L2/21-170

UTC #169 properties feedback & recommendations

Markus Scherer / Unicode properties & algorithms group, 2021-sep-28

Properties & algorithms

We are a group of Unicode contributors who take an interest in properties and algorithms.
We look at relevant feedback reports and documents that Unicode receives, do some research, and submit UTC documents with recommendations as input to UTC meetings.

This group started with the UCD file and production tool maintainers, and with Markus Scherer as the chair. Several UTC participants have requested and received invitations to join. We discuss via email, shared documents, and sometimes video meetings.

Participants

The following people have contributed to this document:

Markus Scherer (chair), Mark Davis, Asmus Freytag, Ken Whistler, Ned Holbrook, Josh Hadley, Rick McGowan, Christopher Chapman

Public feedback

Feedback received via the Unicode reporting form, see L2/21-169 “Comments on Public Review Issues (July 20 - Sept 25, 2021)”.

F1: UTS #39 data file default property values

Recommended UTC actions

1. Action item for Mark Davis, Markus Scherer: In IdentifierStatus.txt and IdentifierType.txt, replace comments with appropriate @missing lines, for a future version of Unicode. See L2/21-170 item F1.
```
IdentifierStatus.txt: # @missing: 0000..10FFFF; Restricted
IdentifierType.txt: # @missing: 0000..10FFFF; Not_Character
```

Feedback (verbatim)

Date/Time: Fri Aug 6 16:34:05 CDT 2021
Name: Peter Constable
Report Type: Other Question, Problem, or Feedback
Opt Subject: UTS #39 data file default property values

UTC #168 discussed enhancements to use of @missing lines to indicate default property values.
Coincidentally, I notice that the Identifier_Type and Identifier_Status data files for UTS #39
do not use the @missing convention to indicate default values at all. Rather, each has a prose

statement (not machine readable) describing default values. Moreover, each has two separate statements.

If UTC is going to be enhancing mechanisms for machine-readable default property values, it should consider incorporating the same mechanisms into all data files where relevant.

Background information / discussion

www.unicode.org/reports/tr39/#Data_Files → www.unicode.org/Public/security/latest/

F2: UTS #46 ToASCII does not account for trailing dots

Recommended UTC actions

1. Authorize a proposed update of UTS #46 for Unicode 15.
2. Action item for Markus Scherer, Editorial Committee: For Unicode 15.0, add notes to the effect of the following to UTS #46 section 4.2 ToASCII, Processing step 4.2:
    a. Note: Technically, a complete domain name ends with an empty label for the DNS root (see [STD13] [RFC1034] section 3). This empty label, and the trailing dot, is almost always omitted.
    b. When *VerifyDnsLength* is false, then the empty root label is passed through.
    c. When *VerifyDnsLength* is true, then the empty root label is disallowed. This corresponds to the syntax in [RFC1034] section 3.5 Preferred name syntax which also defines the label length restrictions.
3. Action item for Rick McGowan: Post a Public Review Issue for a proposed update of UTS #46 for Unicode 15.0.

Feedback (verbatim)

Date/Time: Mon Aug 30 03:46:13 CDT 2021
Name: Anne van Kesteren
Report Type: Error Report
Opt Subject: ToASCII does not account for trailing dots

If you invoke https://www.unicode.org/reports/tr46/#ToASCII with VerifyDnsLength set to true it seems you cannot pass a domain such as `example.org.` (note the trailing dot) even though that is a valid domain.

Credit: Gijs Kruitbosch.

Background information / discussion

UTS #46 / 4.2 ToASCII / Processing step 4:
If *VerifyDnsLength* flag is true, then verify DNS length restrictions. This may record an error. For more information, see [STD13] and [STD3].

1. The length of the domain name, excluding the root label and its dot, is from 1 to 253.
2. The length of each label is from 1 to 63.

STD13 RFC1034

- One label is reserved, and that is the null (i.e., zero length) label used for the root.
- Since a complete domain name ends with the root label, this leads to a printed form which ends in a dot. We use this property to distinguish between: [complete vs. incomplete domain names]
- [However, in this RFC, most examples of complete domain names omit the trailing dot.]
- 3.5. Preferred name syntax
    - [This is where the letters/digits/hyphen, length, and other restrictions are specified.]
    - [*The syntax does not allow a trailing dot for the empty root label.*]

Roughly “no one” writes domain names with a trailing dot.

F3: incorrect grammar in UTS #18: Character Classes with Strings

Recommended UTC actions

1. Action for Rick McGowan: Respond to Mickey Rose, pointing to PRI #427. See L2/21-170 item F3 for details.

Feedback (verbatim)

Date/Time: Thu Sep 9 06:59:02 CDT 2021
Name: Mickey Rose
Report Type: Error Report
Opt Subject: incorrect grammar in UTS #18: Character Classes with Strings

The auxiliary grammar presented in 2.2.1 Character Classes with Strings
(https://unicode.org/reports/tr18/#Character_Ranges_with_Strings) doesn't
generate the examples given further.

Here are some of the examples (within character class):
  [a-z\q{x\u{323}}]
  [a-z ñ \q{ch} \q{ll} \q{rr}]

And here is the grammar:
  ITEM := "\q{" (CODE_POINT (SP CODE_POINT)*)? "}"
  SP   := \u{20}

The grammar suggests that a single SP is required between individual
CODE_POINTs. Which if true would be confusing, for example [\q{c h}].

Besides, this ITEM production is supposed to be embedded in CHARACTER_CLASS
grammar (https://unicode.org/reports/tr18/#character_ranges) which
already allows and ignores whitespace:

  >> Whitespace is allowed between any elements, but to simplify the presentation the many occurrences of sequences of spaces (" "*) are omitted.

So I believe what was actually intended is this:

```
  ITEM := "\q{" CODE_POINT2* "}"
```

(with whitespace allowed by virtue of being embedded within CHARACTER_CLASS grammar)

In this scenario [\q{aa ch}] is equivalent to [\q{aach}].

Alternatively, if SP is intended to separate whole strings inside \q{}, then you need to allow multiple CODE_POINTs without SP between them:

```
  ITEM := "\q{" CODE_POINT2* (SP CODE_POINT2+)* "}"
```

In this scenario [\q{aa ch}] is equivalent to [\q{aa}\q{ch}]. But then it would be very confusing that only \u{20} would act as separator, while other whitespace like \u{09} wouldn't.

In either case, some examples with spaces inside \q{...} should be given for clarification.

Background information / discussion

The pending draft of UTS #18 overhauls the syntax grammar and no longer includes a space between string literal characters.

- www.unicode.org/review/pri427/ “Proposed Update UTS #18, Unicode Regular Expressions”
- www.unicode.org/reports/tr18/tr18-22.html#Character_Ranges_with_Strings
  - '\q{' LITERAL* ('|' LITERAL*)*'}'

Unicode Tools issues

The Unicode Tools maintainers use a GitHub repo with an issue tracker:
https://github.com/unicode-org/unicodetools/issues

Occasionally, we receive public feedback there that could be relevant to the UTC. We will be assessing these soon. For example:

- unicodetools/issues/39 “Bug in Word Segmentation demo”
- unicodetools/issues/107 “Testcases for invalid punycode with surrogates for IdnaTestV2”

Documents

D1: Proposal to Adjust Identifier Types for Certain Modifier Letters

L2/21-136 from Asmus Freytag

Recommended UTC actions

No UTC action needed at this time.

The properties & algorithms group (including Asmus) reviewed and discussed L2/21-136. Asmus will take the feedback into account and revise his document.

Summary

Several scripts have modifier letters or spacing tone marks that are indistinguishable from Punctuation, including punctuation found directly in fully qualified domain names or punctuation that may be used to terminate or surround domain names.

While they are PVALID in IDNA2008, these should not have Identifier_Type “Recommended”. Instead, it would be more appropriate to move them to “Inclusion”. They may be appropriate (or harmless) in some identifier contexts, but are definitely problematic in domain names.
[...]
As a result, despite being PVALID, these letters have been excluded from the DNS Root Zone for security reasons.

Background information / discussion

https://www.unicode.org/reports/tr39/#Identifier_Status_and_Type

D2: Allocating Arabic Extended-C in SMP and Arabic code point changes

L2/21-181 from Roozbeh Pournader

Recommended UTC actions

If the SAH recommends adopting this proposal, then:

1. Action item for Mark Davis, Properties & Algorithms Group: In DerivedBidiClass.txt, change the default Bidi_Class value for 10EC0..10EFF from R to AL, for Unicode 15.0.

Summary

I propose allocating 10EC0..10EFF to Arabic Extended-C, ...

Background information / discussion

Mostly a topic for the script ad hoc and the roadmap team, but adding here, too because presumably we will want to change the default Bidi_Class values for the new block from R to AL.

https://unicode.org/roadmaps/smp/

```
00010800-00010FFF Alphabetic and syllabic RTL scripts
```

https://www.unicode.org/Public/UCD/latest/ucd/extracted/DerivedBidiClass.txt

```
# The unassigned code points that default to R are in the ranges:
#     [\u0590-\u05FF \u07C0-\u085F \uFB1D-\uFB4F
#      \U00010800-\U00010CFF \U00010D40-\U00010F2F \U00010F70-\U00010FFF
#      \U0001E800-\U0001EC6F \U0001ECC0-\U0001ECFF \U0001ED50-\U0001EDFF
\U0001EF00-\U0001EFFF]
```