Proposal: Data File for ISO 15924 Script IDs Supported in Unicode

Peter Constable
October 22, 2017

Summary

This document is a proposal to add a data file to UCD to indicate all of the ISO 15924 script IDs that are supported in the given version of Unicode. An alternative is also considered: to have this reflected in ISO 15924 itself.

Background

For a given version of Unicode, the PropertyValueAliases.txt file within the UCD will list script property values for all scripts supported in that Unicode version: for each script, the script property long name and abbreviated name are given, with the abbreviated name being the same as the ISO 15924 script ID. The following snippet from PropertyValueAliases.txt illustrates this:

```
# Script (sc)

sc ; Adlm                           ; Adlam
sc ; Aghb                           ; Caucasian_Albanian
sc ; Ahom                           ; Ahom
sc ; Arab                           ; Arabic
sc ; Armi                           ; Imperial_Aramaic
sc ; Armn                           ; Armenian
...
sc ; Guru                           ; Gurmukhi
sc ; Hang                           ; Hangul
sc ; Hani                           ; Han
sc ; Hano                           ; Hanunoo
...
```

This data covers all scripts supported by that Unicode version, but it doesn't cover all of the ISO 15924 IDs denoting scripts that are supported. In particular, ISO 15924 includes script IDs for script variants, and also script IDs for certain subsets or combinations of scripts, yet these are not listed in PropertyValueAliases.txt. The following are some examples:

- In addition to "Arab" denoting Arabic script, ISO 15924 also includes "Aran" denoting "Arabic (Nastaliq variant)".
- In addition to "Hani" denoting Han ideographs, ISO 15924 also includes "Hans" denoting "Han (Simplified variant), and "Hant" denoting "Han (Traditional variant).
- In addition to "Hang" denoting Hangul script, ISO 15924 also includes "Jamo" denoting "Jamo (alias for Jamo subset of Hangul)".
- In additional "Hang" and "Hani", ISO 15924 also includes "Kore" denoting "Korean (alias for Hangul + Han)".

Some applications require the complete list of ISO 15924 script IDs that are supported by Unicode, including the IDs for script variants and for aliases for script subsets and script combinations.

One large class of applications are those that use BCP 47 language tags to denote the language of certain content, the language of a user preference, etc. BCP 47 language tags include a required language subtag that can be combined with other, optional subtags to qualify the language in certain ways, such as a script ID. All of the valid subtags are defined in the Language Subtag Registry, maintained by IANA. Per the BCP 47 specification, all script IDs from ISO 15924 are included in the Language Subtag Registry, and hence are usable in BCP 47 language tags. If an application is supporting a given version of Unicode, then it may want to know the subset of all ISO 15924 IDs that could occur in language tags for text content supported by that Unicode version.

A specific application scenario pertains to the OpenType font specification. The OpenType format includes a metadata table, designated with table tag, 'meta'. The 'meta' table allows a font developer to add various categories of metadata information pertaining to the font. The supported categories are defined in the OpenType specification and designated by a four-character tag. One of these is "Design languages" (tag 'dlng') which "Indicates languages and/or scripts for the user audiences that the font was primarily designed for." For example, an Arabic font could include a 'dlng' entry to indicate that the font is designed to support Arabic script. OpenType applications could make use of this information in various ways, such as filtering a list of fonts to show only those fonts that are designed for Arabic, or to show previews of the font with the 'dlng' value being used to determine the preview string to be shown.

The 'dlng' entries in an OpenType font use "ScriptLangTag" values, which are adapted from BCP 47 language tags with a syntactic modification: whereas in a BCP 47 language tag the language subtag is required and a script subtag is optional, in OpenType ScriptLangTag values it is the script subtag that is required and other subtags are optional. Some examples:

- A 'dlng' value "Arab" can be used to declare that the font is designed for Arabic script.
- A 'dlng' value "fa-Arab" can be used to declare more specifically that that font is designed for the Persian language written in Arabic script, as opposed to any Arabic-script text.
- A 'dlng' value "Aran" can be used to declare that the font is designed for the Nastaliq style of Arabic script.

Because the OpenType ScriptLangTag values are based on BCP 47 and its use of the Language Subtag Registry, and since all ISO 15924 IDs are included in the Language Subtag Registry, then any ISO 15924 script ID can potentially be used in a 'dlng' entry within an OpenType font. But an application that uses standardized Unicode encoding can only support those ISO 15924 IDs that are supported in Unicode. For example, if an application is displaying font previews using a preview string determined by each font's 'dlng' value, the application can only include preview strings for those script IDs that are supported in the given version of Unicode that the app supports.

The data gap

It is reasonably easy for a developer to create a script to extract a list of supported script IDs from PropertyValueAliases.txt. But that list will not include all script IDs supported by the given Unicode version. A developer must manually review and interpret ISO 15924 to determine what additional script

IDs are supported by that Unicode version. Since, in principle, additional variant/alias script IDs can be added to ISO 15924 at any time, this manual process must be repeated each time the app is updated.

Proposed solution: new UCD data file

The data in question depends on the combination of the script IDs defined in ISO 15924 plus the scripts supported in the given Unicode version. Because of the latter criterion, UCD seems like a logical home for this data. This data could be considered an informative, mutable property of the Unicode version.

A simple UCD data file, SupportedScripts.txt, could have the following format:

```
# SupportedScript-10.0.0.txt
...
# FORMAT
# Each line lists an ISO 15924 script ID supported in this version of Unicode.
# Each line consists of a single field, the ISO 15924 script ID, which is
# followed by a comment, delimited by "#", with the ISO 15924 English name.
# ================================================================================

Adlm # Adlam
Aghb # Caucasian Albanian
Ahom # Ahom, Tai Ahom
Arab # Arabic
Aran # Arabic (Nastaliq variant)
...
```

This would be accompanied by a new section added to UAX #24 to describe this informative data file.

A variation of this proposal would add a derived-age field within the same data file, to indicate the version of Unicode in which a script was first supported:

```
Adlm ; 9.0 # Adlam
Aghb ; 7.0 # Caucasian Albanian
Ahom ; 8.0 # Ahom, Tai Ahom
Arab ; 1.1 # Arabic
Aran ; 1.1 # Arabic (Nastaliq variant)
...
```

This may be a convenient reference,[1] but would not be essential for the anticipated use of this data file except in the case of an application needing to support an earlier version of Unicode that did not yet include this data file.

Alternate proposal: add to ISO 15924 data file

As noted above, the data in question depends on the combination of the script IDs defined in ISO 15924 plus the scripts supported in the given Unicode version. Since, in principle, new script-variant IDs can be

[1] The derived-age information is also available in the "Supported Scripts" page on the Unicode site, http://www.unicode.org/standard/supported.html. That page also includes the ISO 15924 script IDs, including IDs for script variants, embedded as table-cell title properties. Those title properties do not include script subset/alias IDs, however; and parsing an HTML file that was not designed for extraction of data is both less convenient and also generally not to be recommended.

added to ISO 15924 at any time, data provided in a UCD data file could become incomplete, since UCD files for a given Unicode version would not be updated to reflect additions to ISO 15924. For this reason, ISO 15924 may be the best home for this data.

The ISO 15924 RA site provides code tables for script IDs as well as a machine-readable data file. Strictly speaking Unicode-specific information is outside the scope of ISO 15924, but the ISO 15924 content already includes Unicode script property-value aliases as informative data for cross reference to Unicode. The desired additional information could be accommodated by adding a Unicode derived-age field to code tables and the data file.

The current ISO 15924 data file has a line for each script ID with the following format:

```
# Format:
#             Code;N°;English Name;Nom français;PVA;Date
```

The fields are, in order:

- A four-character script ID
- A numeric ID
- English name/description
- French name/description
- Unicode property value alias
- The date when the script ID was added.

The following is an example snippet from the data file:

```
Adlm;166;Adlam;adlam;Adlam;2016-12-05
Afak;439;Afaka;afaka;;2010-12-21
Aghb;239;Caucasian Albanian;aghbanien;Caucasian_Albanian;2014-11-15
Ahom;338;Ahom, Tai Ahom;âhom;Ahom;2015-07-07
Arab;160;Arabic;arabe;Arabic;2004-05-01
Aran;161;Arabic (Nastaliq variant);arabe (variante nastalique);;2014-11-15
...
```

An additional field could be added to give the first Unicode version in which the script was supported:

```
# Format:
#             Code;N°;English Name;Nom français;PVA;Unicode Version;Date
```

In the case of scripts not yet encoded in Unicode, no value would be given. The lines from the current data file shown above would be modified as follows:

```
Adlm;166;Adlam;adlam;Adlam;9.0;2016-12-05
Afak;439;Afaka;afaka;;;2010-12-21
Aghb;239;Caucasian Albanian;aghbanien;Caucasian_Albanian;7.0;2014-11-15
Ahom;338;Ahom, Tai Ahom;âhom;Ahom;8.0;2015-07-07
Arab;160;Arabic;arabe;Arabic;1.1;2004-05-01
Aran;161;Arabic (Nastaliq variant);arabe (variante nastalique);;1.1;2014-11-15
...
```