L2/13-240

Re: **Reconciling Script and Script_Extensions**
To: **UTC**
From: **Mark**
Date: **Dec 12, 2013**
Live: https://docs.google.com/document/d/1R16tPV-6uPrpe0lEMtKC9I_RuT9I01WMQy92u0vUFB0

A. Anomalies

For 423 characters, the Script_Extensions value is only different from the Script value if the Script value is Common or Inherited. For another 32 characters (see **List 1** below), however, that is not the case. For these characters:

1. the Script value ≠ the Script_Extensions value, and yet
2. the Script value is neither Common nor Inherited

The Unicode Standard gives no principle for when this is done or why. There is a cost to this anomaly in terms of usability and understandability, but by giving users of our data no clue as to why this is done, we don't provide any value for the cost.

We should resolve this by choosing one of the following two policies. Either of these policies could work, but we should choose one.

Document that when the Script_Extensions value ≠ the Script value for a character, the Script value is:

1. *Only* Common or Inherited.
   - **And** change the 32 characters in **List 1** to be Script=Common, and add an invariant test.
   - **Advantage:** Slightly easier for API usage, since implementations need only lookup extra scx info for Common or Inherited characters.
2. *Only* different from Common or Inherited *if* that single script accounts for the vast majority of usage.
   - **And** *consider* changing the script value for certain characters (see **List 2** below for candidates).
   - **Advantage:** For implementations that don't use Script Extensions, in a majority of cases better results would obtain. For example, a string containing U+0660 (٠) ARABIC-INDIC DIGIT ZERO and some Common symbols would be presumed to be Arabic by such an implementation. A more sophisticated implementation could still use the Script_Extensions values to make a more nuanced decision.

B. Policies

The non-explicit Script values have certain well-defined constraints. The Script values do not permit **Common** or **Inherited** as values of Script_Extensions (they don't make sense for it). Moreover, the value **Unknown** is exactly coextensive with certain GC values. For implementers to be able to optimize, it would be useful to have published policies regarding those. So I suggest we request of the officers to add:

6.0.0+ Where not derived from the Script value, the set of Script_Extensions values for a character must only include explicit Script values (that is, they cannot include the values Common, Inherited, or Unknown).

5.0.0+ The set of characters with Script=Unknown is the same as the set of characters with General_Category values Unassigned, Private_Use, or Surrogate

C. ALM

ALM should become sc=Common, scx={Common}. There's no need for it to specify script(s). It was only encoded in the Arabic block to get a default bidi class of AL. A gratuitous differentiation from other bidirectional controls, which are all sc=Common, scx={Common}, adds to the confusion partially created by its name and block. The character is not at all restricted to those scripts in usage, and if used with other scripts, should not trigger shaping engines to switch to a different layout engine which could be very disruptive.

Lists

List 1. Script Value ≠ Common|Inherited

```
# sc=Arabic, scx={Arabic Syriac Thaana}
061C ;     Arabic     # ARABIC LETTER MARK

# sc=Arabic, scx={Arabic Thaana}
FDF2 ;     Arabic     # ARABIC LIGATURE ALLAH ISOLATED FORM

# sc=Bengali, scx={Bengali Syloti_Nagri Chakma}
09E6..09EF ;        Bengali   # BENGALI DIGIT ZERO..BENGALI DIGIT NINE

# sc=Devanagari, scx={Devanagari Kaithi}
0966..096F ;        Devanagari          # DEVANAGARI DIGIT ZERO..DEVANAGARI DIGIT NINE

# sc=Myanmar, scx={Myanmar Tai_Le Chakma}
1040..1049 ;        Myanmar             # MYANMAR DIGIT ZERO..MYANMAR DIGIT NINE
```

List 2. Candidate Policy #2 Script Changes

These were produced by looking at all Script_Extension values, and selecting those that contained exactly 1 script that is in UAX#31 - RECOMMENDED and is not Thaana. (The reason that Thaana is not included is that compared to Arabic, it has only about 0.06% of the literate speaker population that use the script, and only about 0.02% of the characters on the web. It is at the very bottom of the RECOMMENDED list in terms of those two metrics. Thus it is roughly over 1,000 times more likely to be part of Arabic-script text than Thaana.)

These are only candidates. There is no requirement to make these changes in order to adopt Policy #2.

There are two groups. The first group has multiple Script_Extension values per character, and would result in a Script value of Arabic.

```
# old-sc=Common, new-sc=Arabic, scx={Arabic Syriac Mandaic}
0640 ;     Common              # ARABIC TATWEEL

# old-sc=Common, new-sc=Arabic, scx={Arabic Syriac Thaana}
060C ;     Common              # ARABIC COMMA
061B ;     Common              # ARABIC SEMICOLON
061F ;     Common              # ARABIC QUESTION MARK

# old-sc=Common, new-sc=Arabic, scx={Arabic Thaana}
0660..0669 ;        Common              # ARABIC-INDIC DIGIT ZERO..ARABIC-INDIC DIGIT NINE
FDFD ;     Common              # ARABIC LIGATURE BISMILLAH AR-RAHMAN AR-RAHEEM

# old-sc=Inherited, new-sc=Arabic, scx={Arabic Syriac}
064B..0655 ;        Inherited           # ARABIC FATHATAN..ARABIC HAMZA BELOW
0670 ;     Inherited           # ARABIC LETTER SUPERSCRIPT ALEF
```

The second group already only has a single Script_Extension value per character, and the Script value could just become the same. NOTE: for this second group, we already document that this represents cases where we suspect that there are more scripts, but are not yet certain. If we retain the scx values, then that signals this information. Moreover, these are all relatively rare characters, unlike most of the characters in the first group. *So we really don't have to do anything with this group.*

```
# old-sc=Common, new-sc=Devanagari, scx={Devanagari}
1CE1 ;    Common          # VEDIC TONE ATHARVAVEDIC INDEPENDENT SVARITA
1CF2..1CF3 ;    Common          # VEDIC SIGN ARDHAVISARGA..VEDIC SIGN ROTATED
ARDHAVISARGA

# old-sc=Inherited, new-sc=Devanagari, scx={Devanagari}
1CD0..1CD2 ;    Inherited       # VEDIC TONE KARSHANA..VEDIC TONE PRENKHA
1CD4..1CE0 ;    Inherited       # VEDIC SIGN YAJURVEDIC MIDLINE SVARITA..VEDIC TONE
RIGVEDIC KASHMIRI INDEPENDENT SVARITA
1CE2..1CE8 ;    Inherited       # VEDIC SIGN VISARGA SVARITA..VEDIC SIGN VISARGA ANUDATTA
WITH TAIL
1CED ;    Inherited       # VEDIC SIGN TIRYAK
1CF4 ;    Inherited       # VEDIC TONE CANDRA ABOVE

# old-sc=Inherited, new-sc=Greek, scx={Greek}
0342 ;    Inherited       # COMBINING GREEK PERISPOMENI
0345 ;    Inherited       # COMBINING GREEK YPOGEGRAMMENI
1DC0..1DC1 ;    Inherited       # COMBINING DOTTED GRAVE ACCENT..COMBINING DOTTED ACUTE
ACCENT

# old-sc=Inherited, new-sc=Latin, scx={Latin}
0363..036F ;    Inherited       # COMBINING LATIN SMALL LETTER A..COMBINING LATIN SMALL
LETTER X
```