

Unicode identifier styles

To: UTC
 From: Robin Leroy, Source code ad hoc working group
 Date: 2022-10-20

Many style guides require some case and separation convention for some identifiers (*e.g.*, CamelCase, snake_case, etc.), which is often enforced by automated tooling. It can be unclear how to generalize these styles and their enforcement beyond Basic Latin, and how to do so without preventing the effective use of unicameral scripts.


This document defines generalizations of common identifier styles.


Note: Document [L2/22-229](#) proposes incorporating the definitions from this document into a new Unicode Technical Standard. The purpose of this document is to serve as a more detailed rationale for its technicalities.

This file uses the regular expression syntax defined in [UTS #18 Unicode Regular Expressions, version 23](#).

Definition

An implementation claiming to implement Unicode identifier styles shall emit some of the diagnostics defined below.

1. BactrianCamel: 
 A diagnostic shall be emitted if an identifier matches the following regular expression:

$$\text{^\{Ll\} | \{LC\}[\{Mn\}\{Me\}]^* \{Pc\} \{LC\}}$$
2. dromedaryCamel: 
 A diagnostic shall be emitted if an identifier matches the following regular expression:

$$\text{^\{Lu\}\{Lt\} | \{LC\}[\{Mn\}\{Me\}]^* \{Pc\} \{LC\}}$$
3. small_snake:
 A diagnostic shall be emitted if an identifier matches the following regular expression:

$$\text{[\{Lu\}\{Lt\}]}$$
4. Title_Snake:
 A diagnostic shall be emitted if an identifier matches the following regular expression:

$$\text{(^ | \{Pc\}) \{Ll\}}$$
5. CAPITAL_SNAKE: A diagnostic shall be emitted if an identifier, once normalized under Normalization Form C, matches the following regular expression:

$$\text{[\{Ll\} \{Lt\}]}$$

Alternatively, it shall declare a profile, and define the situations in which the aforementioned diagnostics are suppressed and the additional situations in which they are emitted.

Example: An implementation could implement the BactrianCamel diagnosis with a profile that additionally prohibits $\text{(\{Lu\}[\{Mn\}\{Me\}]^*)\{4\}}$ (four uppercase letters in a row).

Example: An implementation could implement the `Title_Snake` diagnostic with a profile that allows lowercase after a Connector Punctuation (allowing `Proud_snake_case`).

Example: An implementation which meets requirement UAX31-R1 with a profile adding the hyphen-minus (-) to `Continue` could implement the various diagnostics with a profile that replaces `\p{Pc}` in the above regular expressions by `[\p{Pc}\p{Pd}]`, treating the hyphen-minus like the low line (allowing “kebab-case”).

Rationale

The `BactrianCamel` and `dromedaryCamel` diagnostics are emitted if Connector Punctuation separates bicameral scripts. This generalizes the prohibition of low line (`_`), but allows low line and similar characters adjacent to a unicameral script, where case cannot be used to visually separate words.

Note that Connector Punctuation is allowed between a bicameral and a unicameral script; this is necessary as letters from some unicameral scripts are confusable with letters from bicameral scripts. In the absence of separation, the result would be visually confusing. See [L2/22-231](#).

The `BactrianCamel` and `dromedaryCamel` diagnostics are also emitted when the initial letter is in a bicameral script and in the incorrect case.

The various flavours of snake diagnostics check for letters with incorrect case in bicameral scripts; for `Title_Snake` this check is performed next to Connector Punctuation and at the beginning, generalizing the prohibition of `(^|_)[a-z]`.

The normalization step in `CAPITAL_SNAKE` is needed to turn some innocent-looking Greek `\p{Lu}\p{Mn}` into `\p{Lt}`, which should not be present in upper case. See the Core Specification, [Section 5.18 “Case Mappings”, Subsection “Complications for Case Mapping”](#), Paragraph “Greek *iota subscript*”, and [Section 7.2 “Greek”](#), Paragraph “Iota”.

Note that Other Punctuation is treated differently from Connector Punctuation: indeed, the characters in `[[:Po:]]` & `[[:XID_Continue:]]` are in `[[:XID_Continue:]]` not as word separators, but because they are needed as part of words in Catalan; see [UAX #31 Unicode Identifier and Pattern Syntax, Section 2.4 “Specific Character Adjustments”](#).

Acknowledgements

We thank Manish Goregaokar for bringing this problem to our attention. Richard Smith suggested the example of `Proud_snake_case`.