters that provide for a range of skin tones for human emoji were released in Unicode Version 8.0. These characters are based on the six tones of the Fitzpatrick scale, a recognized standard for dermatology. The exact shades may vary between implementations.

Emoji Modifiers

Code	CLDR Short Name	Unicode Character Name	Samples	
U+1F3FB	*light skin tone*	EMOJI MODIFIER FITZPATRICK TYPE-1-2		
U+1F3FC	*medium-light skin tone*	EMOJI MODIFIER FITZPATRICK TYPE-3		
U+1F3FD	*medium skin tone*	EMOJI MODIFIER FITZPATRICK TYPE-4		
U+1F3FE	*medium-dark skin tone*	EMOJI MODIFIER FITZPATRICK TYPE-5		
U+1F3FF	*dark skin tone*	EMOJI MODIFIER FITZPATRICK TYPE-6		

These characters have been designed so that even where diverse color images for human emoji are not available, readers can see the intended meaning.

When used alone, the default representation of these modifier characters is a color swatch. Whenever one of these characters *immediately* follows certain characters (such as WOMAN), then a font should show the sequence as a single glyph corresponding to the image for the person(s) or body part with the specified skin tone, such as the following:

However, even if the font doesn't show the combined character, the user can still see that a skin tone was intended:

When a human emoji is not *immediately* followed by an emoji modifier character, it should use a generic, *non-realistic* skin tone, such as RGB #FFCC22 (one of the colors typically used for the smiley faces).

No particular hair color is required, however, dark hair is generally regarded as more neutral because black or dark brown hair is widespread among people of every skin tone. This does not apply to emoji that already have an explicit hair color such as PERSON WITH BLOND HAIR (originally added for compatibility with Japanese mobile phone emoji), which needs to have blond hair regardless of skin tone.

To have an effect on an emoji, an emoji modifier must immediately follow that base emoji character. Emoji presentation selectors are neither needed nor recommended for emoji characters when they are followed by emoji modifiers, and should not be used in newly generated emoji modifier sequences; the emoji modifier automatically implies the emoji presentation style. See *ED-13. emoji modifier sequence*. However, some older data may include *defective* emoji modifier sequences in which an emoji presentation selector does occur between the base emoji character and the emoji modifier; this is the only exception to the rule that an emoji modifier must immediately follow the character that it modifies. In this case the emoji presentation selector should be ignored. For handling text presentation selectors in sequences, see *Section 4, Presentation Style*.

<U+270C VICTORY HAND, FE0F, TYPE-3>

Any other intervening character causes the emoji modifier to appear as a free-standing character. Thus

2.4.1 Implementations

Implementations can present the emoji modifiers as separate characters in an input palette, or present the combined characters using mechanisms such as long press.

The emoji modifiers are not intended for combination with arbitrary emoji characters. Instead, they are restricted to the emoji modifier base characters: no other characters are to be combined with emoji modifiers. This set may change over time, with successive versions of this document. To find the exact list of emoji modifier bases for each version, use the Emoji_Modifier_Base character property, as described in Annex A: Emoji Properties and Data Files.

Sample Emoji Modifier Bases

The following chart shows the expected display with emoji modifiers, depending on the preceding character and the level of support for the emoji modifier. The “Unsupported” rows show how the character would typically appear on a system that does not have a font with that character in it: with a missing glyph indicator. In some circumstances, display of an emoji modifier following an Emoji_Modifier_Base character should be suppressed:

> If an emoji modifier base has no skin visible on a particular system, then any following emoji modifier should be suppressed.

In other circumstances, display of an emoji modifier following an Emoji_Modifier_Base character may be suppressed:

> If a particular emoji modifier base uses a non-realistic skin tone that differs from the default skin tone used for other Emoji_Modifier_Base characters, then any following emoji modifier may be suppressed. For example, suppose *genie* is shown with blue skin in a particular implementation while other Emoji_Modifier_Base characters are shown with neon yellow skin in the absence of emoji modifiers; any emoji modifier following *genie* may be suppressed.

Expected Emoji Modifiers Display

Support Level	Emoji Modifier Base	Sequence	Display
Fully supported	Yes		
	Yes		
	Yes, but no skin visible		
	Yes, but unusual default skin tone		
	No		
Fallback	Yes		
	No		
Unsupported	Yes		

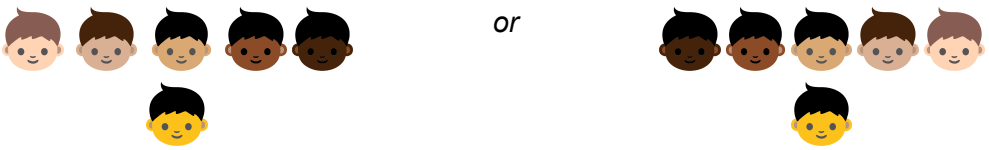
	No	⛽ + 🟫	⛽□

As noted above at the end of *Section 2.4, Diversity,* emoji presentation selectors are neither needed nor recommended for use in emoji modifier sequences. See *ED-13. emoji modifier sequence*. However, older data may include *defective* emoji modifier sequences which do include emoji presentation selectors.

2.4.2 Emoji Modifiers in Text

For input, the composition of an emoji sequence does not need to be apparent to the user: it appears on the screen as a single image. On a phone, for example, a long press on a human figure can bring up a minipalette of different skin tones, without the user having to separately find the human figure and then the modifier. The following shows some possible appearances:

Minipalettes



Of course, there are many other types of diversity in human appearance besides different skin tones: Different hair styles and color, use of eyeglasses, various kinds of facial hair, different body shapes, different headwear, and so on. It is beyond the scope of Unicode to provide an encoding-based mechanism for representing every aspect of human appearance diversity that emoji users might want to indicate. The best approach for communicating very specific human images—or any type of image in which preservation of specific appearance is very important—is the use of embedded graphics, as described in Longer Term Solutions.

2.5 Emoji ZWJ Sequences

The U+200D ZERO WIDTH JOINER (ZWJ) can be used between the elements of a sequence of characters to indicate that a single glyph should be presented if available. An implementation uses this mechanism to handle such an emoji ZWJ sequence as a single glyph, with a palette or keyboard that generates the appropriate sequences for the glyphs shown. To the user of such a system, these behave like single emoji characters, even though internally they are sequences.

When an emoji ZWJ sequence is sent to a system that does not have a corresponding single glyph, the ZWJ characters are ignored and a fallback sequence of separate emoji is displayed. Thus an emoji ZWJ sequence should only be defined and supported by implementations where the fallback sequence would also make sense to a recipient.

For example, the following are possible displays:

ZWJ Sequence Display

Sequence	Display	Combined glyph?
👩 ZWJ ❤️ ZWJ 👩	👩‍❤️‍👩	Yes
	👩❤️👩	No

See also the **Emoji ZWJ Sequences** [emoji-charts].

The use of ZWJ sequences may be difficult in some implementations, so caution should be taken before adding new sequences.

For recommendations on the use of variation selectors in ZWJ sequences, see *Section 2.7, Emoji Implementation Notes* below.

2.6 Multi-Person Groupings

There are several emoji that depict more than one person interacting. When implemented with a choice of genders or skin tones, special handling is required on a case-by-case basis. These emoji are listed below:

Multi-Person Groupings

Hex	Char	CLDR Name
U+1F91D		handshake
U+1F46F		people with bunny ears
U+1F93C		people wrestling
U+1F46B		woman and man holding hands
U+1F46C		men holding hands
U+1F46D		women holding hands
U+1F48F		kiss
U+1F491		couple with heart

U+1F46A *family* is a similar case that also requires special consideration: see section 2.6.1 for further discussion.

There are some other emoji that would share the same gender and skin tone, such as folded hands. As far as gender and skin tone are concerned, these behave just like a single person and so need no special treatment. Other examples include:

- For U+1F486 *person getting massage*, the hands of the person providing the massage should be depicted with no skin tone showing, perhaps in gloves.
- For the following emoji and their skin-tone variants, the infant should be depicted with no skin tone showing, perhaps covered in a blanket, so that the emoji is treated as a single person for purposes of skin tone modification:
  - U+1F931 *breast-feeding*
  - U+1F469 U+200D U+1F37C *woman feeding baby*
  - U+1F468 U+200D U+1F37C *man feeding baby*
  - U+1F9D1 U+200D U+1F37C *person feeding baby*

2.6.1 Multi-Person Gender

The emoji for multi-person groupings have unspecified gender (unless modified) with the exception of the three characters for people holding hands. The handshake itself does not provide for gender differences.

Family sequences can depict combinations of one or two adults along with one or two children. RGI sequences allow specifying gender of family members but vendors are encouraged to use maximally-generic depictions of families, such as silhouettes; visible gender distinctions are not required. In addition, gendered family sequences need not be available for input and any such sequence may be treated the same as the corresponding non-gendered sequence.

Non-Gendered Family Sequences

Description	Sample Display	Internal Representation
family: adult, child	👪	👨 ZWJ 👧
family: adult, child, child	👪	👨 ZWJ 👧 ZWJ 👧
family: adult, adult, child	👪	👨 ZWJ 👨 ZWJ 👧
family: adult, adult, child, child	👪	👨 ZWJ 👨 ZWJ 👧 ZWJ 👧

Gender is applied to other multi-person groupings by using either a sign format type ZWJ sequence as described in the table Gender Appearance Mechanisms, or ZWJ sequences with MAN, WOMAN, and ADULT. The data files list the RGI versions of these, such as the following:

Examples of Gender with Multi-Person Groupings

Description	Sample Display	Internal Representation
men with bunny ears	👯	👯 ZWJ ♂
women with bunny ears	👯	👯 ZWJ ♀
kiss: woman, man	👩‍❤️‍💋‍👨	👩 ZWJ ❤ ZWJ 💋 ZWJ 👨
couple with heart: man, man	👨‍❤️‍👨	👨 ZWJ ❤ ZWJ 👨

2.6.2 Multi-Person Skin Tones

As with gender, skin tones can be applied to multi-person groupings in a similar manner. Emoji represented internally by sequences may have skin tone modifiers (**Emoji_Modifier** characters) added after each of the characters that take them (those with **Emoji_Modifier_Base**). This is illustrated by the table Examples of Skin Tones for Multi-Person Groupings Using RGI Sequences below.

Multi-person sequences that mix people characters without skin tones and people characters with skin tones should not be generated. That is, for an input system, if one person character in a multi-person emoji sequence has a skin tone modifier, then all people characters in that sequence should have skin tone modifiers.

In Emoji 12.0, the Emoji_Modifier_Base property, emoji modifier sequences and RGI ZWJ sequences were updated to add 25 skin tone combinations for woman and man holding hands, and 15

combinations each for women holding hands, men holding hands, and people holding hands. These sequences appear as 70 different images.

In Emoji 12.1, the RGI ZWJ sequences for women holding hands, men holding hands, and people holding hands were further updated to add 10 more sequences each, so their sequences correspond to those for woman and man holding hands. The new sequences are for people of different skin tones, but with the darker skin tone later in the sequence instead of earlier. For example:

> Emoji 12.0 sequence: 1F468 **1F3FD** 200D 1F91D 200D 1F468 **1F3FB** ; men holding hands: **medium** skin tone, **light** skin tone
>
> Emoji 12.1 addition: 1F468 **1F3FB** 200D 1F91D 200D 1F468 **1F3FD** ; men holding hands: **light** skin tone, **medium** skin tone

The only difference between the above sequences is that the inferred positions of the medium-skin-tone man and the light-skin-tone man are swapped, left and right.

Implementations can use the same image for both sequences. For the multi-person emoji, implementations are not required to have different images for people of the same gender depending solely on position. The choice of whether to do so may depend on design considerations specific to particular vendor images.

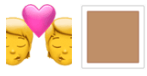
Other multi-person groups with different skin tone combinations can be represented as valid sequences, but not all such sequences are RGI. The following table provides examples of RGI sequences for multi-person groupings with skin-tone modifications.

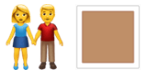
Examples of Skin Tones for Multi-Person Groupings Using RGI Sequences

Description	Internal Representation
women holding hands: medium, dark skin tones	
people holding hands: medium, dark skin tones	
handshake: medium, dark skin tones	

Skin tone modifiers can be applied to each of the eight characters listed in the table **Multi-Person Groupings**, and as of Emoji 18.0 all of the resulting emoji modifier sequences have RGI status; examples for some of these characters are illustrated in the following table. This gives all of the people in the group the same skin tone, which is similar to how the gender marker works.

Examples of Skin Tones for Multi-Person Groupings Using Single Characters

Description	Internal Representation
kiss: medium skin tone	
handshake: medium skin tone	
people with bunny ears: medium skin tone	
women with bunny ears: medium skin tone	

woman and man holding hands: medium skin tone	

2.7 Emoji Implementation Notes

This section describes important implementation features of emoji, including the use of emoji and text presentation selectors, how to do segmentation, and handling of tag characters.

2.7.1 Emoji and Text Presentation Selectors

This section describes where the emoji presentation selectors can be used. The text presentation selector only occurs in text presentation sequences, which are not displayed as emoji.

Characters	Variation / Behavior
emoji character	~~*may*~~ *should* have an emoji or text presentation selector added if the result is a valid *emoji presentation sequence* or *text presentation sequence*
	~~*should* have an emoji presentation selector added if Emoji_Presentation=No whenever an emoji presentation is desired~~
emoji flag sequence	*does not* contain an emoji or text presentation selector
	should be displayed with an emoji presentation by default
emoji modifier sequence	*does not* contain an emoji or text presentation selector
	should be displayed with an emoji presentation by default, whether or not the modifier base has Emoji_Presentation=Yes • Implementations *may* choose to support old data that contains *defective* emoji_modifier_sequences, that is, having emoji presentation selectors.
emoji ZWJ sequence	*may have* an emoji presentation selector The recommended behavior is: **User Input:** • only fully-qualified emoji ZWJ sequences should be generated by keyboards and other user input devices. **Processing and Display:** • fully-qualified emoji ZWJ sequences should be handled appropriately in processing, such as display, editing, segmentation, and so on. • minimally-qualified or unqualified emoji ZWJ sequences may be handled in the same way as their fully-qualified forms; the choice is up to the implementation. A text presentation selector applied to any element of an emoji ZWJ sequence breaks that sequence, preventing it from displaying as a single image. The partial sequences should be displayed as separate images, each with presentation style as specified by any presentation selectors present, or by default style for those emoji that do not have any variation selectors.

2.7.2 Handling Tag Characters

The properties for tag characters U+E0020..U+E007F (TAG SPACE..CANCEL TAG) have been modified for use in indicating variants or extensions of emoji characters. For detailed information on handling TAG sequences correctly, see Annex C: Valid Emoji Tag Sequences.

2.8 Hair Components

Emoji Version 11.0 introduced hair components, which can be used in ZWJ sequences to indicate hair colors or styles. The sequences recommended for general interchange (RGI) are listed in the data files. The components include:

- Red-haired (ginger)
- Curly-haired
- White-haired
- Bald

There are hundreds of possible distinctions among hair colors and styles, but to limit the number of combinations—and because emoji are presented with a “cartoon” style—there is a small number of hair components. Note that the hair color blond has already been provided for by an explicit blond man/woman/person emoji. Brown/black-haired are already typical defaults for hair color in human-form emoji.

2.9 Color

Nine large colored square emoji may be used in ZWJ sequences to indicate that a base emoji should be displayed with that color if possible. The color of the resulting image may not be exactly the same as the color square. The color squares used for this purpose are:

- U+2B1B BLACK LARGE SQUARE
- U+2B1C WHITE LARGE SQUARE
- U+1F7E5 LARGE RED SQUARE … U+1F7EB LARGE BROWN SQUARE

Where the implementation does not provide a single emoji image in that color, the user should see the fallback appearance showing an indication of the desired color. Where color ZWJ sequences are supported and the base emoji already has that color, the color square should be ignored.

Emoji Glyph Color Examples

	Internal Representation			Sample Display	Fallback Appearance
black cat					
	U+1F408	U+200D	U+2B1B		
orange cat					
	U+1F408	U+200D	U+1F7E7		

The squares require a ZWJ; they do not behave like the five skin-tone modifiers listed in Emoji Modifiers.

The *white square* emoji is often presented as a light gray, to set it off from white backgrounds.

In Emoji Version 18.0 there are four RGI emoji ZWJ sequences of this form.

2.10 Emoji Glyph Facing Direction

Emoji with glyphs that face to the right or left may face either direction, according to vendor practice. However, that inconsistency can cause a change in meaning when exchanging text across platforms. The following ZWJ mechanism can be used to explicitly indicate direction. If the base emoji image is not available facing in that direction, the user should see the fallback appearance showing an indication of the desired direction. If direction ZWJ sequences are supported and the base emoji already faces that direction, the direction emoji should be ignored.

Emoji Glyph Direction Examples

Internal Representation			Intended Display	Fallback Appearance
🏃	ZWJ	⬅️	🏃	🏃⬅️
U+1F3C3	U+200D	U+2B05 U+FE0F		
🏃	ZWJ	➡️	🏃	🏃➡️
U+1F3C3	U+200D	U+27A1 U+FE0F		

A direction RGI sequence can also exist for emoji where there is no inconsistency across vendors: in this case there will be an RGI sequence for only one direction; vendors may choose to handle the non-RGI sequence for the opposite direction (corresponding to the unmodified emoji) to suppress the arrow of the fallback appearance.

In Emoji Version 18.0 there are 108 RGI emoji ZWJ sequences of this form.

2.11 Order of Emoji ZWJ Sequences

When representing emoji ZWJ sequences for an individual person, the following order should be used:

Order	Category	Section
1	Base	*Section 1.4.1, Emoji Characters*
2	Emoji modifier or emoji presentation selector	*Section 2.4, Diversity*
3	Hair component	*Section 2.8, Hair Component*
4	Color	*Section 2.9, Color*
5	Gender sign or object	*Section 2.3.1, Gender-Neutral Emoji*
6	Direction indicator	*Section 2.10, Emoji Glyph Facing Direction*

3 Which Characters are Emoji

There are different ways to count the emoji in Unicode, especially because an emoji sequence may display as a single emoji image. The following provides an overview of the ways to count emoji; it can be (for example):

- The count of code points that can be used in emoji, though this includes some code points that are only used as part of sequences and don't have emoji appearance by themselves;
- All sequences of one or more characters that can appear as a single glyph (which is probably closer to what users think of as the number of emoji), though typically only a subset of possible sequences are displayed as a single glyph on any platform, and some sequences may be platform-specific extensions.

It is recommended that any font or keyboard whose goal is to support Unicode emoji should support the characters and sequences listed in the [emoji-data] data files. The best definition of the full set is in the **emoji-test.txt** file [emoji-data].

Emoji Counts [emoji-charts] provides more detail about the various counts as of the current version of this specification. The various column and row headers are described in Emoji Counts Key.

- The "**Subtotal**" row in the chart indicates the count of what users typically think of as emoji. For example, the 26 Regional Indicator (RI) code points are not included there; even though they have Emoji status, they are typically only used in pairs to represent flags.

- Typical keyboards may normally present even fewer emoji, since they may use mechanisms like a long press to display modifier sequences for specific emoji, and would thus not simultaneously display all of the images associated with the chart rows that count emoji with explicit skin tones.

Separate [emoji-charts] provide more information on many of these subsets and others.

4 Presentation Style

Certain emoji have defined variation sequences, in which an emoji character can be followed by an invisible emoji presentation selector or text presentation selector.

This capability was added in Unicode 6.1. Some systems may also provide this distinction with higher-level markup, rather than variation sequences. For more information on these selectors, see **Emoji Presentation Sequences** [emoji-charts]. For details regarding the use of emoji or text presentation selectors in emoji sequences specifically, see *Section 2.7, Emoji Implementation Notes*.

Implementations should support both styles of presentation for the characters with emoji and text presentation sequences, if possible. Most of these characters are emoji that were unified with preexisting characters. Because people are now using emoji presentation for a broader set of characters, Unicode 9.0 added emoji and text presentation sequences for all emoji with default text presentation (see discussion below). These are the characters shown in the column labeled “Default Text Style; no VS in U8.0” in the **Text vs Emoji** chart [emoji-charts].

However, even for cases in which the emoji and text presentation selectors are available, it had not been clear for implementers whether the *default* presentation for pictographs should be emoji or text. That means that a piece of text may show up in a different style than intended when shared across platforms. While this is all perfectly legitimate for Unicode characters—*presentation style is never guaranteed*—a shared sense among developers of when to use emoji presentation by default is important, so that there are fewer unexpected or jarring presentations. Implementations need to know what the generally expected default presentation is, to promote interoperability across platforms and applications.

There had been no clear line for implementers between three categories of Unicode characters:

1. **emoji-default:** those expected to have an emoji presentation by default, but can also have a text presentation
2. **text-default:** those expected to have a text presentation by default, but could also have an emoji presentation
3. **text-only:** those that should only have a text presentation

These categories can be distinguished using properties listed in Annex A: Emoji Properties and Data Files. The first category are characters with **Emoji=Yes** and **Emoji_Presentation=Yes**. The second category are characters with **Emoji=Yes** and **Emoji_Presentation=No**. The third category are characters with **Emoji=No**.

The presentation of a given emoji character depends on the environment, whether or not there is an emoji or text presentation selector, and the default presentation style (emoji versus text). In informal environments like texting and chats, it is more appropriate for most emoji characters to appear with a colorful emoji presentation, and only get a text presentation with a text presentation selector. Conversely, in formal or technical environments ~~such as word processing~~, it is generally better for emoji characters to appear with a text presentation, and only get the colorful emoji presentation with the emoji presentation selector. The format or metadata of a document may also be a signal that it was created prior to the introduction of emoji, in which case defaulting to text presentation is appropriate and defaulting to emoji presentation would be inappropriate.

Based on those factors, here is typical presentation behavior. However, these guidelines may change with changing user expectations.

Emoji versus Text Display

Example Environment	With Emoji Presentation Selector	With Text Presentation Selector	With Neither	
			text-default	emoji-default
~~word processing~~ formal or technical	✏️	✏	✏	✏
~~plain~~ web pages	✏️	✏	✏	✏️
texting, chats	✏️	✏	✏️	✏️

Computer languages use the Pattern_Syntax property to identify code points that have been reserved for syntactic use. Some of the code points with the Pattern_Syntax property have default emoji presentation. When emoji are used as part of computer language syntax, text presentation sequences can be used to unambiguously express that they should be displayed and interpreted as syntactic characters, rather than emoji. See *Section 7.2, Emoji Profile*, in Unicode Standard Annex #31, “Unicode Identifiers and Syntax” [UAX31].

4.1 Emoji and Text Presentation Selectors

Every emoji character with a default text presentation allows for an emoji or text presentation selector. Thus the presentation of these characters can be controlled on a character-by-character basis. The characters that can have these selectors applied to them are listed in **Emoji Presentation Sequences** [emoji-charts].

In addition, the next two sections describe two other mechanisms for globally controlling the emoji presentation: using language tags with locale extensions, or using special script codes. Though these are new mechanisms and not yet widely supported, vendors are encouraged to support the locale extension for most general usage such as in browsers; the special script codes may be appropriate for more specific usage such as OpenType font selection, or in APIs. For more information, see [CLDR].

4.2 Emoji Locale Extension

The locale extension “-em” can be used to specify desired presentation for characters that may have both text-style and emoji-style presentations available. There are three values that can be used, here illustrated with “sr-Latn”:

Locale Code	Description
sr-Latn-u-em-**emoji**	use an emoji presentation for emoji characters where possible
sr-Latn-u-em-**text**	use a text presentation for emoji characters where possible
sr-Latn-u-em-**default**	use the default presentation (only needed to reset an inherited -em setting).

This can be used in HTML, for example, with `<html lang="sr-Latn-u-em-emoji">`. Note that this approach does not have the disadvantages listed below for the script-tag approach.

4.3 Emoji Script Codes

Two script subtags can be used to control the presentation style. These use script codes defined by ISO 15924 but given more specific semantics by CLDR, see unicode_script_subtag:

- Zsye—prefer emoji style for characters that have both text and emoji styles available.
- Zsym—prefer text style for characters that have both text and emoji styles available.

These script codes are not suitable for use in general language tags:

- They cannot be used with language-script combinations; for example, if the language is sr-Latn (Serbian in Latin script), then Zsye cannot be used.
- They may confuse processes that depend on language tags, such as spell checkers.

However, they may be useful by themselves in specific contexts such as OpenType font selection, or in APIs that take script codes.

4.4 Other Approaches for Control of Emoji Presentation

Other approaches for control of emoji presentation are also in use. For example, in some CSS implementations, if any font in the lookup list is an emoji font, then emoji presentation is used whenever possible.

5 Ordering and Grouping

Neither the Unicode code point order, nor the default collation provided by the Unicode Collation Algorithm (DUCET), are currently well suited for emoji, because they separate conceptually-related characters. From the user's perspective, the ordering in the following selection of characters sorted by DUCET appears quite random, as illustrated by the following example:

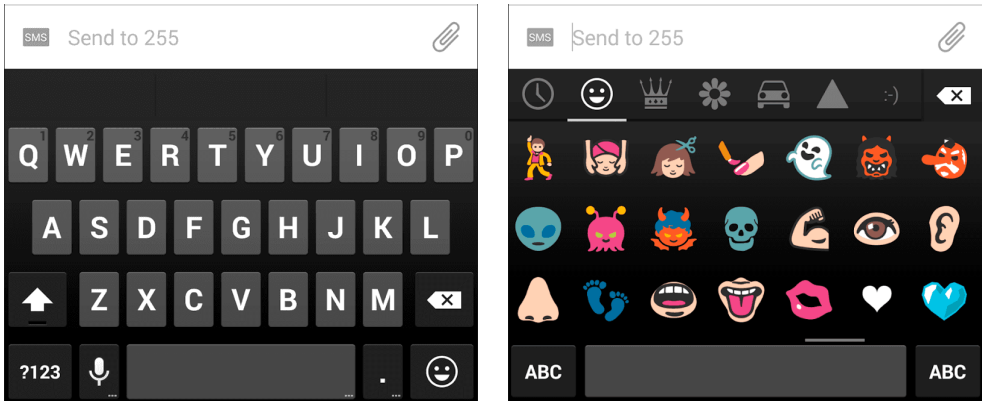
Emoji Ordering [emoji-charts] shows an ordering for emoji characters that groups them together in a more natural fashion. This data has been incorporated into [CLDR].

This ordering presents a cleaner and more expected ordering for sorted lists of characters. The groupings include: faces, people, body-parts, emotion, clothing, animals, plants, food, places, transport, and so on. The ordering also groups more naturally for the purpose of selection in input palettes. However, for sorting, each character must occur in only one position, which is not a restriction for input palettes. See *Section 6, Input*.

6 Input

Emoji are not typically typed on a keyboard. Instead, they are generally picked from a palette, or recognized via a dictionary. The mobile keyboards typically have a ☺ button to select a palette of emoji, such as in the left image below. Clicking on the ☺ button reveals a palette, as in the right image.

Palette Input



The palettes need to be organized in a meaningful way for users. They typically provide a small number of broad categories, such as People, Nature, and so on. These categories typically have 100-200 emoji.

Many characters can be categorized in multiple ways: an orange is both a plant and a food. Unlike a sort order, an input palette can have multiple instances of a single character. It can thus extend the sort ordering to add characters in any groupings where people might reasonably be expected to look for them.

More advanced palettes will have long-press enabled, so that people can press-and-hold on an emoji and have a set of related emoji pop up. This allows for faster navigation, with less scrolling through the palette.

Annotations for emoji characters are much more finely grained keywords. They can be used for searching characters, and are often easier than palettes for entering emoji characters. For example, when someone types “hourglass” on their mobile phone, they could see and pick from either of the matching emoji characters ⏳ or ⌛. That is often much easier than scrolling through the palette and visually inspecting the screen. Input mechanisms may also map *emoticons* to emoji as keyboard shortcuts: typing :-) can result in 😄.

In some input systems, a word or phrase bracketed by colons is used to explicitly pick emoji characters. Thus typing in “I saw an *:ambulance:*” is converted to “I saw an 🚑”. For completeness, such systems might support all of the full Unicode names, such as *:first quarter moon with face:* for 🌛. Spaces within the phrase may be represented by _, as in the following:

“my *:alarm_clock:* didn’t work”

“my ⏰ didn’t work”.

However, in general the full Unicode names are not especially suitable for that sort of use; they were designed to be unique identifiers, and tend to be overly long or confusing.

For emoji that have gender and/or skin tone variants, input systems should fully specify the intended appearance, rather than relying on a particular system’s default appearance; see for example *Section 2.3.2, Marking Gender in Emoji Input*.

7 Searching

Searching includes both searching for emoji characters in queries, and finding emoji characters in the target. These are most useful when they include the annotations as synonyms or hints. For example, when someone searches for ⛽ on a recommendations site, they see matches for “gas station”. Conversely, searching for “gas pump” in a search engine could find pages containing ⛽. Similarly, searching for “gas pump” in an email program can bring up all the emails containing ⛽.

There is no requirement for uniqueness in both palette categories and annotations: an emoji should show up wherever users would expect it. A gas pump ⛽ might show up under “object” and “travel”; a heart 💔 under “heart” and “emotion”, a 😻 under “animal”, “cat”, and “heart”.

Annotations are language-specific: a German user would expect a search for ⛽ to result in matches for “Tankstelle”. Thus annotations need to be in multiple languages to be useful across languages. They should also include regional annotations within a given language, like “petrol station” for British English. An English annotation cannot simply be translated into different languages, because different words may have different associations in different languages. The emoji 🌵 may be associated with Mexican or Southwestern restaurants in the US, but not be associated with them in, say, Greece.

As noted in *Section 2.1 Names*, there is one further kind of annotation, called a CLDR short name. This is also referred to as the *TTS name*, for use in text-to-speech processing such as providing a short, descriptive emoji name when reading text for accessibility purposes. In this case the CLDR names provide several advantages over formal Unicode character names:

- They can be shorter and less cumbersome than the formal name, whose requirement for name uniqueness often results in names that are overly long, such as *BLACK RIGHT-POINTING TRIANGLE WITH DOUBLE VERTICAL BAR* for ⏯.
- They can apply to emoji that are represented by sequences as well as those represented by single characters.
- They can be updated to better reflect current emoji depictions and usage.

TTS names are also outside the current scope of this document.

8 Longer Term Solutions

The longer-term goal for implementations should be to support embedded graphics, in addition to the emoji characters. Embedded graphics allow arbitrary emoji symbols, and are not dependent on additional Unicode encoding. Some examples of this are found in Skype and LINE.

However, to be as effective and simple to use as emoji characters, a full solution requires significant infrastructure changes to allow simple, reliable input and transport of images (stickers) in texting, chat, mobile phones, email programs, virtual and mobile keyboards, and so on. (Even so, such images will never interchange in environments that only support plain text, such as email addresses.) Until that time, many implementations will need to use Unicode emoji instead.

For example, mobile keyboards need to be enhanced. Enabling embedded graphics would involve adding an additional custom mechanism for users to add in their own graphics or purchase additional sets, such as a ➕ sign to add an image to the palette above. This would prompt the user to paste or otherwise select a graphic, and add annotations for dictionary selection.

With such an enhanced mobile keyboard, the user could then select those graphics in the same way as selecting the Unicode emoji. If users started adding many custom graphics, the mobile keyboard might even be enhanced to allow ordering or organization of those graphics so that they can be quickly accessed. The extra graphics would need to be disabled if the target of the mobile keyboard (such as an email header line) would only accept text.

Other features required to make embedded graphics work well include the ability of images to scale with font size, inclusion of embedded images in more transport protocols, switching services and applications to use protocols that do permit inclusion of embedded images (for example, MMS versus SMS for text messages). There will always, however, be places where embedded graphics cannot be used—such as email headers, SMS messages, or file names. There are also privacy aspects to implementations of embedded graphics: if the graphic itself is not packaged with the text, but instead is just a reference to an image on a server, then that server could track usage.

Annex A: Emoji Properties and Data Files

The following binary character properties are available for emoji characters.

Emoji Character Properties

Property	Abbr	Property Values
Emoji	Emoji	**=Yes** for characters that are emoji
Emoji_Presentation	EPres	**=Yes** for characters that have emoji presentation by default
Emoji_Modifier	EMod	**=Yes** for characters that are emoji modifiers
Emoji_Modifier_Base	EBase	**=Yes** for characters that can serve as a base for emoji modifiers
Emoji_Component	EComp	**=Yes** for characters used in emoji sequences that normally do not appear on emoji keyboards as separate choices, such as keycap base characters or Regional_Indicator characters. All characters in emoji sequences are either **Emoji** or **Emoji_Component**. Implementations must not, however, assume that all **Emoji_Component** characters are also **Emoji**.

		There are some non-emoji characters that are used in various emoji sequences, such as tag characters and **ZWJ**.
Extended_Pictographic	ExtPict	**=Yes** for characters that are used to future-proof segmentation. The **Extended_Pictographic** characters contain all the **Emoji** characters except for some **Emoji_Component** characters.

If **Emoji=No**, then **Emoji_Presentation=No**, **Emoji_Modifier=No**, and **Emoji_Modifier_Base=No**.

A.1 Data Files

The emoji properties are specified in the emoji data files (see [emoji-data]):

Data Files

emoji-data.txt	Property value for the properties listed in the Emoji Character Properties table
emoji-variation-sequences.txt	All permissible **emoji presentation sequences** and **text presentation sequences**
emoji-zwj-sequences.txt	ZWJ sequences used to represent emoji
emoji-sequences.txt	Other sequences used to represent emoji
emoji-test.txt	Test file for emoji characters and sequences

See [emoji-charts] for a collection of charts that have been generated from the emoji data files and the related [CLDR] emoji data (annotations and ordering). They are purely illustrative; the data to use for implementation is in [emoji-data].

The data file comments and their structure are purely informative, and may change across releases without notice. For version conventions used in the data files, see *Section 1.5.2, Versioning*.

Annex B: Valid Emoji Flag Sequences

While the syntax of a well-formed ***emoji flag sequence*** is defined in ***ED-14***, only valid sequences are displayed as flags by conformant implementations, where:

- The valid region sequences correspond to two-letter Unicode region subtags as defined in [CLDR], with idStatus = “regular”, “deprecated”, or “macroregion”. For macroregions, only **UN** and **EU** are valid.
    1. Deprecated regions are included in the list of valid region sequences so that deprecations in the future do not invalidate previously valid emoji flag sequences. RGI emoji flag sequences with deprecated regions are recommended for support. Non-RGI emoji flag sequences with deprecated regions should not be generated.
    2. Macroregion region sequences generally do not have official flags, with the exception of **UN** and **EU**.

Some region sequences represent countries (as recognized by the United Nations, for example); others represent territories that are associated with a country. Such territories may have flags of their own, or may use the flag of the country with which they are associated. Depictions of images for flags may be subject to constraints by the administration of that region.

Caveats:

- Although a pair of REGIONAL INDICATOR symbols is referred to as an emoji_flag_sequence, it really represents a specific region, not a specific flag for that region. The actual flag displayed for the pair may be different on different platforms, for example for territories which do not have an official flag. The displayed flag may change over time as regions change their flags and platforms update their software.

- For some territories (especially those without separate official flags), the displayed flag may be the same as the flag for the country with which they are associated. For more about cases where characters have the same appearance, see *UTR #36: Unicode Security Considerations* [UTR36].

For additional information see the sub-section on Regional Indicator Symbols in *Section 22.10, Enclosed and Square* of [Unicode].

B.1 Presentation

Emoji are generally presented with a square aspect ratio, which presents a problem for flags. The flag for Qatar is over 150% wider than tall; for Switzerland it is square; for Nepal it is over 20% taller than wide. To avoid a ransom-note effect, implementations may want to use a fixed ratio across all flags, such as 150%, with a blank band on the top and bottom. (The average width for flags is between 150% and 165%.) Presentation as a waving flag, or clipping to a circle, can help to present a uniform appearance, masking the aspect differences.

Flags should have a visible edge. One option is to use a one-pixel gray line chosen to be contrasting with the adjacent field color.

Options for presenting an emoji_flag_sequence for which a system does not have a specific flag or other glyph include:

- Display each REGIONAL INDICATOR symbol separately as a letter in a dotted square, as shown in the Unicode charts. This provides information about the specific region indicated, but may be mystifying to some users.
- For all unsupported REGIONAL INDICATOR pairs, display the same missing flag glyph, such as the image shown below. This would indicate that the supported pair was intended to represent the flag of some region, without indicating which one.

B.2 Ordering

The code point order of flags is by region code, which will not be intuitive for users, because that rarely matches the order of countries in the user's language. English speakers are surprised that the flag for Germany comes before the flag for Djibouti. An alternative is to present the sorted order according to the localized country name, using [CLDR] data.

Annex C: Valid Emoji Tag Sequences

While the syntax of a well-formed emoji tag sequence is defined in ***ED-14a***, not all possible tag sequences are valid. The only valid sequences in this version of Unicode Emoji are defined by sections in this annex, which specify valid combinations of <tag_base> characters and <tag_spec> sequences and their expected presentation. Conformant implementations only display valid sequences as emoji, and display invalid sequences with a special presentation to show that they are invalid, such as in the examples below.

There is one common constraint on valid emoji tag sequences: *the entire emoji_tag_sequence, including tag_base and tag_end, must not be longer than 32 code points*. This provides a practical limit needed by many rendering systems, and is consistent with the 32-code-point buffer limit specified for the Stream-Safe Text Format as defined in Unicode Standard Annex #15, "Unicode Normalization Forms" [UAX15].

If a platform supports tag sequences, but a particular emoji tag sequence is invalid or cannot be displayed, then to reduce spoofing risk that emoji tag sequence should be displayed using a *missing emoji glyph* if feasible. The following are examples, where the tag_base character is a *black flag*.

Sample images	Condition

	The implementation supports tag sequences, but this particular sequence is either not supported or simply invalid. (If the font technology permits, the missing emoji glyph can be overlaid on the `tag_base` character, thus occupying the same physical dimensions as if the sequence were supported.)
	The implementation does not support tag sequences at all. (The tag characters are normally invisible, and thus only the base character displays.)

In examples in this section, underlined ASCII characters represent the corresponding tag characters, while ✦ represents the `tag_end`.

C.1 Flag Emoji Tag Sequences

A valid flag emoji tag sequence must satisfy the following constraints:

1. The `tag_base` and `tag_spec` are limited to the following:

`tag_base`	U+1F3F4 BLACK FLAG
`tag_spec`	(U+E0030 TAG DIGIT ZERO .. U+E0039 TAG DIGIT NINE, U+E0061 TAG LATIN SMALL LETTER A .. U+E007A TAG LATIN SMALL LETTER Z)+

   `tag_end` is U+E007F CANCEL TAG, as described in ***ED-14a***.

2. Let SD be the result of mapping each character in the `tag_spec` to a character in [0-9a-z] by subtracting 0xE0000.
   1. SD must then be a specification as per [CLDR] of either a Unicode subdivision_id or a 3-digit unicode_region_subtag, and
   2. SD must have CLDR idStatus = “regular”, “deprecated”, or “macroregion”.

Notes:

1. Deprecated SD values are included in the list of valid region sequences so that deprecations in the future do not invalidate previously valid emoji tag sequences. RGI emoji tag sequences with deprecated SD values are recommended for support. Non-RGI emoji tag sequences with deprecated SD values should not be generated.
2. There is no hyphen in the `tag_spec`, unlike ISO subdivisions like “GB-SCT”.
3. These flag emoji tag sequences are used to request an image for whatever is currently the flag of the specified subregion. Like the emoji flag sequences, they are not intended to provide a mechanism for *versioned* representations of any particular flag image.
4. Specific platforms and programs decide which emoji extended flag sequences they will support. There is no requirement that any be supported, and no expectation that more than a small number be commonly supported by vendors.
5. Note that SD cannot be a two-letter code like “US” or “us”.

C.1.1 Sample Valid Emoji Tag Sequences

A completely tag-unaware implementation will display any sequence of tag characters as invisible, without any effect on adjacent characters. The following sections apply to conformant implementations that support at least one tag sequence.

An implementation may support emoji tag sequences, but not support a particular valid emoji tag sequence.

Images for unsupported valid emoji tag sequences must indicate that the sequence image is missing, by showing the base glyph with either a following “missing emoji glyph” or with an overlay “missing” glyph. The overlay glyph approach is recommended, so that the sequence would have the same width as if supported. A tag-unaware implementation (TU) will show just the base character.

Display of Valid Emoji Tag Sequences

Sequence	Sample Images			Comments	RGI sequence?
	Supported	Unsupported	TU		
🏴gbeng✦					England	**Yes**
🏴gbsct✦					Scotland	**Yes**
🏴gbwls✦					Wales	**Yes**
🏴usca✦					California	No
🏴caon✦					Ontario	No
🏴chzh✦					Canton Zürich	No
🏴frnor✦					Normandy	No

C.1.2 Sample Invalid Emoji Tag Sequences

Images for invalid (but well-formed) emoji tag sequences must not be interpreted as if they were regular emoji tag sequences for a different appearance. They must instead indicate that there is something wrong with the sequence. The recommended approach is to also show the base glyph with either a following “missing emoji glyph” or with an overlay “missing” glyph.

Display of Invalid Emoji Tag Sequences

Sequence	Rec. Images		TU	Comments
🏴ushuh✦				*Incorrect subregion with “us” region*
🏴uksct✦				*No “uk” region so incorrect subregion*
👨usca✦				*Base invalid for flag tag emoji sequence*
👨olvikan✦				*Invalid base and tag_spec — not conformant to show as a “demon” or other non-missing image*

C.1.3 Sample Ill-formed Emoji Tag Sequences

Images for an ill-formed tag sequence should indicate that there is something wrong with the sequence. The recommended approach is to show the ill-formed tag sequence as a “missing emoji glyph”.

Display of Ill-formed Emoji Tag Sequences

Sequence	Rec. Images	TU	Comments
Ausca✦		A	*No emoji base*

usca✦			*No base*
usca			*No terminator*
usca			*No base, no terminator*

Acknowledgments

Mark Davis and Peter Edberg created the initial versions of this document and maintained the text for many versions. Mark Davis and Ned Holbrook now maintain the text.

Thanks to Shervin Afshar, Julie Allen, Deborah Anderson, Rachel Been, Nicole Bleuel, Charlotte Buff, Jeremy Burge, Mathias Bynens, Charles Carson, Christopher Chapman, Chenjintao (陈锦涛), Chenshiwei, Michele Coady, Peter Constable, David Corbett, Craig Cummings, Jennifer Daniel, Kamilé Demir, Monica Dinculescu, Behnam Esfahbod, Doug Ewell, Kara Fong, Agustin Fonts, Asmus Freytag, Claudia Galvan, Andrew Glass, Seb Grubb, Bryan Haggerty, Casey Henson, Paul Hunt, Olli Jones, Tayfun Karadeniz, Hiroyuki Komatsu, Mike LaJoie, Jennifer 8. Lee, Robin Leroy, Norbert Lindenberg, Ken Lunde, Gwyneth Marshall, Rick McGowan, Katsuhiko Momoi, Lisa Moore, Sarah Neufeld, Katsuhiro Ogata, Christoph Päper, Katrina Parrott, Michelle Perham, Addison Phillips, Roozbeh Pournader, Alexander Robertson, Judy Safran-Aasen, Markus Scherer, Alolita Sharma, Jane Solomon, Sean Stewart, Michel Suignard, Richard Tunnicliffe, Yifán Wáng, Wilder Wells, and Ken Whistler for feedback on and contributions to this document and related data and charts, including earlier versions.

Thanks to Adobe / Paul Hunt, Apple, Emojination, EmojiOne, Emojipedia, EmojiXpress, Michael Everson, Facebook, Google, iDiversicons, Microsoft, Samsung, and Twitter for supplying images for illustration in this document, or earlier versions of this document.

Rights to Emoji Images

The content for this section, discussing rights and acknowledgments, has been moved to Emoji Images and Rights.

References

[CLDR] CLDR - Unicode Common Locale Data Repository
https://cldr.unicode.org/
For the latest version of the associated specification (LDML), see:
https://www.unicode.org/reports/tr35/

[emoji-charts] The illustrative charts of emoji.
For the 18.0 versions, see:
https://www.unicode.org/emoji/charts-18.0/
For the latest versions, see:
https://www.unicode.org/emoji/charts/

[emoji-data] The associated data files for emoji characters.
For the 18.0 versions, see:
https://www.unicode.org/Public/18.0.0/ucd/emoji/emoji-data.txt
https://www.unicode.org/Public/18.0.0/ucd/emoji/emoji-variation-sequences.txt
https://www.unicode.org/Public/18.0.0/emoji/emoji-sequences.txt
https://www.unicode.org/Public/18.0.0/emoji/emoji-zwj-sequences.txt
https://www.unicode.org/Public/18.0.0/emoji/emoji-test.txt
For the latest versions, see:
https://www.unicode.org/Public/UCD/latest/ucd/emoji/emoji-data.txt
https://www.unicode.org/Public/UCD/latest/ucd/emoji/emoji-variation-sequences.txt
https://www.unicode.org/Public/emoji/latest/emoji-sequences.txt
https://www.unicode.org/Public/emoji/latest/emoji-zwj-sequences.txt
https://www.unicode.org/Public/emoji/latest/emoji-test.txt

[JSources] The UCD sources for the JCarrier symbols
For the latest version, see:
https://www.unicode.org/Public/UCD/latest/ucd/EmojiSources.txt

[UAX14] UAX #14: Unicode Line Breaking Algorithm
https://www.unicode.org/reports/tr14/

[UAX15] UAX #15: Unicode Normalization Forms
https://www.unicode.org/reports/tr15/

[UAX29] UAX #29: Unicode Text Segmentation
https://www.unicode.org/reports/tr29/

[UAX31] UAX #31: Unicode Identifiers and Syntax
https://www.unicode.org/reports/tr31/

[Unicode] The Unicode Standard
For the latest version, see:
https://www.unicode.org/versions/latest/

[UTR36] UTR #36: Unicode Security Considerations
https://www.unicode.org/reports/tr36/

[UTS18] UTS #18: Unicode Regular Expressions
https://www.unicode.org/reports/tr18/

[UTR23] UTR #23: The Unicode Character Property Model
https://www.unicode.org/reports/tr23/

Modifications

The following summarizes modifications from the previous revision of this document.

Revision 30

- *General*
  - Updated versions and dates throughout.
- *Section 2.4.1 Implementations*
  - Changed example.
- *Section 2.7.1 Emoji and Text Presentation Selectors*
  - Revised guidance for emoji characters.
- *Section 4 Presentation Style*
  - Revised guidance for default presentation style.

~~**Revision 29**~~

- ~~*Section 1 Introduction*~~
  - ~~Link directly to referenced L2 documents.~~
- ~~*Section 1.4.6 Emoji Sets*~~
  - ~~Added Standalone_Component to ED-28. RGI_Emoji_Qualification.~~
- ~~*Section 1.5.2 Versioning*~~
  - ~~Added version 17.0.~~
- ~~*Section 2.6.1 Multi-Person Gender*~~
  - ~~Consolidated discussion and examples of multi-person groupings.~~
- ~~*Section 2.6.2 Multi-Person Skin Tones*~~
  - ~~Updated for version 17.0.~~
  - ~~Added examples of newly-RGI sequences to table.~~
- ~~*Section 2.9 Color*~~
  - ~~Updated for version 17.0.~~
- ~~*Section 2.10 Emoji Glyph Facing Direction*~~

- ~~Updated for version 17.0.~~
- ~~References~~
  - ~~Changed file versions and directory structure for version 17.0.~~

Modifications for prior versions can be found in those prior versions.

© 2001–2026 Unicode, Inc. This publication is protected by copyright, and permission must be obtained from Unicode, Inc. prior to any reproduction, modification, or other use not permitted by the Terms of Use. Specifically, you may make copies of this publication and may annotate and translate it solely for personal or internal business purposes and not for public distribution, provided that any such permitted copies and modifications fully reproduce all copyright and other legal notices contained in the original. You may not make copies of or modifications to this publication for public distribution, or incorporate it in whole or in part into any product or publication without the express written permission of Unicode.

Use of all Unicode Products, including this publication, is governed by the Unicode Terms of Use. The authors, contributors, and publishers have taken care in the preparation of this publication, but make no express or implied representation or warranty of any kind and assume no responsibility or liability for errors or omissions or for consequential or incidental damages that may arise therefrom. This publication is provided “AS-IS” without charge as a convenience to users.

Unicode and the Unicode Logo are registered trademarks of Unicode, Inc., in the United States and other countries.