L2/18-202R3

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

From: **ESC**

Date: **2018-06-13 (updated 2018-07-25)**

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, and add Basic_Emoji

```
#   type_field: any of {Emoji_Combining_Sequence, Emoji_Flag_Sequence,
Emoji_Modifier_Sequence}
⇒
#   type_field, one of the following:
#     Basic_Emoji
#     Emoji_Keycap_Sequence
#     Emoji_Flag_Sequence
#     Emoji_Tag_Sequence
#     Emoji_Modifier_Sequence
```

2. In the subheaders, replace space by _ in the type names. Add Basic_Emoji and its data (see below).

```
#     Basic_Emoji: ...
```

// Here insert the Basic_Emoji items. They are:

- // The emoji code points: those with property values **Emoji=Yes**, **Emoji_Presentation=Yes**.
- // The emoji presentation sequences: those whose first characters have the property values **Emoji=Yes** and **Emoji_Presentation=No**.

```
#     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 “minimally-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 “minimally-qualified” when the first character has Emoji_Presentation or starts a valid emoji modifier sequence or emoji presentation sequence, and to “unqualified” otherwise.
    a. Some classes of implementations can support the minimally qualified emoji but not unqualified emoji. The difference is that from the first one or 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
#       minimally-qualified — an minimally-qualified emoji (see ED-18a in UTS #51)
#       unqualified         — a unqualified emoji (See ED-19 in UTS #51)
# Notes:
#   • This includes the individual emoji components that need emoji presentation
#     when isolated (skin tone and hair),
#     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 and components.
#   • The listed minimally-qualified and unqualified 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.
```

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].

   ⇒

   ED-25. 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. Change the definition of emoji to include sequences:
   a. **ED-1. emoji** — A colorful pictograph that can be used inline in text. Internally the representation is either (a) an image or (b) an encoded character or (c) a sequence of encoded characters.
      - For clarity, the term *emoji character* can be used for (b), and *emoji sequence* used for (c).
      - From here on, this specification refers to encoded representations of emoji: that is, (b) or (c), rather than images (a).
3. Add a definition for ***qualified emoji*** before ***ED-18.***
   a. ***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 an default emoji presentation character, but is the first character in an emoji presentation sequence.
4. Change the definition of ED-18/19 to be:
   a. ***ED-18. fully-qualified emoji*** — A qualified emoji character, or an emoji sequence in which each emoji character is qualified.
   b. ***ED-18a. minimally-qualified emoji*** — An emoji sequence in which the first character is qualified but the sequence is not fully qualified.
   c. ***ED-19. unqualified emoji*** — An emoji that is neither fully-qualified nor minimally qualified.
5. Add a type_field **Basic_Emoji** in emoji-sequences.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 must either access the test file, or one construct the basic emoji set programmatically. Easier for implementers and less error-prone to just provide a specific list; it is then parallel to the other sets. (see Background)*

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

   - 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. *Reason: We need terminology to encompass all of the characters that should be supported for general interchange (see Background)*

   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.
7. Section 2.4.1 Emoji and Text Presentation Selectors needs some fixes:
   a. Add at top:
      - 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.

   b. Fix the following (a minimal fix, but see alternative in *Background*)
      - singleton, emoji combining sequence ⇒ emoji combining sequence
8. Add to Acknowledgements , Yifán Wáng and Mathias Bynens

Background

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 above.

NOTE: ED-14b. emoji combining sequence (emoji_combining_sequence) is not particularly productive as a definition. The name is a bit odd, and it only occurs in 1 definition (ED-15. emoji core sequence) and one other place in the text (2.4.1 Emoji and Text Presentation Selectors). The definition could be retired, and the 2 instances replaced by (respectively)

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

singleton,
emoji combining sequence
⇒
emoji character,
emoji presentation sequence,
emoji keycap sequence