Script Extensions Property

L2/11-406R

Re:	**Script Extensions as a Unicode Property**
To:	**UTC**
From:	**Mark Davis**
Date:	**2011-11-03**

The script extensions just exist as a data file, and not a formal property. That makes them clumsier to cite and use. See, for example, the feedback from Karl Williamson on UTS #18, April 30. We already have *many* multivalued Unicode character properties, in Unihan, so there is no formal difficulty in adding Script_Extensions as a provisional property.

Based on the earlier discussion in committee, here are the proposed changes to make Script_Extensions as a provisional property in Unicode 6.1. (These changes may, of course, need some further refinement by the editorial committee.)

In PropertyAliases.txt, add the line:

SCX ; Script_Extensions

In ScriptExtensions.txt, change the header text to be:

```
# The Script_Extensions property indicates which characters are commonly used
# with a limited number of scripts, but with more than one.
# The property is provisional: values are expected to change over time as more information becomes available.
# For each code point, there is one or more property values.  Each such value is a Script property value.
# For more information, see:
#   UAX #24: http://www.unicode.org/reports/tr24/ and
#   UAX #44: http://www.unicode.org/reports/tr44/
#
#  All code points not explicitly listed for Script_Extensions
#  have as their value the corresponding Script property value.

# @missing: 0000..10FFFF; Script_Extensions; <script>
```

TUS, Chapter 3.5: Add after D32

Some properties may have a set of different values for a given code point. For example, for a given code point the kCantonese property has a set of zero or more strings, such as the list {gun3, hung1, zung1}. For a given code point, the Script_Extensions property has a set of one or more Script property values, such as {Arab, Syrc}.

The order of the elements values in the set may or may not carry meaning. For example, the ordering among the element values for kCantonese and for Script_Extensions carries no particular meaning, while the order among the element values for kMandarin indicates preferred usage between zh-Hans (CN) vs zh-Hant (TW).

In UAX #44, Table 9. Property Table

ScriptExtensions.txt			

Script_Extensions	E	P	Script_Extensions values for use in regular expressions and elsewhere. This property has values that are sets of enumerated properties, whose element values are Script property values. For more information, see Unicode Standard Annex #24, "Unicode Script Property" [UAX24]. For information about multivalued properties, see D32a in Section 3.5 “Properties” of [Unicode].

In UAX #44, Table 7. Property Index by Scope of Use

Add an entry in the table for Script_Extensions, linked to the above, after Script.

In UAX #44, add the following after 5.11.4

5.11.5 Multivalued Properties

Properties such as Script_Extensions or kCantonese have property values consisting of a set of element values. In the data files, these element values are separated by spaces. Validation is performed by first splitting into element values at the spaces, then validating each element value. For example, for Script_Extensions, the element values are Script property values and are validated according to the validation requirements of the Script property.

The Name_Alias property has values that are sets of one or more strings. In the data file for this property, each element value occurs on a separate line.

In UAX #24, 2.8 Multiple Script Values

Change the following text (from/to)

To account for these sorts of tasks, an associated provisional data file called *ScriptExtensions.txt* is provided in the Unicode Character Database [UCD].
...
Although characters with ScriptExtensions data will typically be either **Common** or **Inherited**, there is no guarantee that this is the case.
=>
To account for these sorts of tasks, an additional provisional property, Script_Extensions, is provided in the Unicode Character Database [UCD]. This property is primarily targeted at customary modern use of characters, and does not encompass technical usage such as UPA or math. The values are based on the best available knowledge of usage, which may change over time. The values can be expected to change more frequently than the Unicode character properties, as more information is gleaned about the usage of given characters. Thus implementers should be prepared for enhancements and corrections to the values whenever they upgrade to a new version of the file. No stability guarantees are provided for provisional properties. Although a character with a Script_Extensions value with more than one element value will typically have a Script values of either **Common** or **Inherited**, there is no guarantee that this is the case.

In UAX #24, 4 Data Files, Remove the following text.

~~ScriptExtensions.txt~~

~~The format of this provisional data file is similar to Scripts.txt, except that the second field contains a space-delimited list of short script property values. For example:~~

```
~~# Script_Extensions=Arab Syrc~~

~~0640          ; Arab Syrc # Lm       ARABIC TATWEEL~~
~~064B..0655    ; Arab Syrc # Mn  [11] ARABIC FATHATAN..ARABIC HAMZA BELOW~~
```

~~This data is provided provisionally to supplement the data in Scripts.txt. Because this is supplemental data, not associated with a separate Unicode character property, there is no default value for code points not explicitly mentioned in the data file.~~

In UTS #18, 1.2 Properties (just before " RL1.2 Properties"), add:

Matching Multivalued Properties

Certain properties have values that are sets of other values. In such a case, the value of a property value expression is the set of all code points whose property value *contains* the given value.

For example, using the **Script_Extensions** property in the regular expression \p{scx=Arab}, the value of that expression is the set of all code points whose ScriptExtension value *contains* the script value Arab (=Arabic). Suppose for a given version of Unicode there are the following **Script_Extensions** property values:

- U+064B → {Arab, Syrc}
- U+0600 → {Arab}
- U+0710 → {Syrc}

In such a case:

1. \p{scx=Arab} includes both U+064B and U+0600, but not U+0710
2. \p{scx=Syrc} includes both U+064B and U+0710, but not U+0600.

In UTS #18, 1.2 Properties and 2.7 Full Properties

Change the following text (from/to)

To account for such cases, support of the Script Extensions data as a regular-expression property *Script Extensions* (abbreviated as **scx**) is recommended. Note, however, that the values for such a property are likely be extended over time as new information is gathered on the use of characters with different scripts. For more information, see Multiple Script Values in UAX #24: *Unicode Script Property* [UAX24].

=>

To account for such cases, support of the **Script_Extensions** (scx) property is recommended. Note, however, that the values for such a property are likely be extended over time as new information is gathered on the use of characters with different scripts. For more information, see Multiple Script Values in UAX #24: *Unicode Script Property* [UAX24].