

**Re: Fixes to emoji data files for v12.0**

**From: ESC**

**Date: 2018-06-13 (updated 2018-07-17)**

---

The following are suggested fixes for issues in the emoji data files (and related fixes to UTS #51). Most of them are fixes for problems reported by Yifán Wáng and Mathias Bynens.

## Data Files

### [emoji-sequences.txt](#)

1. Fix the header

```
# type_field: any of {Emoji_Combining_Sequence, Emoji_Flag_Sequence,  
Emoji_Modifier_Sequence}
```

⇒

```
# type_field:  
#   Emoji_Keycap_Sequence  
#   Emoji_Flag_Sequence  
#   Emoji_Tag_Sequence  
#   Emoji_Modifier_Sequence
```

2. In the subheaders, replace space by \_ in the type names.

```
#   Emoji Keycap Sequence: ...  
#   Emoji Flag Sequence: ...  
#   Emoji Tag Sequence: ...  
#   Emoji Modifier Sequence: ...
```

### [emoji-zwj-sequences.txt](#)

1. In the subheaders, replace space by \_ in the type names.

```
# Emoji ZWJ Sequence: ...
```

⇒

```
# Emoji_ZWJ_Sequence: ...
```

### [emoji-test.txt](#)

1. Add “component” and “initially-qualified” as status values.
2. Include the 5 skin-tone modifiers in the file (currently we include hair, but exclude skin tones). Both shouldn’t be on the keyboard, but should have emoji presentation.
3. Give any components (that is, hair components and modifier components) the status “component”.

4. Change Status “non-fully-qualified” to “initially-qualified” when the first character has `Emoji_Presentation` or starts a valid emoji modifier sequence or emoji presentation sequence
  - a. Some classes of implementations can support the initially qualified but not non-fully-qualified. The difference is that from the first two code points, you already know that it is emoji, so you don’t have to “backtrack” later.
5. Fix the header in accordance with the changes, as below.

```
# Format
#   Code points; status # emoji name
#       Status
#           fully-qualified – see “Emoji Implementation Notes” in UTS #51
#           non-fully-qualified – see “Emoji Implementation Notes” in UTS #51
# Notes:
#   • This currently omits the 12 keycap bases, the 5 modifier characters, and 26 singleton
#     Regional Indicator characters
#   • The file is in CLDR order, not codepoint order. This is recommended (but not required!)
#     for keyboard palettes.
#   • The groups and subgroups are purely illustrative. See the Emoji Order chart for more
#     information.
⇒
# Format: code points; status # emoji name
#       Code points – list of one or more hex code points, separated by spaces
#       Status
#           component – an Emoji_Component.
#           fully-qualified – a fully-qualified emoji (see ED-18 in UTS #51),
#                           excluding Emoji_Component
#           initially-qualified – an initially-qualified emoji (see ED-18a in UTS #51)
#           non-fully-qualified – a non-fully-qualified emoji (See ED-19 in UTS #51)
# Notes:
#   • This includes the emoji components that need emoji presentation (skin tone and hair)
#     when isolated, but omits the components that need not have an emoji
#     presentation when isolated (Regional_Indicators).
#   • The RGI set is covered by the listed fully-qualified emoji.
#   • The listed initially-qualified and non-fully-qualified cover all cases where an
#     element of the RGI set is missing one or more emoji presentation selectors.
#   • The file is in CLDR order, not codepoint order. This is recommended (but not required!)
#     for keyboard palettes.
#   • The groups and subgroups are illustrative. See the Emoji Order chart for more
#     information.
```

## basic\_emoji.txt

Add a data file **basic\_emoji.txt** that provides a missing piece (see below). The set of emoji code points and emoji presentation sequences intended for general-purpose input.

- The emoji code points are those with property values **Emoji=Yes**, **Emoji\_Component=No**, and **Emoji\_Presentation=Yes**.
- The emoji presentation sequences are those whose base characters have the property values **Emoji=Yes**, **Emoji\_Component=No**, and **Emoji\_Presentation=No**.

## tr51

### 1. change multiple instances of the following

... The specific set of emoji sequences listed in the XXX.txt file [emoji-data] under the category YYY.

⇒

... The specific set of emoji sequences listed in the XXX.txt file [emoji-data] under the type\_field YYY.

Example:

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

⇒

**RGI emoji flag sequence set** — The specific set of emoji sequences listed in the emoji-sequences.txt file [emoji-data] under the type\_field **Emoji\_Flag\_Sequence**.

### 2. Add a definition for **qualified emoji** before **ED-18**.

- a. **ED-17a. qualified emoji** — An emoji character in a string that either (a) has default emoji presentation or (b) starts a modifier sequence or emoji presentation sequence.

### 3. Change the definition of ED-18/19 to be:

- a. **ED-18. fully-qualified emoji** — An emoji (character or sequence) in which every default text presentation character (**ED-7**) is either followed by an emoji modifier or followed by an emoji presentation selector, and there are no other emoji or text presentation selectors in the sequence.
- b. **ED-18a. initially-qualified emoji** — An emoji that is not fully-qualified, but whose first character is qualified.
- c. **ED-19. non-fully-qualified emoji** — An emoji that is neither fully-qualified nor initial qualified.

### 4. I looked into [Mathias Bynens] request for Emoji\_Sequence and here is the current situation regarding definitions.

Well-Formed				RGI	
emoji_character	emoji_combining_sequence	emoji_core_sequence	emoji_sequence	basic emoji set	
emoji_presentation_sequence					
emoji_keycap_sequence				emoji keycap sequence set	
emoji_modifier_sequence				emoji modifier sequence set	
emoji_flag_sequence				RGI emoji flag sequence set	RGI sequence set
emoji_tag_sequence				RGI emoji tag sequence set	
emoji_zwj_sequence				RGI emoji ZWJ sequence set	

The definitions under RGI may have additional constraints put on them. For example, the basic emoji set doesn't include invalid emoji\_presentation\_sequences or Emoji\_Component characters. The conclusion is that the definitions are mostly there, except that we need something that encompasses all the rows under "RGI" below. So that results in #5 and #6

below.

5. Add a data file **basic\_emoji.txt** that provides that basic emoji set, & change ED-20 to reference it. Reason: It should be possible to get the RGI sequence set from the data files, without having to dig into the test files. The missing piece is the basic emoji set. One can either access the test file, or one can construct the basic emoji set programmatically. Easier for implementers and less error-prone to just provide a specific list.

**ED-20. basic emoji set** — The specific set of emoji sequences listed in the **basic\_emoji.txt** file [emoji-data].

- This is the set of emoji code points and emoji presentation sequences intended for general-purpose input.
  - The emoji code points are those with property values **Emoji=Yes**, **Emoji\_Component=No**, and **Emoji\_Presentation=Yes**.
  - The emoji presentation sequences are those whose base characters have the property values **Emoji=Yes**, **Emoji\_Component=No**, and **Emoji\_Presentation=No**.
6. Add a new definition for **RGI set** to include the basic, keycap and modifier sets.

**ED-27. RGI set** — The set of all emoji (characters and sequences) covered by ED-20, ED-21, ED-22, and ED-26.

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