C++ Identifier Security using Unicode Standard Annex 39 v2

Document #: D2538R1
Date: 2022-08-30
Project: Programming Language C++
Audience: SG-16 EWG CWG
Reply-to: Reini Urban <reini.urban@gmail.com>

1 Abstract

Adopt Unicode Annex 39 “Unicode Security Mechanisms” as part of C++26.

Unicode identifiers bury a small risk for homoglyph attacks getting into source code. Compilers are not confused, but reviewers and programmers are as it’s impossible to detect such attacks without special tooling, preferably the compiler as the source of truth. And essentially confusable identifiers are not identifiable anymore.

2 Changes

From R0:

- Add internal links.
- Rename C23 to C26, it’s too late for C++23.
- Disallow non-confusable `Technical` U+1C0..U+1C3
- Fix a lot of not Allowed ID_Start ranges. safec26_start_list from 355 ranges, 115 singles, 99350 codepoints to 243 ranges, 93 singles, 95986 codepoints
- Added U+3C3 GREEK SMALL LETTER SIGMA and U+3BD GREEK SMALL LETTER NU to the Greek confusable exceptions in 19.1.
- Added Appendix G - Medial.
- Change U+B7 Catalan MIDDLE DOT from Inclusion to Uncommon Use.
- Disallow Arabic Presentation Forms-A: U+FB50–U+FDFF and Arabic Presentation Forms-B: U+FE70–U+FEFF
- Added wording feedback from the first SSRG discussion, and restructure the paragraphs a bit to be less technical, and make it more readable to non-Unicode experts.
- Added discussions of the gcc and clang-tidy -Whomoglyph approaches via confusables.
- Extend 8.3 Combining marks script run detection for spoofing. Added Appendix H with the list of affected letters.

3 Introduction

In response to P1949R7, and in parallel to n2932 for C.

Adopt Unicode Annex 39 “Unicode Security Mechanisms” as part of C++26.

- Comply to a variant of the TR39#5.2 Mixed-Scripts Moderately Restrictive profile, but allow some Greek letters without its confusables with Latin,
- Disallow all Limited Use TR31#Table_7 and Excluded scripts TR31#Table_4,
- Only allow TR39#Table 1 Recommended, Inclusion, Technical Identifier Type properties,
- Reject illegal combining mark sequences (Sk, Cf, Mn, Me) with mixed-scripts (SCX) TR39#5.4, if they are not already addressed by the NFC requirement from C++23, as of P1949.

Optionally:

- Implementations may allow an optional `#pragma unicode <LongScript>` that Excluded scripts can be added to the allowed set of scripts per source file.

Open points:

- How to name the `#pragma unicode` extension.
- Which context to use in C++: before-cpp, private (lexically scoped) or after-cpp.
- Go against TR39 recommendations and don’t disallow Excluded Scripts. This would require different initial XID tables, would enlarge the attack surface implementations and font designers have no experience with yet, but would simplify the implementations.

In addition adopt this proposal as a Defect Report against C++23 and earlier. The author provides the libu8ident library (Apache 2 licensed) and its generated tables to all implementors.

TR39 checks could be implemented as warnings, TR31 violations not. But note that even TR31 has bugs still, to be hopefully fixed in the next Unicode version.

4 Motivation

One driving factor for addressing this now is that GCC has fixed their long standing bug 67224 “UTF-8 support for identifier names in GCC”. Clang has always supported too many C++11 code points in

source code. MSVC in its usual configuration defaults to code page 1252, but can be told to accept UTF-8 source. With GCC now allowing it, the barrier to use of Unicode characters outside the basic source character set has dropped considerably. Use of characters via universal character names was always possible, but never widely used. Examples found in the wild of use of UCNs in identifiers come from compiler and related tool test suites, but it's trivial to come up with such spoofing attacks. There is no report yet from misuse in C ABI's from linkers and binutils.

Restricting the profile of characters is much easier if no one is depending on them yet.

Some actual C++11 user-code representing the epsilon transport equation:

```
    solve(div(U * ε) - div(νt * grad(ε)) / σε + C2 * ω * Sp(ε)
      == C1 * ω * G, ε, α);
```

From visual inspection you can not decide if the greek identifiers here are actual greek, latin or cyrillic.

Even worse is this, which writes an identifier in latin and then in cyrillic. These can be in different source files. The string also btw, but strings don't need to be identifiable.

```
int CHECK (const char *arg) {
  return strcmp(arg, "check") == 0;
}
int CHECK (const char *arg) {
  return strcmp(arg, "check") == 0;
}
```

Adopting TR39 would fix all of the known security problems with C++/C identifiers. With C++ it is more severe as declarations are easily confusable with initializations. But not as severe as in unstrict dynamic languages.

The recent https://trojansource.codes effort caused gcc to emit a new bidi warning, and github to implement similar warnings. Note that secure identifiers don't help against bidi overrides in strings or comments, these issues are orthogonal. The Unicode consortium implemented a unicode spoofing taskforce https://www.unicode.org/L2/L2022/22007-avoiding-spoof.pdf. Their ICU library would need an identifier check API at least.

There used to be no linter, but there is now one: My **u8idlint** from https://github.com/rurban/libu8ident, which can be used to check for ALLOWED,SAFEC26,C23,ID,XID,C11 or ALLUTF8 TR31 profiles, for various TR39 mixed script profile violations, confusables, invalid

combining marks and TR15 normalization problems. Go also came up with a unicode spoofing linter lately: https://github.com/NebulousLabs/glyphcheck

So far only Rust, cperl and Java follow a TR39 Unicode security guideline for identifiers, zig and J refused to support non-ASCII identifiers. Everbody else is vulnerable to potential security attacks and does allow non-identifiable identifiers. They should rename their identifiers to "symbols".

Links:

- https://cwe.mitre.org/data/definitions/1007.html (*The gcc CWE-1007 -Whomoglyph warning is linking to it.*)
- https://websec.github.io/unicode-security-guide/visual-spoofing/
- http://www.unicode.org/reports/tr31/, http://www.unicode.org/reports/tr36/ and http://www.unicode.org/reports/tr39
- https://twitter.com/zygoloid/status/1187150150835195905, https://github.com/golang/go/issues/20209, https://twitter.com/jupenur/status/1244286243518713857
- https://certitude.consulting/blog/en/invisible-backdoor/
- https://github.com/rurban/libu8ident/tree/master/texts/ with *-sec*.c

5 Design

First we are discussing two different approaches found in praxis:

1. TR39#4 Confusable_Detection,

   vs.

2. TR39#5.1 Mixed_Script_Detection.

TR39 lists some options how to implement a security mechanism for identifiers. In praxis there are three successful usages of the mixed-script approach in java, cperl and rust, as specified here. No other language implemented TR39 since and uses it. Python tried the confusables approach optionally, and gcc and clang-tidy is trying it out now. See 12 Implementations and Strategies.

GCC has a new **-Whomoglyph** warning patch at PR 103027 (see also my github for an updated version). It implements the "skeleton" algorithm from TR39#4 so that every new identifier is mapped to a "skeleton", and if the skeleton is already in use by a different identifier, issue a `-Whomoglyph` diagnostic. It uses the `security/confusables.txt` table to determine which characters are confusable. It uses a NFD lookup and three hash lookups per identifier. NFD is relatively cheap

compared to NFC, mandatory since C23 and C++23, but much more expensive than the mixed script approach which uses only a single range-lookup in most cases.

clang-tidy https://reviews.llvm.org/D112916 was fairly unsucessful so far, and used the similar confusables approach.

Pros:

- confusables need not to care about scripts, in which language, the document is written. The first variant of an identifier is the accepted one, and the subsequent ones with expanded confusable matches are invalid. First come, first serves.
- Forbidding rarely used scripts can be seen as politically loaded.

Cons:

- `confusables.txt` has much more bugs and oddities than TR31, the should-be stable list of XID identifiers. So far I've found 3 bugs in TR31 for Unicode v14. In confusables.txt ASCII has 12 exceptions to be ignored, Greek needs 12 exceptions out of 260, and I didn't check any other scripts.
- Following TR39#5 Mixed Scripts would be easier to understand, as it is defined by simple rules, and not a hand-curated, buggy and unstable table. Even the first violation is an error, thus no surprises when code moves around.
- Implementing the `confusable.txt` checks only (as proposed in the two gcc and clang tickets) can be slow (as experienced in clang-tidy), and led to a huge number of warnings (over 100.000). The GCC implementation (see my github) is fast, but needs recursive dynamic hash lookups. Whilst implementing the mixed-scripts strategy as laid out here is extremely fast and led to no warnings so far in published code.
- Mixed scripts are already successfully used in praxis for several years, without any complaints.

There were a few more design decisions made, over TR39 recommendations:

- Allow some Greek letters mixed with Latin, that are not confusable with Latin letters. The rationale is that the by far mostly used script is Greek, because of its mathematical symbols and physical constants actively used by C++ physicists. This is in fact the only usage of unicode identifiers in the wild. There is some overlap with Latin symbols, and in all cases where a Greek letter is confusable with a Latin one, the Latin one must be used. See Appendix F.

TR39 recommends to forbid some sets of Limited Use scripts for identifiers, and recommends to only optionally allow some other Excluded scripts. See 7.2 Script restrictions.

Speed/Size summary:

The mixed-script approach was also sucessfully used in a dynamic language, with much stricter performance restrictions than offline compilers or linters. There was no noticable compile-time performance degradation, as unicode identifiers are extremely rare, and the NFC check is by far slower than the mixed script and illegal combining mark checks. NFC needs 183K alone, the mixed script check with the TR31, medial and mark tables 131K in my unoptimized, generic implementation. C++26 can do a bit better, but this is good enough.

6 Summary

P1949 correctly detected that Unicode identifiers are still not identifiable, and are prone to bidi- and homoglyph attacks. But it stated that implementing TR39 would be too hard. Having properly implemented the Unicode Security Guidelines for identifers for several years, plus pushed Rust to do so also, proves the contrary. TR39 would catch all known homoglyph and bidi identifier attacks.

Further restriction of the TR31 profile as recommended by TR39 to only recommended scripts leads to smaller sets for identifiers, and implementation of a proper TR39 mixed script profile and identifier types fixes most of the known unicode security problems with identifiers. The only remaining problems are bidi overrides in strings or comments evading syntax, which cannot be handled with identifier restrictions, but tokenizer or preprocessor warnings, as recently added to gcc and clang. `#include filename` restrictions should be done also, but that is also out of the scope of this document, as the existing filesystems care much less about unicode security for identifiers than programming languages. Spoofing attacks on filenames are not yet seen in the wild, but will appear sooner or later, same as they appeared in browsers and email. Also names in C/C++ object files: linkers, .def files, ffi's.

Implementing TR39 mixed script detection per document (C++ Header and Source file) forbids insecure mixes of Greek and Cyrillic, dangerous Arabic RTL bidi attacks and confusables. You can still write in your language, but then only in commonly written languages, and not mixed with others. Identifiers are still identifiable.

The question remains if TR39 security violations should be ill-formed

(throw an compilation error or warning), or not. Since we do have the `-std=c++26` option, and the issues are security relevant, ill-formeded seems to be best. Implementations might choose to go for compiler warnings or linters or just toolchain implementations, i.e. editors and reviewer tools. The practical security problems are not severe and are easy to fix, as we had none in the years clang allowed insecure unicode, and there were no major known problems on the easier to attack dynamic languages. But gcc just added it now with gcc-10, so the impact might just come later. TR39 is considered stable and not a moving target. There were no impactful changes in the last 10 years.

7 What will this proposal change

7.1 The set of TR31 XID characters will become much smaller

Restricting the **Identifier Type** plus the Recommended Scripts, will shrink the original XID set from 971267 codepoints to 99350 codepoints. The ranges expand from 36 to 426. (when split by scripts already, 25 splits happen). Additionally the Halfwidth and Fullwidth Forms, U+FF00..U+FFEF, the Arabic Presentation Forms-A: U+FB50–U+FDFF and Arabic Presentation Forms-B: U+FE70–U+FEFF are now forbidden.

`ID_Start` consists of Lu + Ll + Lt + Lm + Lo + Nl, `+Other_ID_Start`, `-Pattern_Syntax`, `-Pattern_White_Space`

131899 codepoints

`ID_Continue` consists of `ID_Start`, + Mn + Mc + Nd + Pc, `+Other_ID_Continue`, `-Pattern_Syntax`, `-Pattern_White_Space`.

135072 codepoints (= ID_Start + 3173)

`XID_Start` and `XID_Continue` ensure that `isIdentifer(string)` then `isIdentifier(NFKx(string))` (*removing the NFKC quirks*)

`XID_Start`: 131876 codepoints, `XID_Continue`: 135053 codepoints (= `XID_Start` + 3173)

See Appendix A - C26XID_Start and Appendix B - C26XID_Continue.

For the medial positions see Section 19 Appendix G - Medial. They are not allowed as first nor as last character in a word, but this set of identifiers contain none, as we disallow the legacy Arabic Presentation forms.

7.2 Script restrictions

P1949R7 for C++23 previously stated: *"This paper also does not propose excluding any scripts categorically, regardless of their status as historic or obsolete. Characters from Anatolian Hieroglyphs would be available for use, to the extent that anyone wishes to do so."*

TR31#Table 4 states: *"Some scripts are not in customary modern use, and thus implementations may want to exclude them from identifiers. These include historic and obsolete scripts, scripts used mostly liturgically, and regional scripts used only in very small communities or with very limited current usage. Some scripts also have unresolved architectural issues that make them currently unsuitable for identifiers. The scripts in Table 4, Excluded Scripts are recommended for exclusion from identifiers."*

These Excluded Scripts are initially disallowed TR31#Table_4 but can be optionally be allowed via a new `#pragma unicode ExcludedScript`:

`Ahom Anatolian_Hieroglyphs Avestan Bassa_Vah Bhaiksuki Brahmi Braille Buginese Buhid Carian Caucasian_Albanian Chorasmian Coptic Cuneiform Cypriot Cypro_Minoan Deseret Dives_Akuru Dogra Duployan Egyptian_Hieroglyphs Elbasan Elymaic Glagolitic Gothic Grantha Gunjala_Gondi Hanunoo Hatran Imperial_Aramaic Inscriptional_Pahlavi Inscriptional_Parthian Kaithi Kharoshthi Khitan_Small_Script Khojki Khudawadi Linear_A Linear_B Lycian Lydian Mahajani Makasar Manichaean Marchen Masaram_Gondi Medefaidrin Mende_Kikakui Meroitic_Cursive Meroitic_Hieroglyphs Modi Mongolian Mro Multani Nabataean Nandinagari Nushu Ogham Old_Hungarian Old_Italic Old_North_Arabian Old_Permic Old_Persian Old_Sogdian Old_South_Arabian Old_Turkic Old_Uyghur Osmanya Pahawh_Hmong Palmyrene Pau_Cin_Hau Phags_Pa Phoenician Psalter_Pahlavi Rejang Runic Samaritan Sharada Shavian Siddham SignWriting Sogdian Sora_Sompeng Soyombo Tagalog Tagbanwa Takri Tangsa Tangut Tirhuta Toto Ugaritic Vithkuqi Warang_Citi Yezidi Zanabazar_Square`

"Modern scripts that are in more limited use are listed in Table 7, Limited Use Scripts. To avoid security issues, some implementations may wish to disallow the limited-use scripts in identifiers. For more information on usage, see the Unicode Locale project [CLDR]." These Limited Use Scripts are now disallowed TR31#Table_7:

`Adlam Balinese Bamum Batak Canadian_Aboriginal Chakma Cham Cherokee Hanifi_Rohingya Javanese Kayah_Li Lepcha Limbu Lisu Mandaic Meetei_Mayek Miao New_Tai_Lue Newa Nko Nyiakeng_Puachue_Hmong Ol_Chiki Osage Saurashtra Sundanese Syloti_Nagri Syriac Tai_Le Tai_Tham`

`Tai_Viet Tifinagh Vai Wancho Yi Unknown`

This recommendation follows TR39, to recommended scripts only, Excluded and Limited Use not. For some years until Unicode 10 there was a "Aspirational Use Scripts" table, which included a subset of the optional Limited Use scripts to be allowed in identifiers. But "*this has not proven to be productive for the derivation of identifier-related classes used in security profiles*".

Thus these scripts will stay allowed:

`Common Inherited Latin Arabic Armenian Bengali Bopomofo Cyrillic Devanagari Ethiopic Georgian Greek Gujarati Gurmukhi Hangul Han Hebrew Hiragana Katakana Kannada Khmer Lao Malayalam Myanmar Oriya Sinhala Tamil Telugu Thaana Thai Tibetan`

Stability:

Historically the most changes in latest unicode versions have been with adding to Emojis and Limited Use scripts. Thus the expected set of valid identifiers looks stable, when all the current TR31 bugs will be fixed. I have no idea about the TR39 confusables.txt bugs, as there is no categorization yet.

The script property and its name are defined in TR24. We use the long Unicode Script property value, not the abbrevated 4-letter short name, which maps somehow to the 4-letter ISO 15924 Codes.

7.3 Documents with identifiers in many multiple scripts/languages will become illegal

C++26 (and C26) will follow the TR39 Security Profile 4 **Moderately Restrictive**, with an exception for Greek.

- All identifiers in a document qualify as Single Script, or
- All identifiers in a document are covered by any of the following sets of scripts, according to the definition in Mixed Scripts:
  - Latin + Han + Hiragana + Katakana (Japanese)
  - Latin + Han + Bopomofo (Chinese)
  - Latin + Han + Hangul (Korean), or
- All identifiers in a document are covered by Latin and any one other Recommended script, except Cyrillic.
- Allow some Greek letters mixed with Latin, that are not confusable with Latin letters.

See Section 10 TR39 Mixed Scripts.

7.4 Mixed-script runs with combining marks will become illegal

C++26 (and C26) will check for unlikely sequences of **combining marks**, and reject some. Combining Marks have no script property per se, but a variable list of allowed SCX scripts, which need to be checked against the base character. Also 4 Japanese KATAKANA-HIRAGANA PROLONGED SOUND MARK modifier letters.

This section is technically security-relevant, as over-long runs of combining marks may lead to overflow in sequences.

See 8.2 “SCX Extensions” and 8.3 “Combining marks script run detection for spoofing” below.

8 TR24 Scripts, the SC and SCX properties

8.1 SC

C++ only needs to map unicode characters to a script property via a single byte. There are currently 161 scripts assigned, 32 of them are in common use as identifiers, hence called **Recommended** scripts. The rest is split up into 127-31 **Excluded** scripts, which are not in common use, and 161-127 **Limited_Use** scripts, which are not to be used in identifiers at all.

Regarding the discriminatory aspect of Excluded Scripts from TR31#Table_4. *“Some scripts are not in customary modern use, and thus implementations may want to exclude them from identifiers. These include historic and obsolete scripts, scripts used mostly liturgically, and regional scripts used only in very small communities or with very limited current usage. Some scripts also have unresolved architectural issues that make them currently unsuitable for identifiers. The scripts in Table 4, Excluded Scripts are recommended for exclusion from identifiers.”* Nevertheless an implementation might choose to allow some optionally via a new `#pragma unicode Script`.

Regarding Limited Use scripts: TR31#2.4: *“Modern scripts that are in more limited use are listed in Table 7, Limited Use Scripts. To avoid security issues, some implementations may wish to disallow the limited-use scripts in identifiers. For more information on usage, see the Unicode Locale project CLDR.”*

Regarding stability: New scripts are added on a yearly basis, but nothing was added to the stable set of recommended scripts. For a while there was a list of **Aspirational** scripts to be added eventually, but this list was abandoned with Unicode 10.0. Probably also

because nobody but Java, cperl and Rust implemented its identifier profile by scripts, rather went with insecure identifiers.

For error messages and an optional pragma to allow certain Exluded scripts, we use the long **Script property value**. Do not use the term "script name", as this is ambigious and misused. The Script Property Value is the titlecased name of the script from the UCD, with spaces replaced by underscores. They are defined in the yearly updated Scripts.

8.2 SCX Extensions

Not all characters are uniquely used in a single script only. Many are used in a variable numbers of scripts. These are assigned to the Common or Inherited script, and are exactly specified in the ScriptExtensions, aka SCX. The SCX property is a list of possible scripts per character. This list is using the short 4-letter script property, which needs to be resolved via the PropValue to its long script property value. (E.g. Syrc to Syriac)

```
# Script_Extensions=Arab Syrc

064B..0655 ; Arab Syrc # Mn  [11] ARABIC FATHATAN..ARABIC HAMZA BELOW

# Script_Extensions=Adlm Arab Mand Mani Ougr Phlp Rohg Sogd Syrc

0640     ; Adlm Arab Mand Mani Ougr Phlp Rohg Sogd Syrc # Lm  ARABIC TATWEEL
```

Some of the SCX scripts contain only a single script. These could be directly added to the list of SC scripts for the purpose of identifier security checks, but I advise against, for easier Combining Marks checks against the base character script. See below 8.3.

E.g.

```
3006       ; Hani # Lo       IDEOGRAPHIC CLOSING MARK
```

U+3006 with the Common script property is assigned to the Hani -> Han script.

Multiple SCX list entries can be resolved when the previous scripts in the identifier context are already resolved to one or the other possibility. Thus for SCX=(Arab Syrc) we need to check if Arabic or Syriac was already seen. If not, the new character with that SCX is illegal, violating our Mixed Script profile.

8.3 Combining marks script run detection for spoofing

Check for unlikely sequences of **combining marks**:

- Forbid starting combining marks.
- Forbid sequences of the same nonspacing mark.
- Forbid sequences of more than 4 nonspacing marks (gc=Mn or gc=Me).
- Optionally forbid sequences of base character + nonspacing mark that look the same as or confusingly similar to the base character alone (because the nonspacing mark overlays a portion of the base character). An example is U+0069 LOWERCASE LETTER I + U+0307 COMBINING DOT ABOVE.
- Optionally forbid non-spacing marks with base chars already including the non-spacing mark, like Ä with DIAERESIS.

Since we disallow already most combining marks (at least the Latin ones) with the requirement of NFC in P1949R7, this set of cases is quite small.

Special-cases:

Using the Script property alone will not detect that the U+30FC (ー) KATAKANA-HIRAGANA PROLONGED SOUND MARK (Script=Common, SCX=Hira Kana, gc=Lm) should not be mixed with Latin. See TR39#5.4 and TR46. We only have to check only 4 such explicitly japanese-only PROLONGED SOUND MARKs, all other Lm modifiers may mix with all SCX.

The list of allowed combining mark characters (with Common or Inherited scripts) in the C++26 TR31 profile is: Lm `Modifier_Letter`, Mc `Spacing_Mark`, Mn `Nonspacing_Mark`, Me `Enclosing_Mark`. Sk and Cf are not part of XIDs.

67 matches for “XID_Continue # Lm” in buffer: DerivedCoreProperties.txt See Appendix C for all.

```
02B0..02C1    ; XID_Continue # Lm  [18] MODIFIER LETTER SMALL H..
                                        MODIFIER LETTER REVERSED GLOTTAL STOP
02C6..02D1    ; XID_Continue # Lm  [12] MODIFIER LETTER CIRCUMFLEX ACCENT..
                                        MODIFIER LETTER HALF TRIANGULAR
...
```

513 matches for “XID_Continue # M” in buffer: DerivedCoreProperties.txt See Appendix D for all.

```
0300..036F    ; XID_Continue # Mn [112] COMBINING GRAVE ACCENT..
                                        COMBINING LATIN SMALL LETTER X
0483..0487    ; XID_Continue # Mn   [5] COMBINING CYRILLIC TITLO..
```

```
                                                COMBINING CYRILLIC POKRYTIE
...
```

From these 67 Lm plus 513 M[cn] ranges filtering out the non-C++26 XID candidates, only #8 Identifier_Type = Recommended, Inclusion, non-confusable Technical, plus only #4.2 Recommended Scripts, plus only codepoints with multiple SCX entries, plus only codepoints which don't decompose to NFC, leads only to the Lm characters, which can mix with all scripts. Not a single Mn or Mc codepoints is left.

So some of the Common `XID_Continue` marks therefore cannot be detected with the SCX logic. But all of them do not combine with Latin and are already filtered by the C++26 Mixed Script profile.

Most Lm Modifier Letters (besides the 4 Japanese PROLONGED SOUND MARKs) are freestanding base characters, which can be combined with any other letter.

See TR31#2.1 Combining_Marks and TR31#2.2 Modifier_Letters

Most of the Combining Marks are caught by the NFC requirement from C++23 (P1949r7), but some optional checks for gc=Mn against base chars still might be added:

Forbid U+0069 LOWERCASE LETTER I + U+0307 COMBINING DOT ABOVE as a special-case.

Forbid non-spacing marks with base chars already including the non-spacing mark. There are currently 82 non-spacing marks, e.g. GRAVE, DOT ABOVE, … most of them in the U+0300-U+0341 range, but some also in U+20D0-U+20E1 and U+3099-U+309A. From these there exist 24 letters which already include those non-spacing marks. See Appendix H

See also TR24#5.1 Handling Characters with the Common Script Property and TR24#5.2 Handling Combining Marks.

9 TR39 Identifier Type

TR39 recommends to disable some characters from recommended scripts: *"Some characters used with recommended scripts may still be problematic for identifiers, for example because they are part of extensions that are not in modern customary use, and thus implementations may want to exclude them from identifiers. These include characters for historic and obsolete orthographies, characters used mostly liturgically, and in orthographies for languages used only in very small communities or with very limited current or declining us-*

age. Some characters also have architectural issues that may make them unsuitable for identifiers."

The **Identifier Type** property TR39#Table 1 recommendation should be mandatory, with the addition of the non-confusable **Technical** Identifier Type to be allowed.

I.e. `Limited_Use, Obsolete, Exclusion, Not_XID, Not_NFKC, Default_Ignorable, Deprecated, Not_Character` are not part of identifiers.

Allowed are Recommended, Inclusion, and all non-confusable Technical TR39 Identifier Types.

Note that several Technical Identifier_Type are confusable, but not marked as such. So far only the Latin letters U+1C0 ǀ, U+1C1 ǁ, U+1C3 ǃ which are confusable with operators.

Additionally the Halfwidth and Fullwidth Forms, U+FF00..U+FFEF are forbidden, even if allowed in TR31. They are confusable with the Latin base alphabet A-Z.

Additionally the Arabic Presentation Forms-A: U+FB50–U+FDFF and Arabic Presentation Forms-B: U+FE70–U+FEFF are now forbidden. Forms-A contains a list of Arabic presentation forms encoded as characters primarily for compatibility reasons. Forms-B are for compatibility with preexisting standards and legacy implementations that use these forms as character. Instead of these, letters from the Arabic block (U+0600..U+06FF) should be used for identifiers. See https://www.unicode.org/versions/Unicode14.0.0/ch09.pdf#G37489 and https://www.unicode.org/reports/tr53/. The TR39 idtype of these should be changed to Obsolete.

There are 79 Technical ranges added to the original list of Recommended and Inclusion ID's, with the confusables U+1C0..U+1C3 manually excluded.

```
grep ', U8ID_Technical' scripts.h | egrep -v 'Not_XID|U8ID_Obsolete|U8ID_Exclusion'
```

See Appendix E - IDType Technical.

10 TR39 Mixed Scripts

[TR39#5.2] defines some security profiles for identifers to avoid the most common identifier insecurities, that identifiers will stay identifiable.

We want to choose a variant of the **Moderately Restrictive** profile, with an exception for non-confusable Greek. I called this profile C26_4 or SAFEC26 in libu8ident.

- All identifiers in a document qualify as Single Script, or
- All identifiers in a document are covered by any of the following sets of scripts, according to the definition in Mixed Scripts:
  - Latin + Han + Hiragana + Katakana (Japanese),
  - Latin + Han + Bopomofo (Chinese),
  - Latin + Han + Hangul (Korean), or
- All identifiers in a document are covered by Latin and any one other Recommended script, except Cyrillic.
- Allow some Greek letters mixed with Latin, that are not confusable with Latin letters.

Greek alone is always allowed, as Cyrillic, but wherever we have a valid Latin letter which looks the same as the Greek counterpart, the Greek letter is forbidden, choose the Latin one instead. E.g. (A → A) GREEK CAPITAL LETTER ALPHA → LATIN CAPITAL LETTER A. See Appendix F for the generated list with 12 exceptions.

Thus it prevents Cyrillic mixed with Latin or any other script, but does allow any East-Asian CFK language, other common and widely used languages and Latin mixed with Greek, mainly used for its popular and actually used mathematical symbols. Many mathematical symbols already exists outside of Greek, but these are mainly used for operators in advanced programming languages, not as identifiers. See also http://xahlee.info/comp/unicode_math_operators.html for a nice overview.

E.g. here we have some:

- U+2217 (∗) ASTERISK OPERATOR (Script=Common). Not_XID
- U+2107 (ℇ) EULER CONSTANT (Script=Common, Lu) is a proper letter, but with Restricted IdentifierStatus.
- U+2126 (Ω) OHM SIGN (Script=Greek, L&) is a greek letter, but with Restricted IdentifierStatus.
- U+2127 (℧) INVERTED OHM SIGN (Script=Common, So). Obsolete, Not_XID
- U+0392 (Β → B) GREEK CAPITAL LETTER BETA → LATIN CAPITAL LETTER B Greek confusable
- U+03F2 (ϲ → c) GREEK LUNATE SIGMA SYMBOL → LATIN SMALL LETTER C Greek confusable
- U+0381 ; (α → a) GREEK SMALL LETTER ALPHA. Not confusable
- U+03F1 ; (ϱ → p) GREEK RHO SYMBOL → LATIN SMALL LETTER P. Not confusable
- U+03C3 ; (σ → o) GREEK SMALL LETTER SIGMA. Not confusable, but in the confusables.txt list. Used for the Stefan-Boltzmann constant.

- U+039A ; (K → K) GREEK CAPITAL LETTER KAPPA → LATIN CAPITAL LETTER K. Confusable.
- U+03BA ; (κ → κ) GREEK SMALL LETTER KAPPA → LATIN SMALL LETTER KRA. Confusable even if nobody uses the Latin counterpart.
- U+03C4 ; (τ → т) GREEK SMALL LETTER TAU → LATIN LETTER SMALL CAPITAL T. Confusable even if nobody uses the Latin counterpart.
- U+03A3 ; (Σ → Σ) GREEK CAPITAL LETTER SIGMA → LATIN CAPITAL LETTER ESH. Confusable even if nobody uses the Latin counterpart.
- U+03B2 ; (β → ß) GREEK SMALL LETTER BETA → LATIN SMALL LETTER SHARP S. Confusable and an edge-case.

And some actual C++ user-code representing the epsilon transport equation:

```
solve(div(U * ε) - div(νt * grad(ε)) / σε + C2 * ω * Sp(ε)
  == C1 * ω * G, ε, α);
```

TR39 also compiles a convenient IdentifierStatus list. But all the math letters with Script=Common from U+2100 to U+2200 are restricted, as Greek is forbidden mixed with Latin in the original TR39 Moderately Restrictive profile. Most are allowed according to the TR31 and TR39 rules of SAFEC26, so we need to come up with our own list of `XID_Start/XID_Continue` codepoints, excluding the Limited Use and Excluded scripts. And if an implementation choses to allow Excluded scripts with more logic to allow only this script.

Since the TR31 XID list also got the median positions wrong (for 98 Arabic codepoints), has some confusables with ops, and forgot about the Halfwidth and Fullwidth, U+FF00..U+FFEF confusables, we need to fixup and generate the XID lists by ourselves.

It is recommended to already exclude Limited Use and Excluded scripts from the initial list of identifier ranges, as this is the most common use-case, and shortens the common search paths. Only with the `#pragma Unicode ExcludedScript` search the full XID lists and the full scripts list.

The TR39 Mixed Scripts profile alone does not prevent from all spoofing attacks, but the additional rules from 8.3 “Combining marks script run detection for spoofing” are kept tiny.

11 Contexts (Scopes)

This is not discussed in any of the unicode security guidelines for identifiers. One could argue that a mixed-script profile is valid only

for a single identifier, or it is valid for the whole source file document. And there needs to be a definition if before or after the preprocessor, and if to treat names in private structs, classes and local names in functions as seperate contexts.

If valid for only a single identifier you could arbitralily mix up Cyrillic with Greek identifiers in a C++ namespace, and thus these identifiers would not be identifiable anymore, as both both can render to the very same glyphs. Thus we adopt the notion of identifier contexts.

With programming languages this is a source file, with objects files this is a module. For identifiers in object files there are open issues with binutils, linkers, exported identifiers, encodings. For filesystems this would be a directory.

For every source file we need to store a context with the list of already seen scripts and how many. The maximal number of scripts is 4, for the case of Japanese mixed with Latin. (`Katakana + Hiragana + Han + Latin`), thus we can save that list in a single 4-byte word, and the lookup and memory management is trivial.

Since the compiler sees the identifiers after the preprocessor included all headers, the context definition is a bit blurry. Is the context for mixed scripts an original source file (before cpp) or the resulting file after inclusion of all files (after cpp). This is similar to the problem with lexical variables a couple of decades ago.

1. **before-cpp**: One could argue that the scope of a variable should be contained in a lexical block, which can be statically determined and safely enclosed. With identifiers that would mean that the preprocessor already should perform the TR31 lexer checks and TR39 security checks, and one could define Arabic headers using private arabic fields, and include another header with Cyrillic only names. This would allow confusables in the resulting object file, and source files would be easy to check with external tools.
2. **private/scoped**: Another argument would be that all exported names end up in the object files and library flat, which would support the seperation of private and public name contexts, where to perform the mixed-script checks. Private contexts (e.g. static structs, private class fields, local names in functions) should be seperated from the rest. This would prevent from confusables in struct/class fields/methods, and the rest is seperated by the checks for the public names. Jabuk Jelinek favored this approach to the GCC -Whomoglyph PR answer: https://gcc.gnu.org/pipermail/gcc-patches/2021-November/583080.html

3. **after-cpp**: The third, strictest variant would define the context in the file after cpp. You would not be able to include a Cyrillic-only header, and you would not be able to use Cyrillic private fields. This would be the least surprising and most secure option. As long as the security risk lies ahead of us, one should go for the strictest option. Cyrillic header projects should be isolated and not used at all outside of non-cyrillic projects. I'm pointing the fingers at Cyrillic because it has the biggest number of confusables with Latin. Arabic headers e.g. are not all confusable with Latin or CFK, but I doubt that any non Hebrew/Arabic speaker can identify and see differences in its names without long training. Same for CFK and the other recommended scripts.

12 Implementations and Strategies

I implemented for cperl, a fork of perl5, the General Security profile "Moderately restrictive" (4) for identifiers in 2017, together with transparent normalization of NFC. This is a dynamic language with the need for fast tokenizing, and compilation. Still I did not see a need to restrict all source code identifiers to be already in NFC. Even with the added unicode checks and dynamic normalization the tokenizer is still faster than the simplier perl5 tokenizer.

Then when GCC went to full insecure identifiers I implemented the more general libu8ident library, which can be used with all known TR39 identifier type profiles, the mixed-script security profiles, TR31 XID character sets and all TR15 normalizations. There I tested various performance strategies of the unicode lookups. Tested was CRoaring, which was only useful for sets of single codepoints, the list of confusables. Most of the needed lists were best structured as binary-search in range pairs. Most of them were fastest with special-casing the codepoints below U+128 with a simple linear search. Binary search in an Eytzinger layout was not convincibly faster, neither hybrid searches by 1. splitting up ranges from single codepoints, nor 2. seperating 16bit from 32bit codepoints. Perfect hashes for singular lookup are used in some similar implementations, esp. for confusables and the normalization check. I'm still working on the perfect hashes approach (the new perl5 unicode tables, PostgresQL PerfectHash.pm, nbperf, gperf).

Rust has a good implementation also.

ICU has no implementation for TR39 checks (yet).

gcc PR 103027 implements the "skeleton" algorithm from TR39#4 via a switch table (!) for all decomposables and confusables, and two

extra dynamic hashtables for the NFD expanded identifiers. There's a lot of room of improvement there, e.g. with perfect hashes instead of a switch table for the huge and sparse tables, but they had no complaints on speed and size yet. Implementing the mixed-scripts approach in C++26/C26 for their -Whomoglyph warnings would be much faster and smaller though.

Also gcc and all other major compilers don't optimize large constant sparse case arrays to perfect hashes yet, so their switch/case lookups are linear, not constant. See e.g. https://programming.sirrida.de/hashsuper.pdf, which becomes a bottleneck just now with adopting large and sparse unicode switch statements, here with decompositions and confusables.

13 Appendix A - C26XID_Start

Created with mkc26 from libu8ident. *The SCX is modelled as if your compiler would allow static initialization of strings as {char,...,0}.*

```
struct sc {
    uint32_t from;
    uint32_t to;
    enum u8id_sc sc; // Scripts
    enum u8id_gc gc; // General Category. GC_L is L& (all letters)
                     // GC_V is varying
    const char *scx; // List of ScriptExtensions, maxsize 8 for U+1CF2
};

// Filtering allowed scripts, XID_Start, safe IDTypes, NFC, !MEDIAL and !MARK
// Ranges split at GC and SCX changes
const struct sc safec_start_list[335] = {
    {'$', '$', SC_Latin, GC_Sc, NULL},
    {'A', 'Z', SC_Latin, GC_Lu, NULL},
    {'_', '_', SC_Latin, GC_Pc, NULL},
    {'a', 'z', SC_Latin, GC_Ll, NULL},
    {0xC0, 0xD6, SC_Latin, GC_Lu, NULL}, //  À..Ö
    {0xD8, 0xF6, SC_Latin, GC_L, NULL}, //  Ø..ö
    {0xF8, 0x131, SC_Latin, GC_L, NULL}, //  ø..ı
    {0x134, 0x13E, SC_Latin, GC_L, NULL}, //  Ĵ..ľ
    {0x141, 0x148, SC_Latin, GC_L, NULL}, //  Ł..ň
    {0x14A, 0x17E, SC_Latin, GC_L, NULL}, //  Ŋ..ž
    {0x180, 0x180, SC_Latin, GC_Ll, NULL}, //  ƀ
    {0x18F, 0x18F, SC_Latin, GC_Lu, NULL}, //  Ə
    {0x1A0, 0x1A1, SC_Latin, GC_L, NULL}, //  Ơ..ơ
    {0x1AF, 0x1B0, SC_Latin, GC_L, NULL}, //  Ư..ư
```

```
{0x1CD, 0x1DC, SC_Latin, GC_L, NULL}, //  Ǎ..ǜ
{0x1DE, 0x1E3, SC_Latin, GC_L, NULL}, //  Ǟ..ǣ
{0x1E6, 0x1F0, SC_Latin, GC_L, NULL}, //  Ǧ..ǰ
{0x1F4, 0x1F5, SC_Latin, GC_L, NULL}, //  Ǵ..ǵ
{0x1F8, 0x21B, SC_Latin, GC_L, NULL}, //  Ǹ..ț
{0x21E, 0x21F, SC_Latin, GC_L, NULL}, //  Ȟ..ȟ
{0x226, 0x236, SC_Latin, GC_L, NULL}, //  Ȧ..ⱶ
{0x250, 0x252, SC_Latin, GC_Ll, NULL}, //  ɐ..ɒ
{0x255, 0x255, SC_Latin, GC_Ll, NULL}, //  ɕ
{0x258, 0x25A, SC_Latin, GC_Ll, NULL}, //  ɘ..ɚ
{0x25C, 0x262, SC_Latin, GC_Ll, NULL}, //  ɜ..ɢ
{0x264, 0x267, SC_Latin, GC_Ll, NULL}, //  ɤ..ɧ
{0x26A, 0x271, SC_Latin, GC_Ll, NULL}, //  ɪ..ɱ
{0x273, 0x276, SC_Latin, GC_Ll, NULL}, //  ɳ..ɶ
{0x278, 0x27B, SC_Latin, GC_Ll, NULL}, //  ɸ..ɻ
{0x27D, 0x288, SC_Latin, GC_Ll, NULL}, //  ɽ..ʈ
{0x28A, 0x291, SC_Latin, GC_Ll, NULL}, //  ʊ..ʑ
{0x293, 0x29D, SC_Latin, GC_L, NULL}, //  ʓ..ʝ
{0x29F, 0x2AF, SC_Latin, GC_Ll, NULL}, //  ʟ..ʯ
{0x2B9, 0x2C1, SC_Common, GC_Lm, NULL}, //  ʹ..ˁ
{0x2C6, 0x2D1, SC_Common, GC_Lm, NULL}, //  ˆ..ˑ
{0x2EC, 0x2EC, SC_Common, GC_Lm, NULL}, //  ˬ
{0x2EE, 0x2EE, SC_Common, GC_Lm, NULL}, //  ˮ
{0x37B, 0x37D, SC_Greek, GC_Ll, NULL}, //  ͻ..ͽ
{0x386, 0x386, SC_Greek, GC_Lu, NULL}, //  Ά
{0x388, 0x38A, SC_Greek, GC_Lu, NULL}, //  Έ..Ί
{0x38C, 0x38C, SC_Greek, GC_Lu, NULL}, //  Ό
{0x38E, 0x3A1, SC_Greek, GC_L, NULL}, //  Ύ..Ρ
{0x3A3, 0x3CF, SC_Greek, GC_L, NULL}, //  Σ..Ϗ
{0x3D7, 0x3D7, SC_Greek, GC_Ll, NULL}, //  ϗ
{0x3FC, 0x3FF, SC_Greek, GC_L, NULL}, //  ϼ..Ͽ
{0x401, 0x45F, SC_Cyrillic, GC_L, NULL}, //  Ё..џ
{0x48A, 0x4FF, SC_Cyrillic, GC_L, NULL}, //  Ҋ..ӿ
{0x510, 0x529, SC_Cyrillic, GC_L, NULL}, //  Ԑ..ԩ
{0x52E, 0x52F, SC_Cyrillic, GC_L, NULL}, //  Ԯ..ԯ
{0x531, 0x556, SC_Armenian, GC_Lu, NULL}, //  Ա..Ֆ
{0x559, 0x559, SC_Armenian, GC_Lm, NULL}, //  ՙ
{0x560, 0x586, SC_Armenian, GC_Ll, NULL}, //  ՠ..ֆ
{0x588, 0x588, SC_Armenian, GC_Ll, NULL}, //  ֈ
{0x5D0, 0x5EA, SC_Hebrew, GC_Lo, NULL}, //  א..ת
{0x5EF, 0x5F2, SC_Hebrew, GC_Lo, NULL}, //  ׯ..ײ
{0x620, 0x63F, SC_Arabic, GC_Lo, NULL}, //  ؠ..ؿ
{0x641, 0x64A, SC_Arabic, GC_Lo, NULL}, //  ف..ي
{0x671, 0x672, SC_Arabic, GC_Lo, NULL}, //  ٱ..ٲ
{0x674, 0x674, SC_Arabic, GC_Lo, NULL}, //  ٴ
{0x679, 0x68D, SC_Arabic, GC_Lo, NULL}, //  ٹ..ڍ
```

```
{0x68F, 0x6A0, SC_Arabic, GC_Lo, NULL}, //  □..□
{0x6A2, 0x6D3, SC_Arabic, GC_Lo, NULL}, //  □..□
{0x6D5, 0x6D5, SC_Arabic, GC_Lo, NULL}, //  □
{0x6E5, 0x6E6, SC_Arabic, GC_Lm, NULL}, //  □..□
{0x6EE, 0x6EF, SC_Arabic, GC_Lo, NULL}, //  □..□
{0x6FA, 0x6FC, SC_Arabic, GC_Lo, NULL}, //  □..□
{0x6FF, 0x6FF, SC_Arabic, GC_Lo, NULL}, //  □
{0x750, 0x77F, SC_Arabic, GC_Lo, NULL}, //  □..□
{0x781, 0x7A5, SC_Thaana, GC_Lo, NULL}, //  □..□
{0x7B1, 0x7B1, SC_Thaana, GC_Lo, NULL}, //  □
{0x870, 0x887, SC_Arabic, GC_Lo, NULL}, //  □..□
{0x889, 0x88E, SC_Arabic, GC_Lo, NULL}, //  □..□
{0x8A0, 0x8AC, SC_Arabic, GC_Lo, NULL}, //  □..□
{0x8B2, 0x8B2, SC_Arabic, GC_Lo, NULL}, //  □
{0x8B5, 0x8C9, SC_Arabic, GC_L, NULL}, //  □..□
{0x904, 0x939, SC_Devanagari, GC_Lo, NULL}, //  □..□
{0x93D, 0x93D, SC_Devanagari, GC_Lo, NULL}, //  □
{0x950, 0x950, SC_Devanagari, GC_Lo, NULL}, //  □
{0x960, 0x961, SC_Devanagari, GC_Lo, NULL}, //  □..□
{0x971, 0x977, SC_Devanagari, GC_L, NULL}, //  □..□
{0x979, 0x97F, SC_Devanagari, GC_Lo, NULL}, //  □..□
{0x985, 0x98C, SC_Bengali, GC_Lo, NULL}, //  □..□
{0x98F, 0x990, SC_Bengali, GC_Lo, NULL}, //  □..□
{0x993, 0x9A8, SC_Bengali, GC_Lo, NULL}, //  □..□
{0x9AA, 0x9B0, SC_Bengali, GC_Lo, NULL}, //  □..□
{0x9B2, 0x9B2, SC_Bengali, GC_Lo, NULL}, //  □
{0x9B6, 0x9B9, SC_Bengali, GC_Lo, NULL}, //  □..□
{0x9BD, 0x9BD, SC_Bengali, GC_Lo, NULL}, //  □
{0x9CE, 0x9CE, SC_Bengali, GC_Lo, NULL}, //  □
{0x9E0, 0x9E1, SC_Bengali, GC_Lo, NULL}, //  □..□
{0x9F0, 0x9F1, SC_Bengali, GC_Lo, NULL}, //  □..□
{0xA05, 0xA0A, SC_Gurmukhi, GC_Lo, NULL}, //  □..□
{0xA0F, 0xA10, SC_Gurmukhi, GC_Lo, NULL}, //  □..□
{0xA13, 0xA28, SC_Gurmukhi, GC_Lo, NULL}, //  □..□
{0xA2A, 0xA30, SC_Gurmukhi, GC_Lo, NULL}, //  □..□
{0xA32, 0xA32, SC_Gurmukhi, GC_Lo, NULL}, //  □
{0xA35, 0xA35, SC_Gurmukhi, GC_Lo, NULL}, //  □
{0xA38, 0xA39, SC_Gurmukhi, GC_Lo, NULL}, //  □..□
{0xA5C, 0xA5C, SC_Gurmukhi, GC_Lo, NULL}, //  □
{0xA72, 0xA74, SC_Gurmukhi, GC_Lo, NULL}, //  □..□
{0xA85, 0xA8D, SC_Gujarati, GC_Lo, NULL}, //  □..□
{0xA8F, 0xA91, SC_Gujarati, GC_Lo, NULL}, //  □..□
{0xA93, 0xAA8, SC_Gujarati, GC_Lo, NULL}, //  □..□
{0xAAA, 0xAB0, SC_Gujarati, GC_Lo, NULL}, //  □..□
{0xAB2, 0xAB3, SC_Gujarati, GC_Lo, NULL}, //  □..□
{0xAB5, 0xAB9, SC_Gujarati, GC_Lo, NULL}, //  □..□
```

```
{0xABD, 0xABD, SC_Gujarati, GC_Lo, NULL}, //  □
{0xAD0, 0xAD0, SC_Gujarati, GC_Lo, NULL}, //  □
{0xAE0, 0xAE1, SC_Gujarati, GC_Lo, NULL}, //  □..□
{0xB05, 0xB0C, SC_Oriya, GC_Lo, NULL}, //  □..□
{0xB0F, 0xB10, SC_Oriya, GC_Lo, NULL}, //  □..□
{0xB13, 0xB28, SC_Oriya, GC_Lo, NULL}, //  □..□
{0xB2A, 0xB30, SC_Oriya, GC_Lo, NULL}, //  □..□
{0xB32, 0xB33, SC_Oriya, GC_Lo, NULL}, //  □..□
{0xB35, 0xB39, SC_Oriya, GC_Lo, NULL}, //  □..□
{0xB3D, 0xB3D, SC_Oriya, GC_Lo, NULL}, //  □
{0xB5F, 0xB61, SC_Oriya, GC_Lo, NULL}, //  □..□
{0xB71, 0xB71, SC_Oriya, GC_Lo, NULL}, //  □
{0xB83, 0xB83, SC_Tamil, GC_Lo, NULL}, //  □
{0xB85, 0xB8A, SC_Tamil, GC_Lo, NULL}, //  □..□
{0xB8E, 0xB90, SC_Tamil, GC_Lo, NULL}, //  □..□
{0xB92, 0xB95, SC_Tamil, GC_Lo, NULL}, //  □..□
{0xB99, 0xB9A, SC_Tamil, GC_Lo, NULL}, //  □..□
{0xB9C, 0xB9C, SC_Tamil, GC_Lo, NULL}, //  □
{0xB9E, 0xB9F, SC_Tamil, GC_Lo, NULL}, //  □..□
{0xBA3, 0xBA4, SC_Tamil, GC_Lo, NULL}, //  □..□
{0xBA8, 0xBAA, SC_Tamil, GC_Lo, NULL}, //  □..□
{0xBAE, 0xBB9, SC_Tamil, GC_Lo, NULL}, //  □..□
{0xBD0, 0xBD0, SC_Tamil, GC_Lo, NULL}, //  □
{0xC05, 0xC0C, SC_Telugu, GC_Lo, NULL}, //  □..□
{0xC0E, 0xC10, SC_Telugu, GC_Lo, NULL}, //  □..□
{0xC12, 0xC28, SC_Telugu, GC_Lo, NULL}, //  □..□
{0xC2A, 0xC33, SC_Telugu, GC_Lo, NULL}, //  □..□
{0xC35, 0xC39, SC_Telugu, GC_Lo, NULL}, //  □..□
{0xC3D, 0xC3D, SC_Telugu, GC_Lo, NULL}, //  □
{0xC5D, 0xC5D, SC_Telugu, GC_Lo, NULL}, //  □
{0xC60, 0xC61, SC_Telugu, GC_Lo, NULL}, //  □..□
{0xC80, 0xC80, SC_Kannada, GC_Lo, NULL}, //  □
{0xC85, 0xC8C, SC_Kannada, GC_Lo, NULL}, //  □..□
{0xC8E, 0xC90, SC_Kannada, GC_Lo, NULL}, //  □..□
{0xC92, 0xCA8, SC_Kannada, GC_Lo, NULL}, //  □..□
{0xCAA, 0xCB3, SC_Kannada, GC_Lo, NULL}, //  □..□
{0xCB5, 0xCB9, SC_Kannada, GC_Lo, NULL}, //  □..□
{0xCBD, 0xCBD, SC_Kannada, GC_Lo, NULL}, //  □
{0xCDD, 0xCDD, SC_Kannada, GC_Lo, NULL}, //  □
{0xCE0, 0xCE1, SC_Kannada, GC_Lo, NULL}, //  □..□
{0xCF1, 0xCF2, SC_Kannada, GC_Lo, NULL}, //  □..□
{0xD05, 0xD0C, SC_Malayalam, GC_Lo, NULL}, //  □..□
{0xD0E, 0xD10, SC_Malayalam, GC_Lo, NULL}, //  □..□
{0xD12, 0xD3A, SC_Malayalam, GC_Lo, NULL}, //  □..□
{0xD3D, 0xD3D, SC_Malayalam, GC_Lo, NULL}, //  □
{0xD4E, 0xD4E, SC_Malayalam, GC_Lo, NULL}, //  □
```

```
{0xD54, 0xD56, SC_Malayalam, GC_Lo, NULL}, //  □..□
{0xD60, 0xD61, SC_Malayalam, GC_Lo, NULL}, //  □..□
{0xD7A, 0xD7F, SC_Malayalam, GC_Lo, NULL}, //  □..□
{0xD85, 0xD8E, SC_Sinhala, GC_Lo, NULL}, //  □..□
{0xD91, 0xD96, SC_Sinhala, GC_Lo, NULL}, //  □..□
{0xD9A, 0xDA5, SC_Sinhala, GC_Lo, NULL}, //  □..□
{0xDA7, 0xDB1, SC_Sinhala, GC_Lo, NULL}, //  □..□
{0xDB3, 0xDBB, SC_Sinhala, GC_Lo, NULL}, //  □..□
{0xDBD, 0xDBD, SC_Sinhala, GC_Lo, NULL}, //  □
{0xDC0, 0xDC6, SC_Sinhala, GC_Lo, NULL}, //  □..□
{0xE01, 0xE30, SC_Thai, GC_Lo, NULL}, //  □..□
{0xE32, 0xE32, SC_Thai, GC_Lo, NULL}, //  □
{0xE40, 0xE46, SC_Thai, GC_L, NULL}, //  □..□
{0xE81, 0xE82, SC_Lao, GC_Lo, NULL}, //  ກ..ຂ
{0xE84, 0xE84, SC_Lao, GC_Lo, NULL}, //  ຄ
{0xE86, 0xE8A, SC_Lao, GC_Lo, NULL}, //  □..ຊ
{0xE8C, 0xEA3, SC_Lao, GC_Lo, NULL}, //  □..ຣ
{0xEA5, 0xEA5, SC_Lao, GC_Lo, NULL}, //  ລ
{0xEA7, 0xEB0, SC_Lao, GC_Lo, NULL}, //  ວ..ະ
{0xEB2, 0xEB2, SC_Lao, GC_Lo, NULL}, //  າ
{0xEBD, 0xEBD, SC_Lao, GC_Lo, NULL}, //  □
{0xEC0, 0xEC4, SC_Lao, GC_Lo, NULL}, //  □..□
{0xEC6, 0xEC6, SC_Lao, GC_Lm, NULL}, //  □
{0xEDE, 0xEDF, SC_Lao, GC_Lo, NULL}, //  □..□
{0xF00, 0xF00, SC_Tibetan, GC_Lo, NULL}, //  □
{0xF40, 0xF42, SC_Tibetan, GC_Lo, NULL}, //  □..□
{0xF44, 0xF47, SC_Tibetan, GC_Lo, NULL}, //  □..□
{0xF49, 0xF4C, SC_Tibetan, GC_Lo, NULL}, //  □..□
{0xF4E, 0xF51, SC_Tibetan, GC_Lo, NULL}, //  □..□
{0xF53, 0xF56, SC_Tibetan, GC_Lo, NULL}, //  □..□
{0xF58, 0xF5B, SC_Tibetan, GC_Lo, NULL}, //  □..□
{0xF5D, 0xF68, SC_Tibetan, GC_Lo, NULL}, //  □..□
{0xF6A, 0xF6C, SC_Tibetan, GC_Lo, NULL}, //  □..□
{0xF88, 0xF8C, SC_Tibetan, GC_Lo, NULL}, //  □..□
{0x1000, 0x102A, SC_Myanmar, GC_Lo, NULL}, //  □..□
{0x103F, 0x103F, SC_Myanmar, GC_Lo, NULL}, //  □
{0x1050, 0x1055, SC_Myanmar, GC_Lo, NULL}, //  □..□
{0x105A, 0x105D, SC_Myanmar, GC_Lo, NULL}, //  □..□
{0x1061, 0x1061, SC_Myanmar, GC_Lo, NULL}, //  □
{0x1065, 0x1066, SC_Myanmar, GC_Lo, NULL}, //  □..□
{0x106E, 0x1070, SC_Myanmar, GC_Lo, NULL}, //  □..□
{0x1075, 0x1081, SC_Myanmar, GC_Lo, NULL}, //  □..□
{0x108E, 0x108E, SC_Myanmar, GC_Lo, NULL}, //  □
{0x10C7, 0x10C7, SC_Georgian, GC_Lu, NULL}, //  □
{0x10CD, 0x10CD, SC_Georgian, GC_Lu, NULL}, //  □
{0x10D0, 0x10F0, SC_Georgian, GC_Ll, NULL}, //  ა..ჰ
```

```
{0x10F7, 0x10FA, SC_Georgian, GC_Ll, NULL}, //  ⴗ..ⴚ
{0x10FD, 0x10FF, SC_Georgian, GC_Ll, NULL}, //  ⴝ..ⴟ
{0x1200, 0x1248, SC_Ethiopic, GC_Lo, NULL}, //  ሀ..ቈ
{0x124A, 0x124D, SC_Ethiopic, GC_Lo, NULL}, //  ቊ..ቍ
{0x1250, 0x1256, SC_Ethiopic, GC_Lo, NULL}, //  ቐ..ቖ
{0x1258, 0x1258, SC_Ethiopic, GC_Lo, NULL}, //  ቘ
{0x125A, 0x125D, SC_Ethiopic, GC_Lo, NULL}, //  ቚ..ቝ
{0x1260, 0x1288, SC_Ethiopic, GC_Lo, NULL}, //  በ..ቈ
{0x128A, 0x128D, SC_Ethiopic, GC_Lo, NULL}, //  ቊ..ቍ
{0x1290, 0x12B0, SC_Ethiopic, GC_Lo, NULL}, //  ነ..ኰ
{0x12B2, 0x12B5, SC_Ethiopic, GC_Lo, NULL}, //  ኲ..ኵ
{0x12B8, 0x12BE, SC_Ethiopic, GC_Lo, NULL}, //  ኸ..ኾ
{0x12C0, 0x12C0, SC_Ethiopic, GC_Lo, NULL}, //  ዀ
{0x12C2, 0x12C5, SC_Ethiopic, GC_Lo, NULL}, //  ዂ..ዅ
{0x12C8, 0x12D6, SC_Ethiopic, GC_Lo, NULL}, //  ወ..ዖ
{0x12D8, 0x1310, SC_Ethiopic, GC_Lo, NULL}, //  ዘ..ጐ
{0x1312, 0x1315, SC_Ethiopic, GC_Lo, NULL}, //  ጒ..ጕ
{0x1318, 0x135A, SC_Ethiopic, GC_Lo, NULL}, //  ጘ..ፚ
{0x1380, 0x138F, SC_Ethiopic, GC_Lo, NULL}, //  ᎀ..ᎏ
{0x1780, 0x17A2, SC_Khmer, GC_Lo, NULL}, //  ក..អ
{0x17A5, 0x17A7, SC_Khmer, GC_Lo, NULL}, //  ឥ..ឧ
{0x17A9, 0x17B3, SC_Khmer, GC_Lo, NULL}, //  ឩ..ឳ
{0x17D7, 0x17D7, SC_Khmer, GC_Lm, NULL}, //  ៗ
{0x17DC, 0x17DC, SC_Khmer, GC_Lo, NULL}, //  ៜ
{0x1C90, 0x1CBA, SC_Georgian, GC_Lu, NULL}, //  Ა..Ჺ
{0x1CBD, 0x1CBF, SC_Georgian, GC_Lu, NULL}, //  Ჽ..Ჿ
{0x1D00, 0x1D25, SC_Latin, GC_Ll, NULL}, //  ᴀ..ᴥ
{0x1D27, 0x1D2A, SC_Greek, GC_Ll, NULL}, //  ᴧ..ᴪ
{0x1D2F, 0x1D2F, SC_Latin, GC_Lm, NULL}, //  ᴯ
{0x1D3B, 0x1D3B, SC_Latin, GC_Lm, NULL}, //  ᴻ
{0x1D4E, 0x1D4E, SC_Latin, GC_Lm, NULL}, //  ᵎ
{0x1D6B, 0x1D77, SC_Latin, GC_Ll, NULL}, //  ᵫ..ᵷ
{0x1D79, 0x1D9A, SC_Latin, GC_Ll, NULL}, //  ᵹ..ᶚ
{0x1E00, 0x1E99, SC_Latin, GC_L, NULL}, //  Ḁ..ẙ
{0x1E9C, 0x1EFF, SC_Latin, GC_L, NULL}, //  ẜ..ỿ
{0x1F01, 0x1F15, SC_Greek, GC_L, NULL}, //  ἁ..ἕ
{0x1F18, 0x1F1D, SC_Greek, GC_Lu, NULL}, //  Ἐ..Ἕ
{0x1F20, 0x1F45, SC_Greek, GC_L, NULL}, //  ἠ..ὅ
{0x1F48, 0x1F4D, SC_Greek, GC_Lu, NULL}, //  Ὀ..Ὅ
{0x1F50, 0x1F57, SC_Greek, GC_Ll, NULL}, //  ὐ..ὗ
{0x1F59, 0x1F59, SC_Greek, GC_Lu, NULL}, //  Ὑ
{0x1F5B, 0x1F5B, SC_Greek, GC_Lu, NULL}, //  Ὓ
{0x1F5D, 0x1F5D, SC_Greek, GC_Lu, NULL}, //  Ὕ
{0x1F5F, 0x1F70, SC_Greek, GC_L, NULL}, //  Ὗ..ὰ
{0x1F72, 0x1F72, SC_Greek, GC_Ll, NULL}, //  ὲ
{0x1F74, 0x1F74, SC_Greek, GC_Ll, NULL}, //  ὴ
```

```
{0x1F76, 0x1F76, SC_Greek, GC_Ll, NULL}, //  ὶ
{0x1F78, 0x1F78, SC_Greek, GC_Ll, NULL}, //  ὸ
{0x1F7A, 0x1F7A, SC_Greek, GC_Ll, NULL}, //  ὺ
{0x1F7C, 0x1F7C, SC_Greek, GC_Ll, NULL}, //  ὼ
{0x1F80, 0x1FB4, SC_Greek, GC_L, NULL}, //  ᾀ..ά
{0x1FB6, 0x1FBA, SC_Greek, GC_L, NULL}, //  ᾶ..Ὰ
{0x1FBC, 0x1FBC, SC_Greek, GC_Lt, NULL}, //  ᾼ
{0x1FC2, 0x1FC4, SC_Greek, GC_Ll, NULL}, //  ῂ..ή
{0x1FC6, 0x1FC8, SC_Greek, GC_L, NULL}, //  ῆ..Ὲ
{0x1FCA, 0x1FCA, SC_Greek, GC_Lu, NULL}, //  Ὴ
{0x1FCC, 0x1FCC, SC_Greek, GC_Lt, NULL}, //  ῌ
{0x1FD0, 0x1FD2, SC_Greek, GC_Ll, NULL}, //  ῐ..ΐ
{0x1FD6, 0x1FDA, SC_Greek, GC_L, NULL}, //  ῖ..Ὶ
{0x1FE0, 0x1FE2, SC_Greek, GC_Ll, NULL}, //  ῠ..ΰ
{0x1FE4, 0x1FEA, SC_Greek, GC_L, NULL}, //  ῤ..Ὺ
{0x1FEC, 0x1FEC, SC_Greek, GC_Lu, NULL}, //  Ῥ
{0x1FF2, 0x1FF4, SC_Greek, GC_Ll, NULL}, //  ῲ..ώ
{0x1FF6, 0x1FF8, SC_Greek, GC_L, NULL}, //  ῶ..Ὸ
{0x1FFA, 0x1FFA, SC_Greek, GC_Lu, NULL}, //  Ὼ
{0x1FFC, 0x1FFC, SC_Greek, GC_Lt, NULL}, //  ῼ
{0x2118, 0x2118, SC_Common, GC_Sm, NULL}, //  ℘
{0x212E, 0x212E, SC_Common, GC_So, NULL}, //  ℮
{0x2C60, 0x2C67, SC_Latin, GC_L, NULL}, //  Ⱡ..ⱨ
{0x2C77, 0x2C7B, SC_Latin, GC_Ll, NULL}, //  ⱷ..ⱻ
{0x2D27, 0x2D27, SC_Georgian, GC_Ll, NULL}, //  ⴧ
{0x2D2D, 0x2D2D, SC_Georgian, GC_Ll, NULL}, //  ⴭ
{0x2D80, 0x2D96, SC_Ethiopic, GC_Lo, NULL}, //  ⶀ..ⶖ
{0x2DA0, 0x2DA6, SC_Ethiopic, GC_Lo, NULL}, //  ⶠ..ⶦ
{0x2DA8, 0x2DAE, SC_Ethiopic, GC_Lo, NULL}, //  ⶨ..ⶮ
{0x2DB0, 0x2DB6, SC_Ethiopic, GC_Lo, NULL}, //  ⶰ..ⶶ
{0x2DB8, 0x2DBE, SC_Ethiopic, GC_Lo, NULL}, //  ⶸ..ⶾ
{0x2DC0, 0x2DC6, SC_Ethiopic, GC_Lo, NULL}, //  ⷀ..ⷆ
{0x2DC8, 0x2DCE, SC_Ethiopic, GC_Lo, NULL}, //  ⷈ..ⷎ
{0x2DD0, 0x2DD6, SC_Ethiopic, GC_Lo, NULL}, //  ⷐ..ⷖ
{0x2DD8, 0x2DDE, SC_Ethiopic, GC_Lo, NULL}, //  ⷘ..ⷞ
{0x3005, 0x3005, SC_Han, GC_Lm, NULL}, //  々
{0x3007, 0x3007, SC_Han, GC_Nl, NULL}, //  〇
{0x3021, 0x3029, SC_Han, GC_Nl, NULL}, //  〡..〩
{0x3031, 0x3035, SC_Common, GC_Lm, {SC_Hiragana,SC_Katakana,0}}, //  〱..〵
{0x303B, 0x303B, SC_Han, GC_Lm, NULL}, //  〻
{0x3041, 0x3096, SC_Hiragana, GC_Lo, NULL}, //  ぁ..ゖ
{0x309D, 0x309E, SC_Hiragana, GC_Lm, NULL}, //  ゝ..ゞ
{0x30A1, 0x30FA, SC_Katakana, GC_Lo, NULL}, //  ァ..ヺ
{0x30FC, 0x30FC, SC_Common, GC_Lm, {SC_Hiragana,SC_Katakana,0}}, //  ー
{0x30FE, 0x30FE, SC_Katakana, GC_Lm, NULL}, //  ヾ
{0x3105, 0x312D, SC_Bopomofo, GC_Lo, NULL}, //  ㄅ..ㄭ
```

```
{0x312F, 0x312F, SC_Bopomofo, GC_Lo, NULL}, //  □
{0x31A0, 0x31BF, SC_Bopomofo, GC_Lo, NULL}, //  □..□
{0x3400, 0x4DBF, SC_Han, GC_Lo, NULL}, //  □..□
{0x4E00, 0x9FFF, SC_Han, GC_Lo, NULL}, //  □..□
{0xA67F, 0xA67F, SC_Cyrillic, GC_Lm, NULL}, //  □
{0xA717, 0xA71F, SC_Common, GC_Lm, NULL}, //  □..ˏ
{0xA788, 0xA788, SC_Common, GC_Lm, NULL}, //  □
{0xA78D, 0xA78E, SC_Latin, GC_L, NULL}, //  Ɥ..ɬ
{0xA792, 0xA793, SC_Latin, GC_L, NULL}, //  □..□
{0xA7AA, 0xA7AA, SC_Latin, GC_Lu, NULL}, //  Ħ
{0xA7AE, 0xA7AF, SC_Latin, GC_L, NULL}, //  □..□
{0xA7B8, 0xA7CA, SC_Latin, GC_L, NULL}, //  □..□
{0xA7D0, 0xA7D1, SC_Latin, GC_L, NULL}, //  □..□
{0xA7D3, 0xA7D3, SC_Latin, GC_Ll, NULL}, //  □
{0xA7D5, 0xA7D9, SC_Latin, GC_L, NULL}, //  □..□
{0xA7FA, 0xA7FA, SC_Latin, GC_Ll, NULL}, //  □
{0xA9E7, 0xA9EF, SC_Myanmar, GC_Lo, NULL}, //  □..□
{0xA9FA, 0xA9FE, SC_Myanmar, GC_Lo, NULL}, //  □..□
{0xAA60, 0xAA76, SC_Myanmar, GC_L, NULL}, //  □..□
{0xAA7A, 0xAA7A, SC_Myanmar, GC_Lo, NULL}, //  □
{0xAA7E, 0xAA7F, SC_Myanmar, GC_Lo, NULL}, //  □..□
{0xAB01, 0xAB06, SC_Ethiopic, GC_Lo, NULL}, //  □..□
{0xAB09, 0xAB0E, SC_Ethiopic, GC_Lo, NULL}, //  □..□
{0xAB11, 0xAB16, SC_Ethiopic, GC_Lo, NULL}, //  □..□
{0xAB20, 0xAB26, SC_Ethiopic, GC_Lo, NULL}, //  □..□
{0xAB28, 0xAB2E, SC_Ethiopic, GC_Lo, NULL}, //  □..□
{0xAB66, 0xAB68, SC_Latin, GC_Ll, NULL}, //  □..□
{0xFA0E, 0xFA0F, SC_Han, GC_Lo, NULL}, //  □..□
{0xFA11, 0xFA11, SC_Han, GC_Lo, NULL}, //  □
{0xFA13, 0xFA14, SC_Han, GC_Lo, NULL}, //  □..□
{0xFA1F, 0xFA1F, SC_Han, GC_Lo, NULL}, //  □
{0xFA21, 0xFA21, SC_Han, GC_Lo, NULL}, //  □
{0xFA23, 0xFA24, SC_Han, GC_Lo, NULL}, //  □..□
{0xFA27, 0xFA29, SC_Han, GC_Lo, NULL}, //  □..□
{0x1B11F, 0x1B11F, SC_Hiragana, GC_Lo, NULL}, //  □
{0x1B121, 0x1B122, SC_Katakana, GC_Lo, NULL}, //  □..□
{0x1B150, 0x1B152, SC_Hiragana, GC_Lo, NULL}, //  □..□
{0x1B164, 0x1B167, SC_Katakana, GC_Lo, NULL}, //  □..□
{0x1DF00, 0x1DF1E, SC_Latin, GC_L, NULL}, //  □..□
{0x1E7E0, 0x1E7E6, SC_Ethiopic, GC_Lo, NULL}, //  □..□
{0x1E7E8, 0x1E7EB, SC_Ethiopic, GC_Lo, NULL}, //  □..□
{0x1E7ED, 0x1E7EE, SC_Ethiopic, GC_Lo, NULL}, //  □..□
{0x1E7F0, 0x1E7FE, SC_Ethiopic, GC_Lo, NULL}, //  □..□
{0x20000, 0x2A6DF, SC_Han, GC_Lo, NULL}, //  □..□
{0x2A700, 0x2B738, SC_Han, GC_Lo, NULL}, //  □..□
{0x2B740, 0x2B81D, SC_Han, GC_Lo, NULL}, //  □..□
```

```
    {0x2B820, 0x2CEA1, SC_Han, GC_Lo, NULL}, //  □..□
    {0x2CEB0, 0x2EBE0, SC_Han, GC_Lo, NULL}, //  □..□
    {0x30000, 0x3134A, SC_Han, GC_Lo, NULL}, //  □..□
};
// 243 ranges, 92 singles, 95986 codepoints
```

14 Appendix B - C26XID_Continue

Created with mkc26 from libu8ident. *The SCX is modelled as if your compiler would allow static initialization of strings as {char,...,0}.*

```
// Filtering allowed scripts, XID_Continue,!XID_Start, safe IDTypes, NFC,
// and !MARK. Split on GC and SCX
const struct sc safec_cont_list[21] = {
    {0x30, 0x39, SC_Common, GC_Nd, NULL}, //  0..9
    {0x5F, 0x5F, SC_Common, GC_Pc, NULL}, //  _
    {0xB7, 0xB7, SC_Common, GC_Po, NULL}, //  ·
    {0x660, 0x669, SC_Arabic, GC_Nd, {SC_Arabic,SC_Thaana,SC_Yezidi,0}}, //  □..□
    {0x6F0, 0x6F9, SC_Arabic, GC_Nd, NULL}, //  □..□
    {0x966, 0x96F, SC_Devanagari, GC_Nd, {SC_Devanagari,SC_Dogra,SC_Kaithi,
        SC_Mahajani,0}}, //  □..□
    {0x9E6, 0x9EF, SC_Bengali, GC_Nd, {SC_Bengali,SC_Chakma,SC_Syloti_Nagri,0}},
    {0xA66, 0xA6F, SC_Gurmukhi, GC_Nd, {SC_Gurmukhi,SC_Multani,0}}, //  □..□
    {0xAE6, 0xAEF, SC_Gujarati, GC_Nd, {SC_Gujarati,SC_Khojki,0}, //  □..□
    {0xB66, 0xB6F, SC_Oriya, GC_Nd, NULL}, //  □..□
    {0xBE6, 0xBEF, SC_Tamil, GC_Nd, {SC_Grantha,SC_Tamil,0}}, //  □..□
    {0xC66, 0xC6F, SC_Telugu, GC_Nd, NULL}, //  □..□
    {0xCE6, 0xCEF, SC_Kannada, GC_Nd, {SC_Kannada,SC_Nandinagari,0}}, //  □..□
    {0xD66, 0xD6F, SC_Malayalam, GC_Nd, NULL}, //  □..□
    {0xE50, 0xE59, SC_Thai, GC_Nd, NULL}, //  □..□
    {0xED0, 0xED9, SC_Lao, GC_Nd, NULL}, //  □..□
    {0xF20, 0xF29, SC_Tibetan, GC_Nd, NULL}, //  □..□
    {0x1040, 0x1049, SC_Myanmar, GC_Nd, {SC_Chakma,SC_Myanmar,SC_Tai_Le,0}},
    {0x1090, 0x1099, SC_Myanmar, GC_Nd, NULL}, //  □..□
    {0x17E0, 0x17E9, SC_Khmer, GC_Nd, NULL}, //  □..□
    {0x203F, 0x2040, SC_Common, GC_Pc, NULL}, //  ‿..□
    {0xA9F0, 0xA9F9, SC_Myanmar, GC_Nd, NULL}, //  □..□
};
// 20 ranges, 1 singles, 172 codepoints
```

15 Appendix C - XID_Continue # Lm

Needed for the combining marks special-cases in Section 8.3 8.3 Combining marks script run detection for spoofing, which is needed

for TR39#5.4 and TR31#2.2 checks.

Practically this list is not needed, as only the 4 Japanese PROLONGED SOUND MARKs need to be checked. All other Lm Modifier Letters are freestanding base characters, which can be combined with any other letter.

67 matches for “XID_Continue # Lm” in buffer: DerivedCoreProperties.txt

```
02B0..02C1    ; XID_Continue # Lm  [18] MODIFIER LETTER SMALL H..
                                        MODIFIER LETTER REVERSED GLOTTAL STOP
02C6..02D1    ; XID_Continue # Lm  [12] MODIFIER LETTER CIRCUMFLEX ACCENT..
                                        MODIFIER LETTER HALF TRIANGULAR COLON
02E0..02E4    ; XID_Continue # Lm   [5] MODIFIER LETTER SMALL GAMMA..
                                        MODIFIER LETTER SMALL REVERSED GLOTTAL STOP
02EC          ; XID_Continue # Lm       MODIFIER LETTER VOICING
02EE          ; XID_Continue # Lm       MODIFIER LETTER DOUBLE APOSTROPHE
0374          ; XID_Continue # Lm       GREEK NUMERAL SIGN
0559          ; XID_Continue # Lm       ARMENIAN MODIFIER LETTER LEFT HALF RING
0640          ; XID_Continue # Lm       ARABIC TATWEEL
06E5..06E6    ; XID_Continue # Lm   [2] ARABIC SMALL WAW..
                                        ARABIC SMALL YEH
07F4..07F5    ; XID_Continue # Lm   [2] NKO HIGH TONE APOSTROPHE..
                                        NKO LOW TONE APOSTROPHE
07FA          ; XID_Continue # Lm       NKO LAJANYALAN
081A          ; XID_Continue # Lm       SAMARITAN MODIFIER LETTER EPENTHETIC YUT
0824          ; XID_Continue # Lm       SAMARITAN MODIFIER LETTER SHORT A
0828          ; XID_Continue # Lm       SAMARITAN MODIFIER LETTER I
08C9          ; XID_Continue # Lm       ARABIC SMALL FARSI YEH
0971          ; XID_Continue # Lm       DEVANAGARI SIGN HIGH SPACING DOT
0E46          ; XID_Continue # Lm       THAI CHARACTER MAIYAMOK
0EC6          ; XID_Continue # Lm       LAO KO LA
10FC          ; XID_Continue # Lm       MODIFIER LETTER GEORGIAN NAR
17D7          ; XID_Continue # Lm       KHMER SIGN LEK TOO
1843          ; XID_Continue # Lm       MONGOLIAN LETTER TODO LONG VOWEL SIGN
1AA7          ; XID_Continue # Lm       TAI THAM SIGN MAI YAMOK
1C78..1C7D    ; XID_Continue # Lm   [6] OL CHIKI MU TTUDDAG..OL CHIKI AHAD
1D2C..1D6A    ; XID_Continue # Lm  [63] MODIFIER LETTER CAPITAL A..
                                        GREEK SUBSCRIPT SMALL LETTER CHI
1D78          ; XID_Continue # Lm       MODIFIER LETTER CYRILLIC EN
1D9B..1DBF    ; XID_Continue # Lm  [37] MODIFIER LETTER SMALL TURNED ALPHA..
                                        MODIFIER LETTER SMALL THETA
2071          ; XID_Continue # Lm       SUPERSCRIPT LATIN SMALL LETTER I
207F          ; XID_Continue # Lm       SUPERSCRIPT LATIN SMALL LETTER N
2090..209C    ; XID_Continue # Lm  [13] LATIN SUBSCRIPT SMALL LETTER A..
                                        LATIN SUBSCRIPT SMALL LETTER T
```

```
2C7C..2C7D    ; XID_Continue # Lm   [2] LATIN SUBSCRIPT SMALL LETTER J..
                                        MODIFIER LETTER CAPITAL V
2D6F          ; XID_Continue # Lm       TIFINAGH MODIFIER LETTER LABIALIZATION MARK
3005          ; XID_Continue # Lm       IDEOGRAPHIC ITERATION MARK
3031..3035    ; XID_Continue # Lm   [5] VERTICAL KANA REPEAT MARK..
                                        VERTICAL KANA REPEAT MARK LOWER HALF
303B          ; XID_Continue # Lm       VERTICAL IDEOGRAPHIC ITERATION MARK
309D..309E    ; XID_Continue # Lm   [2] HIRAGANA ITERATION MARK..
                                        HIRAGANA VOICED ITERATION MARK
30FC..30FE    ; XID_Continue # Lm   [3] KATAKANA-HIRAGANA PROLONGED SOUND MARK..
                                        KATAKANA VOICED ITERATION MARK
A015          ; XID_Continue # Lm       YI SYLLABLE WU
A4F8..A4FD    ; XID_Continue # Lm   [6] LISU LETTER TONE MYA TI..
                                        LISU LETTER TONE MYA JEU
A60C          ; XID_Continue # Lm       VAI SYLLABLE LENGTHENER
A67F          ; XID_Continue # Lm       CYRILLIC PAYEROK
A69C..A69D    ; XID_Continue # Lm   [2] MODIFIER LETTER CYRILLIC HARD SIGN..
                                        MODIFIER LETTER CYRILLIC SOFT SIGN
A717..A71F    ; XID_Continue # Lm   [9] MODIFIER LETTER DOT VERTICAL BAR..
                                        LOW INVERTED EXCLAMATION MARK
A770          ; XID_Continue # Lm       MODIFIER LETTER US
A788          ; XID_Continue # Lm       MODIFIER LETTER LOW CIRCUMFLEX ACCENT
A7F2..A7F4    ; XID_Continue # Lm   [3] MODIFIER LETTER CAPITAL C..
                                        MODIFIER LETTER CAPITAL Q
A7F8..A7F9    ; XID_Continue # Lm   [2] MODIFIER LETTER CAPITAL H WITH STROKE..
                                        MODIFIER LETTER SMALL LIGATURE OE
A9CF          ; XID_Continue # Lm       JAVANESE PANGRANGKEP
A9E6          ; XID_Continue # Lm       MYANMAR MODIFIER LETTER SHAN REDUPLICATION
AA70          ; XID_Continue # Lm       MYANMAR MODIFIER LETTER KHAMTI REDUPLICATION
AADD          ; XID_Continue # Lm       TAI VIET SYMBOL SAM
AAF3..AAF4    ; XID_Continue # Lm   [2] MEETEI MAYEK SYLLABLE REPETITION MARK..
                                        MEETEI MAYEK WORD REPETITION MARK
AB5C..AB5F    ; XID_Continue # Lm   [4] MODIFIER LETTER SMALL HENG..
                                        MODIFIER LETTER SMALL U WITH LEFT HOOK
AB69          ; XID_Continue # Lm       MODIFIER LETTER SMALL TURNED W
FF70          ; XID_Continue # Lm       HALFWIDTH KATA-HIRA PROLONGED SOUND MARK
FF9E..FF9F    ; XID_Continue # Lm   [2] HALFWIDTH KATAKANA VOICED SOUND MARK..
                                        SEMI-VOICED SOUND MARK
10780..10785  ; XID_Continue # Lm   [6] MODIFIER LETTER SMALL CAPITAL AA..
                                        MODIFIER LETTER SMALL B WITH HOOK
10787..107B0  ; XID_Continue # Lm  [42] MODIFIER LETTER SMALL DZ DIGRAPH..
                                        MODIFIER LETTER SMALL V WITH RIGHT HOOK
107B2..107BA  ; XID_Continue # Lm   [9] MODIFIER LETTER SMALL CAPITAL Y..
                                        MODIFIER LETTER SMALL S WITH CURL
16B40..16B43  ; XID_Continue # Lm   [4] PAHAWH HMONG SIGN VOS SEEV..
                                        PAHAWH HMONG SIGN IB YAM
```

```
16F93..16F9F  ; XID_Continue # Lm  [13] MIAO LETTER TONE-2..
                                       MIAO LETTER REFORMED TONE-8
16FE0..16FE1  ; XID_Continue # Lm   [2] TANGUT ITERATION MARK..
                                       NUSHU ITERATION MARK
16FE3         ; XID_Continue # Lm       OLD CHINESE ITERATION MARK
1AFF0..1AFF3  ; XID_Continue # Lm   [4] KATAKANA LETTER MINNAN TONE-2..
                                       KATAKANA LETTER MINNAN TONE-5
1AFF5..1AFFB  ; XID_Continue # Lm   [7] KATAKANA LETTER MINNAN TONE-7..
                                       KATAKANA LETTER MINNAN NASALIZED TONE-5
1AFFD..1AFFE  ; XID_Continue # Lm   [2] KATAKANA LETTER MINNAN NASALIZED TONE-7..
                                       KATAKANA LETTER MINNAN NASALIZED TONE-8
1E137..1E13D  ; XID_Continue # Lm   [7] NYIAKENG PUACHUE HMONG SIGN FOR PERSON..
                                       NYIAKENG PUACHUE HMONG SYLLABLE LENGTHENER
1E94B         ; XID_Continue # Lm       ADLAM NASALIZATION MARK
```

16 Appendix D - XID_Continue # M

Needed for the combining marks checks in Section 8.3 8.3 Combining marks script run detection for spoofing.

513 matches for “XID_Continue # M” in buffer: DerivedCoreProperties.txt

```
0300..036F    ; XID_Continue # Mn [112] COMBINING GRAVE ACCENT..
                                       COMBINING LATIN SMALL LETTER X
0483..0487    ; XID_Continue # Mn   [5] COMBINING CYRILLIC TITLO..
                                       COMBINING CYRILLIC POKRYTIE
0591..05BD    ; XID_Continue # Mn  [45] HEBREW ACCENT ETNAHTA..
                                       HEBREW POINT METEG
05BF          ; XID_Continue # Mn       HEBREW POINT RAFE
05C1..05C2    ; XID_Continue # Mn   [2] HEBREW POINT SHIN DOT..
                                       HEBREW POINT SIN DOT
05C4..05C5    ; XID_Continue # Mn   [2] HEBREW MARK UPPER DOT..
                                       HEBREW MARK LOWER DOT
05C7          ; XID_Continue # Mn       HEBREW POINT QAMATS QATAN
0610..061A    ; XID_Continue # Mn  [11] ARABIC SIGN SALLALLAHOU ALAYHE WASSALLAM..
                                       ARABIC SMALL KASRA
064B..065F    ; XID_Continue # Mn  [21] ARABIC FATHATAN..
                                       ARABIC WAVY HAMZA BELOW
0670          ; XID_Continue # Mn       ARABIC LETTER SUPERSCRIPT ALEF
06D6..06DC    ; XID_Continue # Mn   [7] ARABIC SMALL HIGH LIGATURE SAD WITH LAM
                                       WITH ALEF MAKSURA..HIGH SEEN
06DF..06E4    ; XID_Continue # Mn   [6] ARABIC SMALL HIGH ROUNDED ZERO..MADDA
06E7..06E8    ; XID_Continue # Mn   [2] ARABIC SMALL HIGH YEH..NOON
06EA..06ED    ; XID_Continue # Mn   [4] ARABIC EMPTY CENTRE LOW STOP..MEEM
0711          ; XID_Continue # Mn       SYRIAC LETTER SUPERSCRIPT ALAPH
```

```
0730..074A    ; XID_Continue # Mn  [27] SYRIAC PTHAHA ABOVE..BARREKH
07A6..07B0    ; XID_Continue # Mn  [11] THAANA ABAFILI..THAANA SUKUN
07EB..07F3    ; XID_Continue # Mn   [9] NKO COMBINING SHORT HIGH TONE..
                                        NKO COMBINING DOUBLE DOT ABOVE
07FD          ; XID_Continue # Mn       NKO DANTAYALAN
0816..0819    ; XID_Continue # Mn   [4] SAMARITAN MARK IN..
                                        SAMARITAN MARK DAGESH
081B..0823    ; XID_Continue # Mn   [9] SAMARITAN MARK EPENTHETIC YUT..
                                        SAMARITAN VOWEL SIGN A
0825..0827    ; XID_Continue # Mn   [3] SAMARITAN VOWEL SIGN SHORT A..SIGN U
0829..082D    ; XID_Continue # Mn   [5] SAMARITAN VOWEL SIGN LONG I..
                                        SAMARITAN MARK NEQUDAA
0859..085B    ; XID_Continue # Mn   [3] MANDAIC AFFRICATION MARK..
                                        MANDAIC GEMINATION MARK
0898..089F    ; XID_Continue # Mn   [8] ARABIC SMALL HIGH WORD AL-JUZ..
                                        ARABIC HALF MADDA OVER MADDA
08CA..08E1    ; XID_Continue # Mn  [24] ARABIC SMALL HIGH FARSI YEH..
                                        ARABIC SMALL HIGH SIGN SAFHA
08E3..0902    ; XID_Continue # Mn  [32] ARABIC TURNED DAMMA BELOW..
                                        DEVANAGARI SIGN ANUSVARA
0903          ; XID_Continue # Mc       DEVANAGARI SIGN VISARGA
093A          ; XID_Continue # Mn       DEVANAGARI VOWEL SIGN OE
093B          ; XID_Continue # Mc       DEVANAGARI VOWEL SIGN OOE
093C          ; XID_Continue # Mn       DEVANAGARI SIGN NUKTA
093E..0940    ; XID_Continue # Mc   [3] DEVANAGARI VOWEL SIGN AA..II
0941..0948    ; XID_Continue # Mn   [8] DEVANAGARI VOWEL SIGN U..AI
0949..094C    ; XID_Continue # Mc   [4] DEVANAGARI VOWEL SIGN CANDRA O..AU
094D          ; XID_Continue # Mn       DEVANAGARI SIGN VIRAMA
094E..094F    ; XID_Continue # Mc   [2] DEVANAGARI VOWEL SIGN PRISHTHAMATRA E..AW
0951..0957    ; XID_Continue # Mn   [7] DEVANAGARI STRESS SIGN UDATTA..
                                        DEVANAGARI VOWEL SIGN UUE
0962..0963    ; XID_Continue # Mn   [2] DEVANAGARI VOWEL SIGN VOCALIC L..LL
0981          ; XID_Continue # Mn       BENGALI SIGN CANDRABINDU
0982..0983    ; XID_Continue # Mc   [2] BENGALI SIGN ANUSVARA..VISARGA
09BC          ; XID_Continue # Mn       BENGALI SIGN NUKTA
09BE..09C0    ; XID_Continue # Mc   [3] BENGALI VOWEL SIGN AA..II
09C1..09C4    ; XID_Continue # Mn   [4] BENGALI VOWEL SIGN U..VOCALIC RR
09C7..09C8    ; XID_Continue # Mc   [2] BENGALI VOWEL SIGN E..AI
09CB..09CC    ; XID_Continue # Mc   [2] BENGALI VOWEL SIGN O..AU
09CD          ; XID_Continue # Mn       BENGALI SIGN VIRAMA
09D7          ; XID_Continue # Mc       BENGALI AU LENGTH MARK
09E2..09E3    ; XID_Continue # Mn   [2] BENGALI VOWEL SIGN VOCALIC L..LL
09FE          ; XID_Continue # Mn       BENGALI SANDHI MARK
0A01..0A02    ; XID_Continue # Mn   [2] GURMUKHI SIGN ADAK BINDI..BINDI
0A03          ; XID_Continue # Mc       GURMUKHI SIGN VISARGA
0A3C          ; XID_Continue # Mn       GURMUKHI SIGN NUKTA
```

```
0A3E..0A40    ; XID_Continue # Mc   [3] GURMUKHI VOWEL SIGN AA..II
0A41..0A42    ; XID_Continue # Mn   [2] GURMUKHI VOWEL SIGN U..UU
0A47..0A48    ; XID_Continue # Mn   [2] GURMUKHI VOWEL SIGN EE..AI
0A4B..0A4D    ; XID_Continue # Mn   [3] GURMUKHI VOWEL SIGN OO..
                                        GURMUKHI SIGN VIRAMA
0A51          ; XID_Continue # Mn       GURMUKHI SIGN UDAAT
0A70..0A71    ; XID_Continue # Mn   [2] GURMUKHI TIPPI..GURMUKHI ADDAK
0A75          ; XID_Continue # Mn       GURMUKHI SIGN YAKASH
0A81..0A82    ; XID_Continue # Mn   [2] GUJARATI SIGN CANDRABINDU..
                                        GUJARATI SIGN ANUSVARA
0A83          ; XID_Continue # Mc       GUJARATI SIGN VISARGA
0ABC          ; XID_Continue # Mn       GUJARATI SIGN NUKTA
0ABE..0AC0    ; XID_Continue # Mc   [3] GUJARATI VOWEL SIGN AA..II
0AC1..0AC5    ; XID_Continue # Mn   [5] GUJARATI VOWEL SIGN U..CANDRA E
0AC7..0AC8    ; XID_Continue # Mn   [2] GUJARATI VOWEL SIGN E..AI
0AC9          ; XID_Continue # Mc       GUJARATI VOWEL SIGN CANDRA O
0ACB..0ACC    ; XID_Continue # Mc   [2] GUJARATI VOWEL SIGN O..AU
0ACD          ; XID_Continue # Mn       GUJARATI SIGN VIRAMA
0AE2..0AE3    ; XID_Continue # Mn   [2] GUJARATI VOWEL SIGN VOCALIC L..LL
0AFA..0AFF    ; XID_Continue # Mn   [6] GUJARATI SIGN SUKUN..
                                        GUJARATI SIGN TWO-CIRCLE NUKTA ABOVE
0B01          ; XID_Continue # Mn       ORIYA SIGN CANDRABINDU
0B02..0B03    ; XID_Continue # Mc   [2] ORIYA SIGN ANUSVARA..
                                        ORIYA SIGN VISARGA
0B3C          ; XID_Continue # Mn       ORIYA SIGN NUKTA
0B3E          ; XID_Continue # Mc       ORIYA VOWEL SIGN AA
0B3F          ; XID_Continue # Mn       ORIYA VOWEL SIGN I
0B40          ; XID_Continue # Mc       ORIYA VOWEL SIGN II
0B41..0B44    ; XID_Continue # Mn   [4] ORIYA VOWEL SIGN U..VOCALIC RR
0B47..0B48    ; XID_Continue # Mc   [2] ORIYA VOWEL SIGN E..AI
0B4B..0B4C    ; XID_Continue # Mc   [2] ORIYA VOWEL SIGN O..AU
0B4D          ; XID_Continue # Mn       ORIYA SIGN VIRAMA
0B55..0B56    ; XID_Continue # Mn   [2] ORIYA SIGN OVERLINE..
                                        ORIYA AI LENGTH MARK
0B57          ; XID_Continue # Mc       ORIYA AU LENGTH MARK
0B62..0B63    ; XID_Continue # Mn   [2] ORIYA VOWEL SIGN VOCALIC L..LL
0B82          ; XID_Continue # Mn       TAMIL SIGN ANUSVARA
0BBE..0BBF    ; XID_Continue # Mc   [2] TAMIL VOWEL SIGN AA..I
0BC0          ; XID_Continue # Mn       TAMIL VOWEL SIGN II
0BC1..0BC2    ; XID_Continue # Mc   [2] TAMIL VOWEL SIGN U..UU
0BC6..0BC8    ; XID_Continue # Mc   [3] TAMIL VOWEL SIGN E..AI
0BCA..0BCC    ; XID_Continue # Mc   [3] TAMIL VOWEL SIGN O..AU
0BCD          ; XID_Continue # Mn       TAMIL SIGN VIRAMA
0BD7          ; XID_Continue # Mc       TAMIL AU LENGTH MARK
0C00          ; XID_Continue # Mn       TELUGU SIGN COMBINING CANDRABINDU ABOVE
0C01..0C03    ; XID_Continue # Mc   [3] TELUGU SIGN CANDRABINDU..VISARGA
```

```
0C04          ; XID_Continue # Mn       TELUGU SIGN COMBINING ANUSVARA ABOVE
0C3C          ; XID_Continue # Mn       TELUGU SIGN NUKTA
0C3E..0C40    ; XID_Continue # Mn   [3] TELUGU VOWEL SIGN AA..II
0C41..0C44    ; XID_Continue # Mc   [4] TELUGU VOWEL SIGN U..VOCALIC RR
0C46..0C48    ; XID_Continue # Mn   [3] TELUGU VOWEL SIGN E..AI
0C4A..0C4D    ; XID_Continue # Mn   [4] TELUGU VOWEL SIGN O..SIGN VIRAMA
0C55..0C56    ; XID_Continue # Mn   [2] TELUGU LENGTH MARK..AI LENGTH MARK
0C62..0C63    ; XID_Continue # Mn   [2] TELUGU VOWEL SIGN VOCALIC L..LL
0C81          ; XID_Continue # Mn       KANNADA SIGN CANDRABINDU
0C82..0C83    ; XID_Continue # Mc   [2] KANNADA SIGN ANUSVARA..VISARGA
0CBC          ; XID_Continue # Mn       KANNADA SIGN NUKTA
0CBE          ; XID_Continue # Mc       KANNADA VOWEL SIGN AA
0CBF          ; XID_Continue # Mn       KANNADA VOWEL SIGN I
0CC0..0CC4    ; XID_Continue # Mc   [5] KANNADA VOWEL SIGN II..VOCALIC RR
0CC6          ; XID_Continue # Mn       KANNADA VOWEL SIGN E
0CC7..0CC8    ; XID_Continue # Mc   [2] KANNADA VOWEL SIGN EE..AI
0CCA..0CCB    ; XID_Continue # Mc   [2] KANNADA VOWEL SIGN O..OO
0CCC..0CCD    ; XID_Continue # Mn   [2] KANNADA VOWEL SIGN AU..VIRAMA
0CD5..0CD6    ; XID_Continue # Mc   [2] KANNADA LENGTH MARK..AI LENGTH MARK
0CE2..0CE3    ; XID_Continue # Mn   [2] KANNADA VOWEL SIGN VOCALIC L..LL
0D00..0D01    ; XID_Continue # Mn   [2] MALAYALAM SIGN COMBINING ANUSVARA ABOVE..
                                         CANDRABINDU
0D02..0D03    ; XID_Continue # Mc   [2] MALAYALAM SIGN ANUSVARA..VISARGA
0D3B..0D3C    ; XID_Continue # Mn   [2] MALAYALAM SIGN VERTICAL BAR VIRAMA..
                                         CIRCULAR VIRAMA
0D3E..0D40    ; XID_Continue # Mc   [3] MALAYALAM VOWEL SIGN AA..II
0D41..0D44    ; XID_Continue # Mn   [4] MALAYALAM VOWEL SIGN U..VOCALIC RR
0D46..0D48    ; XID_Continue # Mc   [3] MALAYALAM VOWEL SIGN E..AI
0D4A..0D4C    ; XID_Continue # Mc   [3] MALAYALAM VOWEL SIGN O..AU
0D4D          ; XID_Continue # Mn       MALAYALAM SIGN VIRAMA
0D57          ; XID_Continue # Mc       MALAYALAM AU LENGTH MARK
0D62..0D63    ; XID_Continue # Mn   [2] MALAYALAM VOWEL SIGN VOCALIC L..LL
0D81          ; XID_Continue # Mn       SINHALA SIGN CANDRABINDU
0D82..0D83    ; XID_Continue # Mc   [2] SINHALA SIGN ANUSVARAYA..VISARGAYA
0DCA          ; XID_Continue # Mn       SINHALA SIGN AL-LAKUNA
0DCF..0DD1    ; XID_Continue # Mc   [3] SINHALA VOWEL SIGN AELA-PILLA..
                                         DIGA AEDA-PILLA
0DD2..0DD4    ; XID_Continue # Mn   [3] SINHALA VOWEL SIGN KETTI IS-PILLA..
                                         PAA-PILLA
0DD6          ; XID_Continue # Mn       SINHALA VOWEL SIGN DIGA PAA-PILLA
0DD8..0DDF    ; XID_Continue # Mc   [8] SINHALA VOWEL SIGN GAETTA-PILLA..
                                         GAYANUKITTA
0DF2..0DF3    ; XID_Continue # Mc   [2] SINHALA VOWEL SIGN DIGA GAETTA-PILLA..
                                         GAYANUKITTA
0E31          ; XID_Continue # Mn       THAI CHARACTER MAI HAN-AKAT
0E34..0E3A    ; XID_Continue # Mn   [7] THAI CHARACTER SARA I..PHINTHU
```

```
0E47..0E4E    ; XID_Continue # Mn   [8] THAI CHARACTER MAITAIKHU..YAMAKKAN
0EB1          ; XID_Continue # Mn       LAO VOWEL SIGN MAI KAN
0EB4..0EBC    ; XID_Continue # Mn   [9] LAO VOWEL SIGN I..SEMIVOWEL SIGN LO
0EC8..0ECD    ; XID_Continue # Mn   [6] LAO TONE MAI EK..NIGGAHITA
0F18..0F19    ; XID_Continue # Mn   [2] TIBETAN ASTROLOGICAL SIGN -KHYUD PA..
                                         SDONG TSHUGS
0F35          ; XID_Continue # Mn       TIBETAN MARK NGAS BZUNG NYI ZLA
0F37          ; XID_Continue # Mn       TIBETAN MARK NGAS BZUNG SGOR RTAGS
0F39          ; XID_Continue # Mn       TIBETAN MARK TSA -PHRU
0F3E..0F3F    ; XID_Continue # Mc   [2] TIBETAN SIGN YAR TSHES..MAR TSHES
0F71..0F7E    ; XID_Continue # Mn  [14] TIBETAN VOWEL SIGN AA..RJES SU NGA RO
0F7F          ; XID_Continue # Mc       TIBETAN SIGN RNAM BCAD
0F80..0F84    ; XID_Continue # Mn   [5] TIBETAN VOWEL SIGN REVERSED I..
                                         MARK HALANTA
0F86..0F87    ; XID_Continue # Mn   [2] TIBETAN SIGN LCI RTAGS..YANG RTAGS
0F8D..0F97    ; XID_Continue # Mn  [11] TIBETAN SUBJOINED SIGN LCE TSA CAN..
                                         LETTER JA
0F99..0FBC    ; XID_Continue # Mn  [36] TIBETAN SUBJOINED LETTER NYA..
                                         FIXED-FORM RA
0FC6          ; XID_Continue # Mn       TIBETAN SYMBOL PADMA GDAN
102B..102C    ; XID_Continue # Mc   [2] MYANMAR VOWEL SIGN TALL AA..AA
102D..1030    ; XID_Continue # Mn   [4] MYANMAR VOWEL SIGN I..UU
1031          ; XID_Continue # Mc       MYANMAR VOWEL SIGN E
1032..1037    ; XID_Continue # Mn   [6] MYANMAR VOWEL SIGN AI..DOT BELOW
1038          ; XID_Continue # Mc       MYANMAR SIGN VISARGA
1039..103A    ; XID_Continue # Mn   [2] MYANMAR SIGN VIRAMA..ASAT
103B..103C    ; XID_Continue # Mc   [2] MYANMAR CONSONANT SIGN MEDIAL YA..RA
103D..103E    ; XID_Continue # Mn   [2] MYANMAR CONSONANT SIGN MEDIAL WA..HA
1056..1057    ; XID_Continue # Mc   [2] MYANMAR VOWEL SIGN VOCALIC R..RR
1058..1059    ; XID_Continue # Mn   [2] MYANMAR VOWEL SIGN VOCALIC L..LL
105E..1060    ; XID_Continue # Mn   [3] MYANMAR CONSONANT SIGN MON MEDIAL NA..LA
1062..1064    ; XID_Continue # Mc   [3] MYANMAR VOWEL SIGN SGAW KAREN EU..KE PHO
1067..106D    ; XID_Continue # Mc   [7] MYANMAR VOWEL SIGN WESTERN PWO KAREN EU..
                                         TONE-5
1071..1074    ; XID_Continue # Mn   [4] MYANMAR VOWEL SIGN GEBA KAREN I..KAYAH EE
1082          ; XID_Continue # Mn       MYANMAR CONSONANT SIGN SHAN MEDIAL WA
1083..1084    ; XID_Continue # Mc   [2] MYANMAR VOWEL SIGN SHAN AA..E
1085..1086    ; XID_Continue # Mn   [2] MYANMAR VOWEL SIGN SHAN E ABOVE..FINAL Y
1087..108C    ; XID_Continue # Mc   [6] MYANMAR SIGN SHAN TONE-2..TONE-3
108D          ; XID_Continue # Mn       MYANMAR SIGN SHAN COUNCIL EMPHATIC TONE
108F          ; XID_Continue # Mc       MYANMAR SIGN RUMAI PALAUNG TONE-5
109A..109C    ; XID_Continue # Mc   [3] MYANMAR SIGN KHAMTI TONE-1..AITON A
109D          ; XID_Continue # Mn       MYANMAR VOWEL SIGN AITON AI
135D..135F    ; XID_Continue # Mn   [3] ETHIOPIC COMBINING GEMINATION AND
                                         VOWEL LENGTH MARK..MARK
1712..1714    ; XID_Continue # Mn   [3] TAGALOG VOWEL SIGN I..VIRAMA
```

```
1715          ; XID_Continue # Mc       TAGALOG SIGN PAMUDPOD
1732..1733    ; XID_Continue # Mn   [2] HANUNOO VOWEL SIGN I..U
1734          ; XID_Continue # Mc       HANUNOO SIGN PAMUDPOD
1752..1753    ; XID_Continue # Mn   [2] BUHID VOWEL SIGN I..U
1772..1773    ; XID_Continue # Mn   [2] TAGBANWA VOWEL SIGN I..U
17B4..17B5    ; XID_Continue # Mn   [2] KHMER VOWEL INHERENT AQ..AA
17B6          ; XID_Continue # Mc       KHMER VOWEL SIGN AA
17B7..17BD    ; XID_Continue # Mn   [7] KHMER VOWEL SIGN I..UA
17BE..17C5    ; XID_Continue # Mc   [8] KHMER VOWEL SIGN OE..AU
17C6          ; XID_Continue # Mn       KHMER SIGN NIKAHIT
17C7..17C8    ; XID_Continue # Mc   [2] KHMER SIGN REAHMUK..YUUKALEAPINTU
17C9..17D3    ; XID_Continue # Mn  [11] KHMER SIGN MUUSIKATOAN..BATHAMASAT
17DD          ; XID_Continue # Mn       KHMER SIGN ATTHACAN
180B..180D    ; XID_Continue # Mn   [3] MONGOLIAN FREE VARIATION SELECTOR ONE..
                                        THREE
180F          ; XID_Continue # Mn       MONGOLIAN FREE VARIATION SELECTOR FOUR
1885..1886    ; XID_Continue # Mn   [2] MONGOLIAN LETTER ALI GALI BALUDA..
                                        THREE BALUDA
18A9          ; XID_Continue # Mn       MONGOLIAN LETTER ALI GALI DAGALGA
1920..1922    ; XID_Continue # Mn   [3] LIMBU VOWEL SIGN A..U
1923..1926    ; XID_Continue # Mc   [4] LIMBU VOWEL SIGN EE..AU
1927..1928    ; XID_Continue # Mn   [2] LIMBU VOWEL SIGN E..O
1929..192B    ; XID_Continue # Mc   [3] LIMBU SUBJOINED LETTER YA..WA
1930..1931    ; XID_Continue # Mc   [2] LIMBU SMALL LETTER KA..NGA
1932          ; XID_Continue # Mn       LIMBU SMALL LETTER ANUSVARA
1933..1938    ; XID_Continue # Mc   [6] LIMBU SMALL LETTER TA..LA
1939..193B    ; XID_Continue # Mn   [3] LIMBU SIGN MUKPHRENG..-I
1A17..1A18    ; XID_Continue # Mn   [2] BUGINESE VOWEL SIGN I..U
1A19..1A1A    ; XID_Continue # Mc   [2] BUGINESE VOWEL SIGN E..O
1A1B          ; XID_Continue # Mn       BUGINESE VOWEL SIGN AE
1A55          ; XID_Continue # Mc       TAI THAM CONSONANT SIGN MEDIAL RA
1A56          ; XID_Continue # Mn       TAI THAM CONSONANT SIGN MEDIAL LA
1A57          ; XID_Continue # Mc       TAI THAM CONSONANT SIGN LA TANG LAI
1A58..1A5E    ; XID_Continue # Mn   [7] TAI THAM SIGN MAI KANG LAI..
                                        CONSONANT SIGN SA
1A60          ; XID_Continue # Mn       TAI THAM SIGN SAKOT
1A61          ; XID_Continue # Mc       TAI THAM VOWEL SIGN A
1A62          ; XID_Continue # Mn       TAI THAM VOWEL SIGN MAI SAT
1A63..1A64    ; XID_Continue # Mc   [2] TAI THAM VOWEL SIGN AA..TALL AA
1A65..1A6C    ; XID_Continue # Mn   [8] TAI THAM VOWEL SIGN I..OA BELOW
1A6D..1A72    ; XID_Continue # Mc   [6] TAI THAM VOWEL SIGN OY..THAM AI
1A73..1A7C    ; XID_Continue # Mn  [10] TAI THAM VOWEL SIGN OA ABOVE..
                                        KHUEN-LUE KARAN
1A7F          ; XID_Continue # Mn       TAI THAM COMBINING CRYPTOGRAMMIC DOT
1AB0..1ABD    ; XID_Continue # Mn  [14] COMBINING DOUBLED CIRCUMFLEX ACCENT..
                                        COMBINING PARENTHESES BELOW
```

```
1ABF..1ACE    ; XID_Continue # Mn  [16] COMBINING LATIN SMALL LETTER W BELOW..
                                       INSULAR T
1B00..1B03    ; XID_Continue # Mn   [4] BALINESE SIGN ULU RICEM..SURANG
1B04          ; XID_Continue # Mc       BALINESE SIGN BISAH
1B34          ; XID_Continue # Mn       BALINESE SIGN REREKAN
1B35          ; XID_Continue # Mc       BALINESE VOWEL SIGN TEDUNG
1B36..1B3A    ; XID_Continue # Mn   [5] BALINESE VOWEL SIGN ULU..RA REPA
1B3B          ; XID_Continue # Mc       BALINESE VOWEL SIGN RA REPA TEDUNG
1B3C          ; XID_Continue # Mn       BALINESE VOWEL SIGN LA LENGA
1B3D..1B41    ; XID_Continue # Mc   [5] BALINESE VOWEL SIGN LA LENGA TEDUNG..
                                       TALING REPA TEDUNG
1B42          ; XID_Continue # Mn       BALINESE VOWEL SIGN PEPET
1B43..1B44    ; XID_Continue # Mc   [2] BALINESE VOWEL SIGN PEPET TEDUNG..
                                       BALINESE ADEG ADEG
1B6B..1B73    ; XID_Continue # Mn   [9] BALINESE MUSICAL SYMBOL COMBINING TEGEH..
                                       GONG
1B80..1B81    ; XID_Continue # Mn   [2] SUNDANESE SIGN PANYECEK..PANGLAYAR
1B82          ; XID_Continue # Mc       SUNDANESE SIGN PANGWISAD
1BA1          ; XID_Continue # Mc       SUNDANESE CONSONANT SIGN PAMINGKAL
1BA2..1BA5    ; XID_Continue # Mn   [4] SUNDANESE CONSONANT SIGN PANYAKRA..
                                       SUNDANESE VOWEL SIGN PANYUKU
1BA6..1BA7    ; XID_Continue # Mc   [2] SUNDANESE VOWEL SIGN PANAELAENG..PANOLONG
1BA8..1BA9    ; XID_Continue # Mn   [2] SUNDANESE VOWEL SIGN PAMEPET..PANEULEUNG
1BAA          ; XID_Continue # Mc       SUNDANESE SIGN PAMAAEH
1BAB..1BAD    ; XID_Continue # Mn   [3] SUNDANESE SIGN VIRAMA..
                                       CONSONANT SIGN PASANGAN WA
1BE6          ; XID_Continue # Mn       BATAK SIGN TOMPI
1BE7          ; XID_Continue # Mc       BATAK VOWEL SIGN E
1BE8..1BE9    ; XID_Continue # Mn   [2] BATAK VOWEL SIGN PAKPAK E..EE
1BEA..1BEC    ; XID_Continue # Mc   [3] BATAK VOWEL SIGN I..O
1BED          ; XID_Continue # Mn       BATAK VOWEL SIGN KARO O
1BEE          ; XID_Continue # Mc       BATAK VOWEL SIGN U
1BEF..1BF1    ; XID_Continue # Mn   [3] BATAK VOWEL SIGN U FOR SIMALUNGUN SA..
                                       BATAK CONSONANT SIGN H
1BF2..1BF3    ; XID_Continue # Mc   [2] BATAK PANGOLAT..BATAK PANONGONAN
1C24..1C2B    ; XID_Continue # Mc   [8] LEPCHA SUBJOINED LETTER YA..VOWEL SIGN UU
1C2C..1C33    ; XID_Continue # Mn   [8] LEPCHA VOWEL SIGN E..CONSONANT SIGN T
1C34..1C35    ; XID_Continue # Mc   [2] LEPCHA CONSONANT SIGN NYIN-DO..KANG
1C36..1C37    ; XID_Continue # Mn   [2] LEPCHA SIGN RAN..NUKTA
1CD0..1CD2    ; XID_Continue # Mn   [3] VEDIC TONE KARSHANA..PRENKHA
1CD4..1CE0    ; XID_Continue # Mn  [13] VEDIC SIGN YAJURVEDIC MIDLINE SVARITA..
                                       VEDIC TONE RIGVEDIC KASHMIRI INDEPENDENT
                                       SVARITA
1CE1          ; XID_Continue # Mc       VEDIC TONE ATHARVAVEDIC INDEPENDENT
                                       SVARITA
1CE2..1CE8    ; XID_Continue # Mn   [7] VEDIC SIGN VISARGA SVARITA..
```

```
                                          VEDIC SIGN VISARGA ANUDATTA WITH TAIL
1CED          ; XID_Continue # Mn         VEDIC SIGN TIRYAK
1CF4          ; XID_Continue # Mn         VEDIC TONE CANDRA ABOVE
1CF7          ; XID_Continue # Mc         VEDIC SIGN ATIKRAMA
1CF8..1CF9    ; XID_Continue # Mn   [2]   VEDIC TONE RING ABOVE..DOUBLE RING ABOVE
1DC0..1DFF    ; XID_Continue # Mn  [64]   COMBINING DOTTED GRAVE ACCENT..
                                          RIGHT ARROWHEAD AND DOWN ARROWHEAD BELOW
20D0..20DC    ; XID_Continue # Mn  [13]   COMBINING LEFT HARPOON ABOVE..
                                          COMBINING FOUR DOTS ABOVE
20E1          ; XID_Continue # Mn         COMBINING LEFT RIGHT ARROW ABOVE
20E5..20F0    ; XID_Continue # Mn  [12]   COMBINING REVERSE SOLIDUS OVERLAY..
                                          COMBINING ASTERISK ABOVE
2CEF..2CF1    ; XID_Continue # Mn   [3]   COPTIC COMBINING NI ABOVE..SPIRITUS LENIS
2D7F          ; XID_Continue # Mn         TIFINAGH CONSONANT JOINER
2DE0..2DFF    ; XID_Continue # Mn  [32]   COMBINING CYRILLIC LETTER BE..
                                          IOTIFIED BIG YUS
302A..302D    ; XID_Continue # Mn   [4]   IDEOGRAPHIC LEVEL TONE MARK..
                                          IDEOGRAPHIC ENTERING TONE MARK
302E..302F    ; XID_Continue # Mc   [2]   HANGUL SINGLE DOT TONE MARK..
                                          HANGUL DOUBLE DOT TONE MARK
3099..309A    ; XID_Continue # Mn   [2]   COMBINING KATAKANA-HIRAGANA VOICED
                                          SOUND MARK..SEMI-VOICED SOUND MARK
A66F          ; XID_Continue # Mn         COMBINING CYRILLIC VZMET
A674..A67D    ; XID_Continue # Mn  [10]   COMBINING CYRILLIC LETTER UKRAINIAN IE..
                                          PAYEROK
A69E..A69F    ; XID_Continue # Mn   [2]   COMBINING CYRILLIC LETTER EF..IOTIFIED E
A6F0..A6F1    ; XID_Continue # Mn   [2]   BAMUM COMBINING MARK KOQNDON..TUKWENTIS
A802          ; XID_Continue # Mn         SYLOTI NAGRI SIGN DVISVARA
A806          ; XID_Continue # Mn         SYLOTI NAGRI SIGN HASANTA
A80B          ; XID_Continue # Mn         SYLOTI NAGRI SIGN ANUSVARA
A823..A824    ; XID_Continue # Mc   [2]   SYLOTI NAGRI VOWEL SIGN A..I
A825..A826    ; XID_Continue # Mn   [2]   SYLOTI NAGRI VOWEL SIGN U..E
A827          ; XID_Continue # Mc         SYLOTI NAGRI VOWEL SIGN OO
A82C          ; XID_Continue # Mn         SYLOTI NAGRI SIGN ALTERNATE HASANTA
A880..A881    ; XID_Continue # Mc   [2]   SAURASHTRA SIGN ANUSVARA..VISARGA
A8B4..A8C3    ; XID_Continue # Mc  [16]   SAURASHTRA CONSONANT SIGN HAARU..
                                          SAURASHTRA VOWEL SIGN AU
A8C4..A8C5    ; XID_Continue # Mn   [2]   SAURASHTRA SIGN VIRAMA..CANDRABINDU
A8E0..A8F1    ; XID_Continue # Mn  [18]   COMBINING DEVANAGARI DIGIT ZERO..
                                          SIGN AVAGRAHA
A8FF          ; XID_Continue # Mn         DEVANAGARI VOWEL SIGN AY
A926..A92D    ; XID_Continue # Mn   [8]   KAYAH LI VOWEL UE..TONE CALYA PLOPHU
A947..A951    ; XID_Continue # Mn  [11]   REJANG VOWEL SIGN I..CONSONANT SIGN R
A952..A953    ; XID_Continue # Mc   [2]   REJANG CONSONANT SIGN H..REJANG VIRAMA
A980..A982    ; XID_Continue # Mn   [3]   JAVANESE SIGN PANYANGGA..LAYAR
A983          ; XID_Continue # Mc         JAVANESE SIGN WIGNYAN
```

```
A9B3          ; XID_Continue # Mn       JAVANESE SIGN CECAK TELU
A9B4..A9B5    ; XID_Continue # Mc   [2] JAVANESE VOWEL SIGN TARUNG..TOLONG
A9B6..A9B9    ; XID_Continue # Mn   [4] JAVANESE VOWEL SIGN WULU..SUKU MENDUT
A9BA..A9BB    ; XID_Continue # Mc   [2] JAVANESE VOWEL SIGN TALING..DIRGA MURE
A9BC..A9BD    ; XID_Continue # Mn   [2] JAVANESE VOWEL SIGN PEPET..KERET
A9BE..A9C0    ; XID_Continue # Mc   [3] JAVANESE CONSONANT SIGN PENGKAL..PANGKON
A9E5          ; XID_Continue # Mn       MYANMAR SIGN SHAN SAW
AA29..AA2E    ; XID_Continue # Mn   [6] CHAM VOWEL SIGN AA..OE
AA2F..AA30    ; XID_Continue # Mc   [2] CHAM VOWEL SIGN O..AI
AA31..AA32    ; XID_Continue # Mn   [2] CHAM VOWEL SIGN AU..UE
AA33..AA34    ; XID_Continue # Mc   [2] CHAM CONSONANT SIGN YA..RA
AA35..AA36    ; XID_Continue # Mn   [2] CHAM CONSONANT SIGN LA..WA
AA43          ; XID_Continue # Mn       CHAM CONSONANT SIGN FINAL NG
AA4C          ; XID_Continue # Mn       CHAM CONSONANT SIGN FINAL M
AA4D          ; XID_Continue # Mc       CHAM CONSONANT SIGN FINAL H
AA7B          ; XID_Continue # Mc       MYANMAR SIGN PAO KAREN TONE
AA7C          ; XID_Continue # Mn       MYANMAR SIGN TAI LAING TONE-2
AA7D          ; XID_Continue # Mc       MYANMAR SIGN TAI LAING TONE-5
AAB0          ; XID_Continue # Mn       TAI VIET MAI KANG
AAB2..AAB4    ; XID_Continue # Mn   [3] TAI VIET VOWEL I..U
AAB7..AAB8    ; XID_Continue # Mn   [2] TAI VIET MAI KHIT..VOWEL IA
AABE..AABF    ; XID_Continue # Mn   [2] TAI VIET VOWEL AM..TONE MAI EK
AAC1          ; XID_Continue # Mn       TAI VIET TONE MAI THO
AAEB          ; XID_Continue # Mc       MEETEI MAYEK VOWEL SIGN II
AAEC..AAED    ; XID_Continue # Mn   [2] MEETEI MAYEK VOWEL SIGN UU..AAI
AAEE..AAEF    ; XID_Continue # Mc   [2] MEETEI MAYEK VOWEL SIGN AU..AAU
AAF5          ; XID_Continue # Mc       MEETEI MAYEK VOWEL SIGN VISARGA
AAF6          ; XID_Continue # Mn       MEETEI MAYEK VIRAMA
ABE3..ABE4    ; XID_Continue # Mc   [2] MEETEI MAYEK VOWEL SIGN ONAP..INAP
ABE5          ; XID_Continue # Mn       MEETEI MAYEK VOWEL SIGN ANAP
ABE6..ABE7    ; XID_Continue # Mc   [2] MEETEI MAYEK VOWEL SIGN YENAP..SOUNAP
ABE8          ; XID_Continue # Mn       MEETEI MAYEK VOWEL SIGN UNAP
ABE9..ABEA    ; XID_Continue # Mc   [2] MEETEI MAYEK VOWEL SIGN CHEINAP..NUNG
ABEC          ; XID_Continue # Mc       MEETEI MAYEK LUM IYEK
ABED          ; XID_Continue # Mn       MEETEI MAYEK APUN IYEK
FB1E          ; XID_Continue # Mn       HEBREW POINT JUDEO-SPANISH VARIKA
FE00..FE0F    ; XID_Continue # Mn  [16] VARIATION SELECTOR-1..-16
FE20..FE2F    ; XID_Continue # Mn  [16] COMBINING LIGATURE LEFT HALF..
                                        COMBINING CYRILLIC TITLO RIGHT HALF
101FD         ; XID_Continue # Mn       PHAISTOS DISC SIGN COMBINING OBLIQUE
                                        STROKE
102E0         ; XID_Continue # Mn       COPTIC EPACT THOUSANDS MARK
10376..1037A  ; XID_Continue # Mn   [5] COMBINING OLD PERMIC LETTER AN..SII
10A01..10A03  ; XID_Continue # Mn   [3] KHAROSHTHI VOWEL SIGN I..VOCALIC R
10A05..10A06  ; XID_Continue # Mn   [2] KHAROSHTHI VOWEL SIGN E..O
10A0C..10A0F  ; XID_Continue # Mn   [4] KHAROSHTHI VOWEL LENGTH MARK..
```

```
                                          SIGN VISARGA
10A38..10A3A  ; XID_Continue # Mn   [3] KHAROSHTHI SIGN BAR ABOVE..DOT BELOW
10A3F         ; XID_Continue # Mn       KHAROSHTHI VIRAMA
10AE5..10AE6  ; XID_Continue # Mn   [2] MANICHAEAN ABBREVIATION MARK ABOVE..BELOW
10D24..10D27  ; XID_Continue # Mn   [4] HANIFI ROHINGYA SIGN HARBAHAY..TASSI
10EAB..10EAC  ; XID_Continue # Mn   [2] YEZIDI COMBINING HAMZA MARK..MADDA MARK
10F46..10F50  ; XID_Continue # Mn  [11] SOGDIAN COMBINING DOT BELOW..STROKE BELOW
10F82..10F85  ; XID_Continue # Mn   [4] OLD UYGHUR COMBINING DOT ABOVE..
                                          TWO DOTS BELOW
11000         ; XID_Continue # Mc       BRAHMI SIGN CANDRABINDU
11001         ; XID_Continue # Mn       BRAHMI SIGN ANUSVARA
11002         ; XID_Continue # Mc       BRAHMI SIGN VISARGA
11038..11046  ; XID_Continue # Mn  [15] BRAHMI VOWEL SIGN AA..BRAHMI VIRAMA
11070         ; XID_Continue # Mn       BRAHMI SIGN OLD TAMIL VIRAMA
11073..11074  ; XID_Continue # Mn   [2] BRAHMI VOWEL SIGN OLD TAMIL SHORT E..O
1107F..11081  ; XID_Continue # Mn   [3] BRAHMI NUMBER JOINER..SIGN ANUSVARA
11082         ; XID_Continue # Mc       KAITHI SIGN VISARGA
110B0..110B2  ; XID_Continue # Mc   [3] KAITHI VOWEL SIGN AA..II
110B3..110B6  ; XID_Continue # Mn   [4] KAITHI VOWEL SIGN U..AI
110B7..110B8  ; XID_Continue # Mc   [2] KAITHI VOWEL SIGN O..AU
110B9..110BA  ; XID_Continue # Mn   [2] KAITHI SIGN VIRAMA..KAITHI SIGN NUKTA
110C2         ; XID_Continue # Mn       KAITHI VOWEL SIGN VOCALIC R
11100..11102  ; XID_Continue # Mn   [3] CHAKMA SIGN CANDRABINDU..VISARGA
11127..1112B  ; XID_Continue # Mn   [5] CHAKMA VOWEL SIGN A..UU
1112C         ; XID_Continue # Mc       CHAKMA VOWEL SIGN E
1112D..11134  ; XID_Continue # Mn   [8] CHAKMA VOWEL SIGN AI..CHAKMA MAAYYAA
11145..11146  ; XID_Continue # Mc   [2] CHAKMA VOWEL SIGN AA..EI
11173         ; XID_Continue # Mn       MAHAJANI SIGN NUKTA
11180..11181  ; XID_Continue # Mn   [2] SHARADA SIGN CANDRABINDU..ANUSVARA
11182         ; XID_Continue # Mc       SHARADA SIGN VISARGA
111B3..111B5  ; XID_Continue # Mc   [3] SHARADA VOWEL SIGN AA..II
111B6..111BE  ; XID_Continue # Mn   [9] SHARADA VOWEL SIGN U..O
111BF..111C0  ; XID_Continue # Mc   [2] SHARADA VOWEL SIGN AU..VIRAMA
111C9..111CC  ; XID_Continue # Mn   [4] SHARADA SANDHI MARK..
                                          EXTRA SHORT VOWEL MARK
111CE         ; XID_Continue # Mc       SHARADA VOWEL SIGN PRISHTHAMATRA E
111CF         ; XID_Continue # Mn       SHARADA SIGN INVERTED CANDRABINDU
1122C..1122E  ; XID_Continue # Mc   [3] KHOJKI VOWEL SIGN AA..II
1122F..11231  ; XID_Continue # Mn   [3] KHOJKI VOWEL SIGN U..AI
11232..11233  ; XID_Continue # Mc   [2] KHOJKI VOWEL SIGN O..AU
11234         ; XID_Continue # Mn       KHOJKI SIGN ANUSVARA
11235         ; XID_Continue # Mc       KHOJKI SIGN VIRAMA
11236..11237  ; XID_Continue # Mn   [2] KHOJKI SIGN NUKTA..SHADDA
1123E         ; XID_Continue # Mn       KHOJKI SIGN SUKUN
112DF         ; XID_Continue # Mn       KHUDAWADI SIGN ANUSVARA
112E0..112E2  ; XID_Continue # Mc   [3] KHUDAWADI VOWEL SIGN AA..II
```

```
112E3..112EA  ; XID_Continue # Mn   [8] KHUDAWADI VOWEL SIGN U..VIRAMA
11300..11301  ; XID_Continue # Mn   [2] GRANTHA SIGN COMBINING ANUSVARA ABOVE..
                                        GRANTHA SIGN CANDRABINDU
11302..11303  ; XID_Continue # Mc   [2] GRANTHA SIGN ANUSVARA..VISARGA
1133B..1133C  ; XID_Continue # Mn   [2] COMBINING BINDU BELOW..GRANTHA SIGN NUKTA
1133E..1133F  ; XID_Continue # Mc   [2] GRANTHA VOWEL SIGN AA..I
11340         ; XID_Continue # Mn       GRANTHA VOWEL SIGN II
11341..11344  ; XID_Continue # Mc   [4] GRANTHA VOWEL SIGN U..VOCALIC RR
11347..11348  ; XID_Continue # Mc   [2] GRANTHA VOWEL SIGN EE..AI
1134B..1134D  ; XID_Continue # Mc   [3] GRANTHA VOWEL SIGN OO..VIRAMA
11357         ; XID_Continue # Mc       GRANTHA AU LENGTH MARK
11362..11363  ; XID_Continue # Mc   [2] GRANTHA VOWEL SIGN VOCALIC L..LL
11366..1136C  ; XID_Continue # Mn   [7] COMBINING GRANTHA DIGIT ZERO..SIX
11370..11374  ; XID_Continue # Mn   [5] COMBINING GRANTHA LETTER A..PA
11435..11437  ; XID_Continue # Mc   [3] NEWA VOWEL SIGN AA..II
11438..1143F  ; XID_Continue # Mn   [8] NEWA VOWEL SIGN U..AI
11440..11441  ; XID_Continue # Mc   [2] NEWA VOWEL SIGN O..AU
11442..11444  ; XID_Continue # Mn   [3] NEWA SIGN VIRAMA..ANUSVARA
11445         ; XID_Continue # Mc       NEWA SIGN VISARGA
11446         ; XID_Continue # Mn       NEWA SIGN NUKTA
1145E         ; XID_Continue # Mn       NEWA SANDHI MARK
114B0..114B2  ; XID_Continue # Mc   [3] TIRHUTA VOWEL SIGN AA..II
114B3..114B8  ; XID_Continue # Mn   [6] TIRHUTA VOWEL SIGN U..VOCALIC LL
114B9         ; XID_Continue # Mc       TIRHUTA VOWEL SIGN E
114BA         ; XID_Continue # Mn       TIRHUTA VOWEL SIGN SHORT E
114BB..114BE  ; XID_Continue # Mc   [4] TIRHUTA VOWEL SIGN AI..AU
114BF..114C0  ; XID_Continue # Mn   [2] TIRHUTA SIGN CANDRABINDU..ANUSVARA
114C1         ; XID_Continue # Mc       TIRHUTA SIGN VISARGA
114C2..114C3  ; XID_Continue # Mn   [2] TIRHUTA SIGN VIRAMA..NUKTA
115AF..115B1  ; XID_Continue # Mc   [3] SIDDHAM VOWEL SIGN AA..II
115B2..115B5  ; XID_Continue # Mn   [4] SIDDHAM VOWEL SIGN U..VOCALIC RR
115B8..115BB  ; XID_Continue # Mc   [4] SIDDHAM VOWEL SIGN E..AU
115BC..115BD  ; XID_Continue # Mn   [2] SIDDHAM SIGN CANDRABINDU..ANUSVARA
115BE         ; XID_Continue # Mc       SIDDHAM SIGN VISARGA
115BF..115C0  ; XID_Continue # Mn   [2] SIDDHAM SIGN VIRAMA..NUKTA
115DC..115DD  ; XID_Continue # Mn   [2] SIDDHAM VOWEL SIGN ALTERNATE U..UU
11630..11632  ; XID_Continue # Mc   [3] MODI VOWEL SIGN AA..II
11633..1163A  ; XID_Continue # Mn   [8] MODI VOWEL SIGN U..AI
1163B..1163C  ; XID_Continue # Mc   [2] MODI VOWEL SIGN O..AU
1163D         ; XID_Continue # Mn       MODI SIGN ANUSVARA
1163E         ; XID_Continue # Mc       MODI SIGN VISARGA
1163F..11640  ; XID_Continue # Mn   [2] MODI SIGN VIRAMA..ARDHACANDRA
116AB         ; XID_Continue # Mn       TAKRI SIGN ANUSVARA
116AC         ; XID_Continue # Mc       TAKRI SIGN VISARGA
116AD         ; XID_Continue # Mn       TAKRI VOWEL SIGN AA
116AE..116AF  ; XID_Continue # Mc   [2] TAKRI VOWEL SIGN I..II
```

```
116B0..116B5  ; XID_Continue # Mn   [6] TAKRI VOWEL SIGN U..AU
116B6         ; XID_Continue # Mc       TAKRI SIGN VIRAMA
116B7         ; XID_Continue # Mn       TAKRI SIGN NUKTA
1171D..1171F  ; XID_Continue # Mn   [3] AHOM CONSONANT SIGN MEDIAL LA..
                                        LIGATING RA
11720..11721  ; XID_Continue # Mc   [2] AHOM VOWEL SIGN A..AA
11722..11725  ; XID_Continue # Mn   [4] AHOM VOWEL SIGN I..UU
11726         ; XID_Continue # Mc       AHOM VOWEL SIGN E
11727..1172B  ; XID_Continue # Mn   [5] AHOM VOWEL SIGN AW..KILLER
1182C..1182E  ; XID_Continue # Mc   [3] DOGRA VOWEL SIGN AA..II
1182F..11837  ; XID_Continue # Mn   [9] DOGRA VOWEL SIGN U..ANUSVARA
11838         ; XID_Continue # Mc       DOGRA SIGN VISARGA
11839..1183A  ; XID_Continue # Mn   [2] DOGRA SIGN VIRAMA..NUKTA
11930..11935  ; XID_Continue # Mc   [6] DIVES AKURU VOWEL SIGN AA..E
11937..11938  ; XID_Continue # Mc   [2] DIVES AKURU VOWEL SIGN AI..O
1193B..1193C  ; XID_Continue # Mn   [2] DIVES AKURU SIGN ANUSVARA..CANDRABINDU
1193D         ; XID_Continue # Mc       DIVES AKURU SIGN HALANTA
1193E         ; XID_Continue # Mn       DIVES AKURU VIRAMA
11940         ; XID_Continue # Mc       DIVES AKURU MEDIAL YA
11942         ; XID_Continue # Mc       DIVES AKURU MEDIAL RA
11943         ; XID_Continue # Mn       DIVES AKURU SIGN NUKTA
119D1..119D3  ; XID_Continue # Mc   [3] NANDINAGARI VOWEL SIGN AA..II
119D4..119D7  ; XID_Continue # Mn   [4] NANDINAGARI VOWEL SIGN U..VOCALIC RR
119DA..119DB  ; XID_Continue # Mn   [2] NANDINAGARI VOWEL SIGN E..AI
119DC..119DF  ; XID_Continue # Mc   [4] NANDINAGARI VOWEL SIGN O..VISARGA
119E0         ; XID_Continue # Mn       NANDINAGARI SIGN VIRAMA
119E4         ; XID_Continue # Mc       NANDINAGARI VOWEL SIGN PRISHTHAMATRA E
11A01..11A0A  ; XID_Continue # Mn  [10] ZANABAZAR SQUARE VOWEL SIGN I..
                                        LENGTH MARK
11A33..11A38  ; XID_Continue # Mn   [6] ZANABAZAR SQUARE FINAL CONSONANT MARK..
                                        ZANABAZAR SQUARE SIGN ANUSVARA
11A39         ; XID_Continue # Mc       ZANABAZAR SQUARE SIGN VISARGA
11A3B..11A3E  ; XID_Continue # Mn   [4] ZANABAZAR SQUARE CLUSTER-FINAL LETTER YA..
                                        ZANABAZAR SQUARE CLUSTER-FINAL LETTER VA
11A47         ; XID_Continue # Mn       ZANABAZAR SQUARE SUBJOINER
11A51..11A56  ; XID_Continue # Mn   [6] SOYOMBO VOWEL SIGN I..OE
11A57..11A58  ; XID_Continue # Mc   [2] SOYOMBO VOWEL SIGN AI..AU
11A59..11A5B  ; XID_Continue # Mn   [3] SOYOMBO VOWEL SIGN VOCALIC R..
                                        SOYOMBO VOWEL LENGTH MARK
11A8A..11A96  ; XID_Continue # Mn  [13] SOYOMBO FINAL CONSONANT SIGN G..ANUSVARA
11A97         ; XID_Continue # Mc       SOYOMBO SIGN VISARGA
11A98..11A99  ; XID_Continue # Mn   [2] SOYOMBO GEMINATION MARK..SUBJOINER
11C2F         ; XID_Continue # Mc       BHAIKSUKI VOWEL SIGN AA
11C30..11C36  ; XID_Continue # Mn   [7] BHAIKSUKI VOWEL SIGN I..VOCALIC L
11C38..11C3D  ; XID_Continue # Mn   [6] BHAIKSUKI VOWEL SIGN E..ANUSVARA
11C3E         ; XID_Continue # Mc       BHAIKSUKI SIGN VISARGA
```

```
11C3F         ; XID_Continue # Mn       BHAIKSUKI SIGN VIRAMA
11C92..11CA7  ; XID_Continue # Mn  [22] MARCHEN SUBJOINED LETTER KA..ZA
11CA9         ; XID_Continue # Mc       MARCHEN SUBJOINED LETTER YA
11CAA..11CB0  ; XID_Continue # Mn   [7] MARCHEN SUBJOINED LETTER RA..
                                        MARCHEN VOWEL SIGN AA
11CB1         ; XID_Continue # Mc       MARCHEN VOWEL SIGN I
11CB2..11CB3  ; XID_Continue # Mn   [2] MARCHEN VOWEL SIGN U..E
11CB4         ; XID_Continue # Mc       MARCHEN VOWEL SIGN O
11CB5..11CB6  ; XID_Continue # Mn   [2] MARCHEN SIGN ANUSVARA..CANDRABINDU
11D31..11D36  ; XID_Continue # Mn   [6] MASARAM GONDI VOWEL SIGN AA..
                                        MASARAM GONDI VOWEL SIGN VOCALIC R
11D3A         ; XID_Continue # Mn       MASARAM GONDI VOWEL SIGN E
11D3C..11D3D  ; XID_Continue # Mn   [2] MASARAM GONDI VOWEL SIGN AI..O
11D3F..11D45  ; XID_Continue # Mn   [7] MASARAM GONDI VOWEL SIGN AU..
                                        MASARAM GONDI VIRAMA
11D47         ; XID_Continue # Mn       MASARAM GONDI RA-KARA
11D8A..11D8E  ; XID_Continue # Mc   [5] GUNJALA GONDI VOWEL SIGN AA..UU
11D90..11D91  ; XID_Continue # Mn   [2] GUNJALA GONDI VOWEL SIGN EE..AI
11D93..11D94  ; XID_Continue # Mc   [2] GUNJALA GONDI VOWEL SIGN OO..AU
11D95         ; XID_Continue # Mn       GUNJALA GONDI SIGN ANUSVARA
11D96         ; XID_Continue # Mc       GUNJALA GONDI SIGN VISARGA
11D97         ; XID_Continue # Mn       GUNJALA GONDI VIRAMA
11EF3..11EF4  ; XID_Continue # Mn   [2] MAKASAR VOWEL SIGN I..U
11EF5..11EF6  ; XID_Continue # Mc   [2] MAKASAR VOWEL SIGN E..O
16AF0..16AF4  ; XID_Continue # Mn   [5] BASSA VAH COMBINING HIGH TONE..
                                        BASSA VAH COMBINING HIGH-LOW TONE
16B30..16B36  ; XID_Continue # Mn   [7] PAHAWH HMONG MARK CIM TUB..CIM TAUM
16F4F         ; XID_Continue # Mn       MIAO SIGN CONSONANT MODIFIER BAR
16F51..16F87  ; XID_Continue # Mc  [55] MIAO SIGN ASPIRATION..MIAO VOWEL SIGN UI
16F8F..16F92  ; XID_Continue # Mn   [4] MIAO TONE RIGHT..MIAO TONE BELOW
16FE4         ; XID_Continue # Mn       KHITAN SMALL SCRIPT FILLER
16FF0..16FF1  ; XID_Continue # Mc   [2] VIETNAMESE ALTERNATE READING MARK CA..
                                        VIETNAMESE ALTERNATE READING MARK NHAY
1BC9D..1BC9E  ; XID_Continue # Mn   [2] DUPLOYAN THICK LETTER SELECTOR..
                                        DUPLOYAN DOUBLE MARK
1CF00..1CF2D  ; XID_Continue # Mn  [46] ZNAMENNY COMBINING MARK GORAZDO NIZKO S
                                        KRYZHEM ON LEFT..
                                        ZNAMENNY COMBINING MARK KRYZH ON LEFT
1CF30..1CF46  ; XID_Continue # Mn  [23] ZNAMENNY COMBINING TONAL RANGE MARK
                                        MRACHNO..PRIZNAK MODIFIER ROG
1D165..1D166  ; XID_Continue # Mc   [2] MUSICAL SYMBOL COMBINING STEM..
                                        SPRECHGESANG STEM
1D167..1D169  ; XID_Continue # Mn   [3] MUSICAL SYMBOL COMBINING TREMOLO-1..3
1D16D..1D172  ; XID_Continue # Mc   [6] MUSICAL SYMBOL COMBINING AUGMENTATION
                                        DOT..FLAG-5
1D17B..1D182  ; XID_Continue # Mn   [8] MUSICAL SYMBOL COMBINING ACCENT..LOURE
```

```
1D185..1D18B  ; XID_Continue # Mn   [7] MUSICAL SYMBOL COMBINING DOIT..
                                        MUSICAL SYMBOL COMBINING TRIPLE TONGUE
1D1AA..1D1AD  ; XID_Continue # Mn   [4] MUSICAL SYMBOL COMBINING DOWN BOW..
                                        MUSICAL SYMBOL COMBINING SNAP PIZZICATO
1D242..1D244  ; XID_Continue # Mn   [3] COMBINING GREEK MUSICAL TRISEME..
                                        COMBINING GREEK MUSICAL PENTASEME
1DA00..1DA36  ; XID_Continue # Mn  [55] SIGNWRITING HEAD RIM..
                                        SIGNWRITING AIR SUCKING IN
1DA3B..1DA6C  ; XID_Continue # Mn  [50] SIGNWRITING MOUTH CLOSED NEUTRAL..
                                        SIGNWRITING EXCITEMENT
1DA75         ; XID_Continue # Mn       SIGNWRITING UPPER BODY TILTING FROM
                                        HIP JOINTS
1DA84         ; XID_Continue # Mn       SIGNWRITING LOCATION HEAD NECK
1DA9B..1DA9F  ; XID_Continue # Mn   [5] SIGNWRITING FILL MODIFIER-2..
                                        SIGNWRITING FILL MODIFIER-6
1DAA1..1DAAF  ; XID_Continue # Mn  [15] SIGNWRITING ROTATION MODIFIER-2..-16
1E000..1E006  ; XID_Continue # Mn   [7] COMBINING GLAGOLITIC LETTER AZU..ZHIVETE
1E008..1E018  ; XID_Continue # Mn  [17] COMBINING GLAGOLITIC LETTER ZEMLJA..HERU
1E01B..1E021  ; XID_Continue # Mn   [7] COMBINING GLAGOLITIC LETTER SHTA..YATI
1E023..1E024  ; XID_Continue # Mn   [2] COMBINING GLAGOLITIC LETTER YU..SMALL YUS
1E026..1E02A  ; XID_Continue # Mn   [5] COMBINING GLAGOLITIC LETTER YO..FITA
1E130..1E136  ; XID_Continue # Mn   [7] NYIAKENG PUACHUE HMONG TONE-B..-D
1E2AE         ; XID_Continue # Mn       TOTO SIGN RISING TONE
1E2EC..1E2EF  ; XID_Continue # Mn   [4] WANCHO TONE TUP..WANCHO TONE KOINI
1E8D0..1E8D6  ; XID_Continue # Mn   [7] MENDE KIKAKUI COMBINING NUMBER TEENS..
                                        MENDE KIKAKUI COMBINING NUMBER MILLIONS
1E944..1E94A  ; XID_Continue # Mn   [7] ADLAM ALIF LENGTHENER..ADLAM NUKTA
E0100..E01EF  ; XID_Continue # Mn [240] VARIATION SELECTOR-17..-256
```

17 Appendix E - IDType Technical

Needed for Section 9 TR39 Identifier Type. List of Technical ID characters, added to the TR39 Recommended and Inclusion IDTypes. TR39#Table 1 https://www.unicode.org/reports/tr39/#Identifier_Status_and_Type. In guidance with TR39.

The confusables

```
01C0..01C3   ; Technical  # 1.1   [4] LATIN LETTER DENTAL CLICK..
                                       RETROFLEX CLICK
```

are excluded here.

```
grep ' Technical ' IdentifierType.txt |
 egrep -v 'Not_XID|Obsolete|Exclusion|Uncommon_Use|Limited_Use'
```

```
0180          ; Technical  # 1.1        LATIN SMALL LETTER B WITH STROKE
```

```
0234..0236    ; Technical  # 4.0    [3] LATIN SMALL LETTER L WITH CURL..
                                        T WITH CURL
0250..0252    ; Technical  # 1.1    [3] LATIN SMALL LETTER TURNED A..ALPHA
0255          ; Technical  # 1.1        LATIN SMALL LETTER C WITH CURL
0258          ; Technical  # 1.1        LATIN SMALL LETTER REVERSED E
025A          ; Technical  # 1.1        LATIN SMALL LETTER SCHWA WITH HOOK
025C..0262    ; Technical  # 1.1    [7] LATIN SMALL LETTER REVERSED OPEN E..
                                        LATIN LETTER SMALL CAPITAL G
0264..0267    ; Technical  # 1.1    [4] LATIN SMALL LETTER RAMS HORN..
                                        LATIN SMALL LETTER HENG WITH HOOK
026A..0271    ; Technical  # 1.1    [8] LATIN LETTER SMALL CAPITAL I..
                                        LATIN SMALL LETTER M WITH HOOK
0273..0276    ; Technical  # 1.1    [4] LATIN SMALL LETTER N WITH RETROFLEX
                                        HOOK..LATIN LETTER SMALL CAPITAL OE
0278..027B    ; Technical  # 1.1    [4] LATIN SMALL LETTER PHI..
                                        LATIN SMALL LETTER TURNED R WITH HOOK
027D..0288    ; Technical  # 1.1   [12] LATIN SMALL LETTER R WITH TAIL..
                                        LATIN SMALL LETTER T WITH RETROFLEX HOOK
028A..0291    ; Technical  # 1.1    [8] LATIN SMALL LETTER UPSILON..
                                        LATIN SMALL LETTER Z WITH CURL
0293..029D    ; Technical  # 1.1   [11] LATIN SMALL LETTER EZH WITH CURL..
                                        LATIN SMALL LETTER J WITH CROSSED-TAIL
029F..02A8    ; Technical  # 1.1   [10] LATIN LETTER SMALL CAPITAL L..
                                        LATIN SMALL LETTER TC DIGRAPH WITH CURL
02A9..02AD    ; Technical  # 3.0    [5] LATIN SMALL LETTER FENG DIGRAPH..
                                        LATIN LETTER BIDENTAL PERCUSSIVE
02AE..02AF    ; Technical  # 4.0    [2] LATIN SMALL LETTER TURNED H WITH
                                        FISHHOOK..AND TAIL
02B9..02BA    ; Technical  # 1.1    [2] MODIFIER LETTER PRIME..DOUBLE PRIME
02BD..02C1    ; Technical  # 1.1    [5] MODIFIER LETTER REVERSED COMMA..
                                        MODIFIER LETTER REVERSED GLOTTAL STOP
02C6..02D1    ; Technical  # 1.1   [12] MODIFIER LETTER CIRCUMFLEX ACCENT..
                                        MODIFIER LETTER HALF TRIANGULAR COLON
02EE          ; Technical  # 3.0        MODIFIER LETTER DOUBLE APOSTROPHE
030E          ; Technical  # 1.1        COMBINING DOUBLE VERTICAL LINE ABOVE
0312          ; Technical  # 1.1        COMBINING TURNED COMMA ABOVE
0315          ; Technical  # 1.1        COMBINING COMMA ABOVE RIGHT
0317..031A    ; Technical  # 1.1    [4] COMBINING ACUTE ACCENT BELOW..
                                        COMBINING LEFT ANGLE ABOVE
031C..0320    ; Technical  # 1.1    [5] COMBINING LEFT HALF RING BELOW..
                                        COMBINING MINUS SIGN BELOW
0329..032C    ; Technical  # 1.1    [4] COMBINING VERTICAL LINE BELOW..
                                        COMBINING CARON BELOW
032F          ; Technical  # 1.1        COMBINING INVERTED BREVE BELOW
0333          ; Technical  # 1.1        COMBINING DOUBLE LOW LINE
0337          ; Technical  # 1.1        COMBINING SHORT SOLIDUS OVERLAY
```

```
033A..033F    ; Technical  # 1.1    [6] COMBINING INVERTED BRIDGE BELOW..
                                        COMBINING DOUBLE OVERLINE
0346..034E    ; Technical  # 3.0    [9] COMBINING BRIDGE ABOVE..
                                        COMBINING UPWARDS ARROW BELOW
0350..0357    ; Technical  # 4.0    [8] COMBINING RIGHT ARROWHEAD ABOVE..
                                        HALF RING ABOVE
0359..035C    ; Technical  # 4.1    [4] COMBINING ASTERISK BELOW..
                                        COMBINING DOUBLE BREVE BELOW
035D..035F    ; Technical  # 4.0    [3] COMBINING DOUBLE BREVE..MACRON BELOW
0360..0361    ; Technical  # 1.1    [2] COMBINING DOUBLE TILDE..INVERTED BREVE
0362          ; Technical  # 3.0        COMBINING DOUBLE RIGHTWARDS ARROW BELOW
03CF          ; Technical  # 5.1        GREEK CAPITAL KAI SYMBOL
03D7          ; Technical  # 3.0        GREEK KAI SYMBOL
0560          ; Technical  # 11.0       ARMENIAN SMALL LETTER TURNED AYB
0588          ; Technical  # 11.0       ARMENIAN SMALL LETTER YI WITH STROKE
0953..0954    ; Technical  # 1.1    [2] DEVANAGARI GRAVE ACCENT..
                                        DEVANAGARI ACUTE ACCENT
0D81          ; Technical  # 13.0       SINHALA SIGN CANDRABINDU
0F18..0F19    ; Technical  # 2.0    [2] TIBETAN ASTROLOGICAL SIGN -KHYUD PA..
                                        TIBETAN ASTROLOGICAL SIGN SDONG TSHUGS
17CE..17CF    ; Technical  # 3.0    [2] KHMER SIGN KAKABAT..
                                        KHMER SIGN AHSDA
1ABF..1AC0    ; Technical  # 13.0   [2] COMBINING LATIN SMALL LETTER W BELOW..
                                        TURNED W BELOW
1D00..1D2B    ; Technical  # 4.0   [44] LATIN LETTER SMALL CAPITAL A..
                                        CYRILLIC LETTER SMALL CAPITAL EL
1D2F          ; Technical  # 4.0        MODIFIER LETTER CAPITAL BARRED B
1D3B          ; Technical  # 4.0        MODIFIER LETTER CAPITAL REVERSED N
1D4E          ; Technical  # 4.0        MODIFIER LETTER SMALL TURNED I
1D6B          ; Technical  # 4.0        LATIN SMALL LETTER UE
1D6C..1D77    ; Technical  # 4.1   [12] LATIN SMALL LETTER B WITH MIDDLE TILDE..
                                        LATIN SMALL LETTER TURNED G
1D79..1D9A    ; Technical  # 4.1   [34] LATIN SMALL LETTER INSULAR G..
                                        EZH WITH RETROFLEX HOOK
1DC4..1DCA    ; Technical  # 5.0    [7] COMBINING MACRON-ACUTE..
                                        COMBINING LATIN SMALL LETTER R BELOW
1DCB..1DCD    ; Technical  # 5.1    [3] COMBINING BREVE-MACRON..
                                        COMBINING DOUBLE CIRCUMFLEX ABOVE
1DCF..1DD0    ; Technical  # 5.1    [2] COMBINING ZIGZAG BELOW..
                                        COMBINING IS BELOW
1DE7..1DF5    ; Technical  # 7.0   [15] COMBINING LATIN SMALL LETTER ALPHA..
                                        COMBINING UP TACK ABOVE
1DF6..1DF9    ; Technical  # 10.0   [4] COMBINING KAVYKA ABOVE RIGHT..
                                        COMBINING WIDE INVERTED BRIDGE BELOW
1DFB          ; Technical  # 9.0        COMBINING DELETION MARK
1DFC          ; Technical  # 6.0        COMBINING DOUBLE INVERTED BREVE BELOW
```

```
1DFD          ; Technical  # 5.2         COMBINING ALMOST EQUAL TO BELOW
1DFE..1DFF    ; Technical  # 5.0    [2] COMBINING LEFT ARROWHEAD ABOVE..
                                        COMBINING RIGHT ARROWHEAD AND DOWN
                                        ARROWHEAD BELOW
1E9C..1E9D    ; Technical  # 5.1    [2] LATIN SMALL LETTER LONG S WITH DIAGONAL
                                        STROKE..WITH HIGH STROKE
1E9F          ; Technical  # 5.1         LATIN SMALL LETTER DELTA
1EFA..1EFF    ; Technical  # 5.1    [6] LATIN CAPITAL LETTER MIDDLE-WELSH LL..
                                        LATIN SMALL LETTER Y WITH LOOP
203F..2040    ; Technical  # 1.1    [2] UNDERTIE..
                                        CHARACTER TIE
20D0..20DC    ; Technical  # 1.1   [13] COMBINING LEFT HARPOON ABOVE..
                                        COMBINING FOUR DOTS ABOVE
20E1          ; Technical  # 1.1         COMBINING LEFT RIGHT ARROW ABOVE
20E5..20EA    ; Technical  # 3.2    [6] COMBINING REVERSE SOLIDUS OVERLAY..
                                        COMBINING LEFTWARDS ARROW OVERLAY
20EB          ; Technical  # 4.1         COMBINING LONG DOUBLE SOLIDUS OVERLAY
20EC..20EF    ; Technical  # 5.0    [4] COMBINING RIGHTWARDS HARPOON WITH BARB
                                        DOWNWARDS..COMBINING RIGHT ARROW BELOW
20F0          ; Technical  # 5.1         COMBINING ASTERISK ABOVE
2118          ; Technical  # 1.1         SCRIPT CAPITAL P
212E          ; Technical  # 1.1         ESTIMATED SYMBOL
2C60..2C67    ; Technical  # 5.0    [8] LATIN CAPITAL LETTER L WITH DOUBLE BAR..
                                        LATIN CAPITAL LETTER H WITH DESCENDER
2C77          ; Technical  # 5.0         LATIN SMALL LETTER TAILLESS PHI
2C78..2C7B    ; Technical  # 5.1    [4] LATIN SMALL LETTER E WITH NOTCH..
                                        LATIN LETTER SMALL CAPITAL TURNED E
3021..302D    ; Technical  # 1.1   [13] HANGZHOU NUMERAL ONE..
                                        IDEOGRAPHIC ENTERING TONE MARK
3031..3035    ; Technical  # 1.1    [5] VERTICAL KANA REPEAT MARK..
                                        VERTICAL KANA REPEAT MARK LOWER HALF
303B..303C    ; Technical  # 3.2    [2] VERTICAL IDEOGRAPHIC ITERATION MARK..
                                        MASU MARK
A78E          ; Technical  # 6.0         LATIN SMALL LETTER L WITH RETROFLEX HOOK
                                        AND BELT
A7AF          ; Technical  # 11.0        LATIN LETTER SMALL CAPITAL Q
A7BA..A7BF    ; Technical  # 12.0   [6] LATIN CAPITAL LETTER GLOTTAL A..
                                        LATIN SMALL LETTER GLOTTAL U
A7FA          ; Technical  # 6.0         LATIN LETTER SMALL CAPITAL TURNED M
AB68          ; Technical  # 13.0        LATIN SMALL LETTER TURNED R WITH MIDDLE
                                        TILDE
FE20..FE23    ; Technical  # 1.1    [4] COMBINING LIGATURE LEFT HALF..
                                        COMBINING DOUBLE TILDE RIGHT HALF
FE24..FE26    ; Technical  # 5.1    [3] COMBINING MACRON LEFT HALF..
                                        COMBINING CONJOINING MACRON
FE27..FE2D    ; Technical  # 7.0    [7] COMBINING LIGATURE LEFT HALF BELOW..
```

```
                                          COMBINING CONJOINING MACRON BELOW
FE73          ; Technical  # 3.2          ARABIC TAIL FRAGMENT
1CF00..1CF2D  ; Technical  # 14.0  [46] ZNAMENNY COMBINING MARK GORAZDO NIZKO S
                                          KRYZHEM ON LEFT..KRYZH ON LEFT
1CF30..1CF46  ; Technical  # 14.0  [23] ZNAMENNY COMBINING TONAL RANGE MARK
                                          MRACHNO..PRIZNAK MODIFIER ROG
1D165..1D169  ; Technical  # 3.1    [5] MUSICAL SYMBOL COMBINING STEM..TREMOLO-3
1D16D..1D172  ; Technical  # 3.1    [6] MUSICAL SYMBOL COMBINING AUGMENTATION
                                          DOT..MUSICAL SYMBOL COMBINING FLAG-5
1D17B..1D182  ; Technical  # 3.1    [8] MUSICAL SYMBOL COMBINING ACCENT..LOURE
1D185..1D18B  ; Technical  # 3.1    [7] MUSICAL SYMBOL COMBINING DOIT..
                                          MUSICAL SYMBOL COMBINING TRIPLE TONGUE
1D1AA..1D1AD  ; Technical  # 3.1    [4] MUSICAL SYMBOL COMBINING DOWN BOW..
                                          MUSICAL SYMBOL COMBINING SNAP PIZZICATO
```

18 Appendix F - Greek Confusables

Needed for exclusion in the Section 9 TR39 Mixed Scripts Greek rule. Where-ever we have a Greek letter confusable with Latin, and we already saw Latin, forbid the Greek letter in favor of the Latin letter. See TR39 confusables. Note that these confusables cannot be excluded upfront in the TR31 identifier parsing, as Greek alone is allowed.

18.1 Exceptions

Allow these 12 Greek letters and symbols to be confusable with Latin: 037A, 0381, 0398, 03B5, 03B7, 03B8, 03B9, 03BD, 03C3, 03D1, 03F1, 03F4. The confusables.txt list is extremely buggy.

```
037A ; ( ͺ → i ) GREEK YPOGEGRAMMENI → LATIN SMALL LETTER I
0381 ; ( Ά → a ) GREEK SMALL LETTER ALPHA
0398 ; ( Θ → O– ) GREEK CAPITAL LETTER THETA → LATIN CAPITAL LETTER O, ...
03B5 ; ( ε →  ) GREEK SMALL LETTER EPSILON
03B7 ; ( η → ῃ ) GREEK SMALL LETTER ETA → LATIN SMALL LETTER N, COMBINING
                   VERTICAL LINE BELOW
03B8 ; ( θ → O– ) GREEK SMALL LETTER THETA → LATIN CAPITAL LETTER O, ...
03B9 ; ( ι → i ) GREEK SMALL LETTER IOTA → LATIN SMALL LETTER I
03BD ; ( ν → v ) GREEK SMALL LETTER NU → LATIN SMALL LETTER V
03C3 ; ( σ → o ) GREEK SMALL LETTER SIGMA → LATIN SMALL LETTER O
03D1 ; ( ϑ → O– ) GREEK THETA SYMBOL → LATIN CAPITAL LETTER O, ...
03F1 ; ( ϱ → p ) GREEK RHO SYMBOL → LATIN SMALL LETTER P
03F4 ; ( Θ → O– ) GREEK CAPITAL THETA SYMBOL → LATIN CAPITAL LETTER O, ...
```

18.2 Confusables

List of all the Greek-Latin confusables: Note, these still include the exceptions above.

```
grep GREEK confusables.txt | grep LETTER | grep LATIN
```

```
03B1 ; ( α → a ) GREEK SMALL LETTER ALPHA → LATIN SMALL LETTER A
0391 ; ( Α → A ) GREEK CAPITAL LETTER ALPHA → LATIN CAPITAL LETTER A
1D217; ( 𝈗 → ∀ ) GREEK VOCAL NOTATION SYMBOL-24 → LATIN CAPITAL LETTER TURNED A
0392 ; ( Β → B ) GREEK CAPITAL LETTER BETA → LATIN CAPITAL LETTER B
03F2 ; ( ϲ → c ) GREEK LUNATE SIGMA SYMBOL → LATIN SMALL LETTER C
03F9 ; ( Ϲ → C ) GREEK CAPITAL LUNATE SIGMA SYMBOL → LATIN CAPITAL LETTER C
03B5 ; ( ε → ϵ ) GREEK SMALL LETTER EPSILON → LATIN SMALL LETTER C WITH BAR
03F5 ; ( ϵ → ϵ ) GREEK LUNATE EPSILON SYMBOL → LATIN SMALL LETTER C WITH BAR
037D ; ( ͽ → Ͻ ) GREEK SMALL REVERSED DOTTED LUNATE SIGMA SYMBOL → LATIN SMALL
                  LETTER REVERSED C WITH DOT
03FF ; ( Ͽ → Ͻ ) GREEK CAPITAL REVERSED DOTTED LUNATE SIGMA SYMBOL → LATIN CAPITAL
                  LETTER REVERSED C WITH DOT
03B4 ; ( δ → δ ) GREEK SMALL LETTER DELTA → LATIN SMALL LETTER DELTA
0395 ; ( Ε → E ) GREEK CAPITAL LETTER EPSILON → LATIN CAPITAL LETTER E
1D221; ( 𝈡 → Ɛ ) GREEK INSTRUMENTAL NOTATION SYMBOL-7 → LATIN CAPITAL LETTER
                  OPEN E
1D213; ( 𝈓 → F ) GREEK VOCAL NOTATION SYMBOL-20 → LATIN CAPITAL LETTER F
03DC ; ( Ϝ → F ) GREEK LETTER DIGAMMA → LATIN CAPITAL LETTER F
1D230; ( 𝈰 → Ⅎ ) GREEK INSTRUMENTAL NOTATION SYMBOL-30 → LATIN EPIGRAPHIC
                  LETTER REVERSED F
0397 ; ( Η → H ) GREEK CAPITAL LETTER ETA → LATIN CAPITAL LETTER H
0370 ; ( Ͱ → Ⱶ ) GREEK CAPITAL LETTER HETA → LATIN CAPITAL LETTER HALF H
03B9 ; ( ι → i ) GREEK SMALL LETTER IOTA → LATIN SMALL LETTER I
1FBE ; ( ι → i ) GREEK PROSGEGRAMMENI → LATIN SMALL LETTER I
037A ; ( ͺ → i ) GREEK YPOGEGRAMMENI → LATIN SMALL LETTER I
03F3 ; ( ϳ → j ) GREEK LETTER YOT → LATIN SMALL LETTER J
037F ; ( Ϳ → J ) GREEK CAPITAL LETTER YOT → LATIN CAPITAL LETTER J
039A ; ( Κ → K ) GREEK CAPITAL LETTER KAPPA → LATIN CAPITAL LETTER K
0399 ; ( Ι → l ) GREEK CAPITAL LETTER IOTA → LATIN SMALL LETTER L
1D22A; ( 𝈪 → L ) GREEK INSTRUMENTAL NOTATION SYMBOL-23 → LATIN CAPITAL LETTER L
039C ; ( Μ → M ) GREEK CAPITAL LETTER MU → LATIN CAPITAL LETTER M
03FA ; ( Ϻ → M ) GREEK CAPITAL LETTER SAN → LATIN CAPITAL LETTER M
039D ; ( Ν → N ) GREEK CAPITAL LETTER NU → LATIN CAPITAL LETTER N
03B7 ; ( η → ɳ ) GREEK SMALL LETTER ETA → LATIN SMALL LETTER N, ...
0377 ; ( ͷ → ͵ ) GREEK SMALL LETTER PAMPHYLIAN DIGAMMA → LATIN LETTER SMALL
                  CAPITAL REVERSED N
03BF ; ( ο → o ) GREEK SMALL LETTER OMICRON → LATIN SMALL LETTER O
039F ; ( Ο → O ) GREEK CAPITAL LETTER OMICRON → LATIN CAPITAL LETTER O
1D21A; ( 𝈚 → O‒ ) GREEK VOCAL NOTATION SYMBOL-52 → LATIN CAPITAL LETTER O, ...
03B8 ; ( θ → O‒ ) GREEK SMALL LETTER THETA → LATIN CAPITAL LETTER O, ...
```

```
03D1 ; ( ϑ → O– ) GREEK THETA SYMBOL → LATIN CAPITAL LETTER O, ...
0398 ; ( Θ → O– ) GREEK CAPITAL LETTER THETA → LATIN CAPITAL LETTER O, ...
03F4 ; ( ϴ → O– ) GREEK CAPITAL THETA SYMBOL → LATIN CAPITAL LETTER O, ...
037B ; ( ϲ → ↄ ) GREEK SMALL REVERSED LUNATE SIGMA SYMBOL → LATIN SMALL
                  LETTER OPEN O
03FD ; ( Ϗ → Ↄ ) GREEK CAPITAL REVERSED LUNATE SIGMA SYMBOL → LATIN CAPITAL
                  LETTER OPEN O
03C1 ; ( ρ → p ) GREEK SMALL LETTER RHO → LATIN SMALL LETTER P
03F1 ; ( ϱ → p ) GREEK RHO SYMBOL → LATIN SMALL LETTER P
03A1 ; ( Ρ → P ) GREEK CAPITAL LETTER RHO → LATIN CAPITAL LETTER P
1D29 ; ( ᴩ → ꟼ ) GREEK LETTER SMALL CAPITAL RHO → LATIN LETTER SMALL CAPITAL P
03C6 ; ( φ → ɸ ) GREEK SMALL LETTER PHI → LATIN SMALL LETTER PHI
03D5 ; ( ϕ → ɸ ) GREEK PHI SYMBOL → LATIN SMALL LETTER PHI
03BA ; ( κ → ĸ ) GREEK SMALL LETTER KAPPA → LATIN SMALL LETTER KRA
03F0 ; ( ϰ → ĸ ) GREEK KAPPA SYMBOL → LATIN SMALL LETTER KRA
1D26 ; ( ᴦ → ɾ ) GREEK LETTER SMALL CAPITAL GAMMA → LATIN SMALL LETTER R
1D216; ( 𝈖 → R ) GREEK VOCAL NOTATION SYMBOL-23 → LATIN CAPITAL LETTER R
2129 ; ( ℩ → ɹ ) TURNED GREEK SMALL LETTER IOTA → LATIN SMALL LETTER
                  REVERSED R WITH FISHHOOK
03B2 ; ( β → ß ) GREEK SMALL LETTER BETA → LATIN SMALL LETTER SHARP S
03D0 ; ( ϐ → ß ) GREEK BETA SYMBOL → LATIN SMALL LETTER SHARP S
03A3 ; ( Σ → Σ ) GREEK CAPITAL LETTER SIGMA → LATIN CAPITAL LETTER ESH
03A4 ; ( Τ → T ) GREEK CAPITAL LETTER TAU → LATIN CAPITAL LETTER T
03C4 ; ( τ → ᴛ ) GREEK SMALL LETTER TAU → LATIN LETTER SMALL CAPITAL T
03C5 ; ( υ → u ) GREEK SMALL LETTER UPSILON → LATIN SMALL LETTER U
1D20D; ( 𝈍 → V ) GREEK VOCAL NOTATION SYMBOL-14 → LATIN CAPITAL LETTER V
1D27 ; ( ᴧ → ʌ ) GREEK LETTER SMALL CAPITAL LAMDA → LATIN SMALL LETTER TURNED V
039B ; ( Λ → Λ ) GREEK CAPITAL LETTER LAMDA → LATIN CAPITAL LETTER TURNED V
03A7 ; ( Χ → X ) GREEK CAPITAL LETTER CHI → LATIN CAPITAL LETTER X
03B3 ; ( γ → y ) GREEK SMALL LETTER GAMMA → LATIN SMALL LETTER Y
03A5 ; ( Υ → Y ) GREEK CAPITAL LETTER UPSILON → LATIN CAPITAL LETTER Y
03D2 ; ( ϒ → Y ) GREEK UPSILON WITH HOOK SYMBOL → LATIN CAPITAL LETTER Y
0396 ; ( Ζ → Z ) GREEK CAPITAL LETTER ZETA → LATIN CAPITAL LETTER Z
03F8 ; ( þ → þ ) GREEK SMALL LETTER SHO → LATIN SMALL LETTER THORN
03F7 ; ( Þ → Þ ) GREEK CAPITAL LETTER SHO → LATIN CAPITAL LETTER THORN
03C7 ; ( ᵪ → χ ) LATIN SMALL LETTER CHI → GREEK SMALL LETTER CHI
03C9 ; ( ᵡ → ω ) LATIN SMALL LETTER OMEGA → GREEK SMALL LETTER OMEGA
```

19 Appendix G - Medial

List of all the medial letter and mark ranges. These characters are treated wrongly in all programming languages I checked. In the UCD Standard some are wrongly in `XID_Start`, but must be treated as `XID_Continue`, with a special check that they must not be in the final position of an identifier. Here we prove that for C++26 we don't need

to check for medial positions, because we restrict our TR31 set.

```
grep "; XID_Start " DerivedCoreProperties.txt | grep MEDIAL
```

```
FE77          ; XID_Start # Lo       ARABIC FATHA MEDIAL FORM
FE79          ; XID_Start # Lo       ARABIC DAMMA MEDIAL FORM
FE7B          ; XID_Start # Lo       ARABIC KASRA MEDIAL FORM
FE7D          ; XID_Start # Lo       ARABIC SHADDA MEDIAL FORM
FE7F..FEFC    ; XID_Start # Lo [126] ARABIC SUKUN MEDIAL FORM
                                     ..ARABIC LIGATURE LAM WITH ALEF FINAL FORM
```

All these are in the excluded Arabic Presentation Forms-B: U+FE70–U+FEFF block.

The ones which are correctly in XID_Continue:

```
grep "; XID_Continue " DerivedCoreProperties.txt | grep MEDIAL
```

```
103B..103C    ; XID_Continue # Mc   [2] MYANMAR CONSONANT SIGN MEDIAL YA
                                        ..MYANMAR CONSONANT SIGN MEDIAL RA
103D..103E    ; XID_Continue # Mn   [2] MYANMAR CONSONANT SIGN MEDIAL WA
                                        ..MYANMAR CONSONANT SIGN MEDIAL HA
105E..1060    ; XID_Continue # Mn   [3] MYANMAR CONSONANT SIGN MON MEDIAL NA
                                        ..MYANMAR CONSONANT SIGN MON MEDIAL LA
1082          ; XID_Continue # Mn       MYANMAR CONSONANT SIGN SHAN MEDIAL WA
1A55          ; XID_Continue # Mc       TAI THAM CONSONANT SIGN MEDIAL RA
1A56          ; XID_Continue # Mn       TAI THAM CONSONANT SIGN MEDIAL LA
FE77          ; XID_Continue # Lo       ARABIC FATHA MEDIAL FORM
FE79          ; XID_Continue # Lo       ARABIC DAMMA MEDIAL FORM
FE7B          ; XID_Continue # Lo       ARABIC KASRA MEDIAL FORM
FE7D          ; XID_Continue # Lo       ARABIC SHADDA MEDIAL FORM
FE7F..FEFC    ; XID_Continue # Lo [126] ARABIC SUKUN MEDIAL FORM
                                        ..ARABIC LIGATURE LAM WITH ALEF FINAL FORM
1171D..1171F  ; XID_Continue # Mn   [3] AHOM CONSONANT SIGN MEDIAL LA
                                        ..AHOM CONSONANT SIGN MEDIAL LIGATING RA
11940         ; XID_Continue # Mc       DIVES AKURU MEDIAL YA
11942         ; XID_Continue # Mc       DIVES AKURU MEDIAL RA
```

All these are either combining marks or in the excluded Arabic Presentation Forms-B: U+FE70–U+FEFF block.

Then see also https://www.unicode.org/reports/tr31/#Table_Optional_Medial, even they are mostly not part of any TR31 XID set. For us relevant is only the Catalan U+B7 MIDDLE DOT, which is an identifier in the Latin script. There is no Catalan script (yet), so we cannot disallow that via our mixed script check. Hence we explicitly disallow the '·' U+B7 MIDDLE DOT and punish all our Catalan programmers for security reasons. Usage of the special Catalan characters 'Ŀ' (U+013F) and 'ŀ' (U+0140) for this usecase is also disallowed as they are not NFKC. See https://en.wikipedia.org/wiki/Catalan_orthograph

y#Punt_volat_(middot) and https://en.wikipedia.org/wiki/Interpunct. If this turns out too strict add a NFKC exception to allow ‘L·’ (U+013F) and ‘l·’ (U+0140).

Other middle dot usages have their own codepoints and their own scripts, such as Chinese U+2027, Katakana U+30FB and U+FF65 (Not NFKC), Hangul U+318D, Canadian Aboriginal Syllabics U+1427 (Limited Use Script), and the Latin U+A78F (Uncommon Use).

```
00B7          ; XID_Continue # Po    MIDDLE DOT
```

So there is no medial character to consider, also no initial, isolated, nor final positions in the Arabic presentation forms.

20 Appendix H - Letters with non-spacing marks

List of letters already including its non-spacing mark:

NSM: GRAVE 300

```
 0xc0, 0xc8, 0xcc, 0xd2, 0xd9, 0xe0, 0xe8, 0xec, 0xf2, 0xf9,
0x1f8, 0x1f9, 0x400, 0x40d, 0x450, 0x45d, 0x1e80, 0x1e81, 0x1ef2,
 0x1ef3
```

NSM: ACUTE 301

```
 0xc1, 0xc9, 0xcd, 0xd3, 0xda, 0xdd, 0xe1, 0xe9, 0xed, 0xf3,
0xfa, 0xfd, 0x106, 0x107, 0x139, 0x13a, 0x143, 0x144, 0x154, 0x155,
0x15a, 0x15b, 0x179, 0x17a, 0x1f4, 0x1f5, 0x1fc, 0x1fd, 0x1e30,
 0x1e31, 0x1e3e, 0x1e3f, 0x1e54, 0x1e55, 0x1e82, 0x1e83
```

NSM: CIRCUMFLEX 302

```
 0xc2, 0xca, 0xce, 0xd4, 0xdb, 0xe2, 0xea, 0xee, 0xf4, 0xfb,
0x108, 0x109, 0x11c, 0x11d, 0x124, 0x125, 0x134, 0x135, 0x15c,
 0x15d, 0x174, 0x175, 0x176, 0x177, 0x1e90, 0x1e91
```

NSM: TILDE 303

```
0xc3, 0xd1, 0xd5, 0xe3, 0xf1, 0xf5, 0x128, 0x129, 0x168, 0x169,
 0x1e7c, 0x1e7d, 0x1ebc, 0x1ebd, 0x1ef8, 0x1ef9
```

NSM: MACRON 304

```
0x100, 0x101, 0x112, 0x113, 0x12a, 0x12b, 0x14c, 0x14d, 0x16a,
0x16b, 0x1e2, 0x1e3, 0x232, 0x233, 0x4e2, 0x4e3, 0x4ee, 0x4ef,
0x1e20, 0x1e21, 0x1fb1, 0x1fb9, 0x1fd1, 0x1fd9, 0x1fe1, 0x1fe9
```

NSM: OVERSCORE 305

```
None
```

NSM: BREVE 306

```
0x102, 0x103, 0x114, 0x115, 0x11e, 0x11f, 0x12c, 0x12d, 0x14e,
0x14f, 0x16c, 0x16d, 0x4c1, 0x4c2, 0x4d0, 0x4d1, 0x4d6, 0x4d7
```

NSM: DOT ABOVE 307

```
0x10a, 0x10b, 0x116, 0x117, 0x120, 0x121, 0x130, 0x17b, 0x17c,
0x226, 0x227, 0x22e, 0x22f, 0x6a7, 0x6ac, 0x6b6, 0x6bf, 0x6cf,
0x762, 0x765, 0x87a, 0x1e02, 0x1e03, 0x1e0a, 0x1e0b, 0x1e1e, 0x1e1f,
0x1e22, 0x1e23, 0x1e40, 0x1e41, 0x1e44, 0x1e45, 0x1e56, 0x1e57, 0x1e58,
0x1e59, 0x1e60, 0x1e61, 0x1e6a, 0x1e6b, 0x1e86, 0x1e87, 0x1e8a, 0x1e8b,
 0x1e8e, 0x1e8f, 0x1e9b, 0x312e, 0x10eb0
```

NSM: DIAERESIS 308

```
 0xc4, 0xcb, 0xcf, 0xd6, 0xdc, 0xe4, 0xeb, 0xef, 0xf6, 0xfc,
0xff, 0x178, 0x4d2, 0x4d3, 0x4da, 0x4db, 0x4dc, 0x4dd, 0x4de, 0x4df,
0x4e4, 0x4e5, 0x4e6, 0x4e7, 0x4ea, 0x4eb, 0x4ec, 0x4ed, 0x4f0, 0x4f1,
0x4f4, 0x4f5, 0x4f8, 0x4f9, 0x1df2, 0x1df3, 0x1df4, 0x1e26, 0x1e27,
 0x1e84, 0x1e85, 0x1e8c, 0x1e8d, 0x1e97
```

NSM: HOOK ABOVE 309

```
0x1ea2, 0x1ea3, 0x1eba, 0x1ebb, 0x1ec8, 0x1ec9, 0x1ece, 0x1ecf, 0x1ee6,
 0x1ee7, 0x1ef6, 0x1ef7
```

NSM: RING ABOVE 30a

```
0xc5, 0xe5, 0x16e, 0x16f, 0x1e98, 0x1e99
```

NSM: DOUBLE ACUTE 30b

```
0x150, 0x151, 0x170, 0x171, 0x4f2, 0x4f3
```

NSM: HACEK 30c NSM: VERTICAL LINE ABOVE 30d NSM: DOUBLE VERTICAL LINE ABOVE 30e

```
None
```

NSM: DOUBLE GRAVE 30f

```
0x200, 0x201, 0x204, 0x205, 0x208, 0x209, 0x20c, 0x20d, 0x210, 0x211,
 0x214, 0x215
```

NSM: CANDRABINDU 310

```
None
```

NSM: INVERTED BREVE 311

```
0x202, 0x203, 0x206, 0x207, 0x20a, 0x20b, 0x20e, 0x20f, 0x212, 0x213,
 0x216, 0x217
```

NSM: TURNED COMMA ABOVE 312 NSM: COMMA ABOVE 313

```
Todo
```

NSM: REVERSED COMMA ABOVE 314

```
Todo
```

NSM: COMMA ABOVE RIGHT 315 NSM: GRAVE BELOW 316 NSM: ACUTE BELOW 317 NSM: LEFT TACK BELOW 318 NSM: RIGHT TACK BELOW 319 NSM: LEFT ANGLE ABOVE 31a

```
None
```

NSM: HORN 31b

```
  0x1a0, 0x1a1, 0x1af, 0x1b0
```

NSM: LEFT HALF RING BELOW 31c NSM: UP TACK BELOW 31d NSM: DOWN TACK BELOW 31e NSM: PLUS SIGN BELOW 31f NSM: MINUS SIGN BELOW 320 NSM: PALATALIZED HOOK BELOW 321 NSM: RETROFLEX HOOK BELOW 322

```
  None
```

NSM: DOT BELOW 323

```
 0x68a, 0x694, 0x6a3, 0x6b9, 0x6fa, 0x6fb, 0x6fc, 0x766, 0x88b,
 0x8a5, 0x8b4, 0x1e04, 0x1e05, 0x1e0c, 0x1e0d, 0x1e24, 0x1e25,
 0x1e32, 0x1e33, 0x1e36, 0x1e37, 0x1e42, 0x1e43, 0x1e46, 0x1e47,
 0x1e5a, 0x1e5b, 0x1e62, 0x1e63, 0x1e6c, 0x1e6d, 0x1e7e, 0x1e7f,
 0x1e88, 0x1e89, 0x1e92, 0x1e93, 0x1ea0, 0x1ea1, 0x1eb8, 0x1eb9,
 0x1eca, 0x1ecb, 0x1ecc, 0x1ecd, 0x1ee4, 0x1ee5, 0x1ef4, 0x1ef5,
  0x1bc26
```

NSM: DOUBLE DOT BELOW 324

```
  None
```

NSM: RING BELOW 325

```
  0x1e00, 0x1e01
```

NSM: COMMA BELOW 326

```
  0x218, 0x219, 0x21a, 0x21b
```

NSM: CEDILLA 327

```
 0xc7, 0xe7, 0x122, 0x123, 0x136, 0x137, 0x13b, 0x13c, 0x145, 0x146,
 0x156, 0x157, 0x15e, 0x15f, 0x162, 0x163, 0x228, 0x229, 0x1e10,
  0x1e11, 0x1e28, 0x1e29
```

NSM: OGONEK 328

```
 0x104, 0x105, 0x118, 0x119, 0x12e, 0x12f, 0x172, 0x173, 0x1ea, 0x1eb
```

NSM: VERTICAL LINE BELOW 329 NSM: BRIDGE BELOW 32a NSM: INVERTED DOUBLE ARCH BELOW 32b NSM: HACEK BELOW 32c

None

NSM: CIRCUMFLEX BELOW 32d

0x1e12, 0x1e13, 0x1e18, 0x1e19, 0x1e3c, 0x1e3d, 0x1e4a, 0x1e4b, 0x1e70, 0x1e71, 0x1e76, 0x1e77

NSM: BREVE BELOW 32e

0x1e2a, 0x1e2b

NSM: INVERTED BREVE BELOW 32f

None

NSM: TILDE BELOW 330

0x1e1a, 0x1e1b, 0x1e2c, 0x1e2d, 0x1e74, 0x1e75

NSM: MACRON BELOW 331 NSM: UNDERSCORE 332 NSM: DOUBLE UNDERSCORE 333 NSM: TILDE OVERLAY 334 NSM: SHORT BAR OVERLAY 335 NSM: LONG BAR OVERLAY 336 NSM: SHORT SLASH OVERLAY 337 NSM: LONG SLASH OVERLAY 338 NSM: RIGHT HALF RING BELOW 339 NSM: INVERTED BRIDGE BELOW 33a NSM: SQUARE BELOW 33b NSM: SEAGULL BELOW 33c NSM: X ABOVE 33d NSM: VERTICAL TILDE 33e NSM: DOUBLE OVERSCORE 33f NSM: GRAVE TONE MARK 340 NSM: ACUTE TONE MARK 341 NSM: LEFT HARPOON ABOVE 20d0 NSM: RIGHT HARPOON ABOVE 20d1 NSM: LONG VERTICAL BAR OVERLAY 20d2 NSM: SHORT VERTICAL BAR OVERLAY 20d3 NSM: ANTICLOCKWISE ARROW ABOVE 20d4 NSM: CLOCKWISE ARROW ABOVE 20d5 NSM: LEFT ARROW ABOVE 20d6 NSM: RIGHT ARROW ABOVE 20d7 NSM: RING OVERLAY 20d8 NSM: CLOCKWISE RING OVERLAY 20d9 NSM: ANTICLOCKWISE RING OVERLAY 20da

None

NSM: THREE DOTS ABOVE 20db

0x63f, 0x685, 0x69e, 0x69f, 0x6a0, 0x6a8, 0x6b4, 0x6b7, 0x6bd, 0x763, 0x8a7, 0x8c3, 0x8c4, 0x8c5

NSM: FOUR DOTS ABOVE 20dc

0x690, 0x699, 0x75c

NSM: LEFT RIGHT ARROW ABOVE 20e1

None

NSM: KATAKANA-HIRAGANA VOICED SOUND MARK 3099

Todo

NSM: KATAKANA-HIRAGANA SEMI-VOICED SOUND MARK 309a

Todo

21 References

[AltId] Unicode Standard Annex http://www.unicode.org/reports/tr31/tr31-11.html#Alternative_Identifier_Syntax

[DefId] Unicode Standard Annex. http://www.unicode.org/reports/tr31/tr31-11.html#Default_Identifier_Syntax

[ISO 15924 Codes] TR24 Unicode Script Property Values and ISO 15924 Codes. https://www.unicode.org/reports/tr24/#Relation_To_ISO15924

[libu8ident] Reini Urban. 2022. Unicode security guidelines for identifiers. https://github.com/rurban/libu8ident/

[N3146] Clark Nelson. 2010. Recommendations for extended identifier characters for C and C++.

: https://wg21.link/n3146

[P1949] Steve Downey et al. 2021. C++ Identifier Syntax using Unicode Standard Annex 31.

: http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2021/p1949r7.html

[TR15] Ken Whistler. Unicode Normalization Forms. http://www.unicode.org/reports/tr15

[TR24] Ken Whistler. Unicode Script Property. https://www.unicode.org/reports/tr24/#Common

[TR24#5.1] Handling Characters with the Common Script Property. https://www.unicode.org/reports/tr24/#Common

[TR24#5.2] Handling Combining Marks. https://www.unicode.org/reports/tr24/#Nonspacing_Marks

[TR31] Mark Davis. Unicode Identifier and Pattern Syntax. http://www.unicode.org/reports/tr31

[TR31#2.1] Combining Marks. https://www.unicode.org/reports/tr31/#Combining_Marks

[TR31#2.2] Modifier Letters. https://www.unicode.org/reports/tr31/#Modifier_Letters

[TR31#Table 4] Table Candidate Characters for Exclusion from Identifiers. https://www.unicode.org/reports/tr31/#Table_Candidate_Characters_for_Exclusion_from_Identifiers

[TR31#Table 7] Limited Use Scripts. http://www.unicode.org/reports/tr31/#Table_Limited_Use_Scripts

[TR36] Mark Davis and Michel Suignard. Unicode Security Considerations. http://www.unicode.org/reports/tr36

[TR39] Mark Davis and Michel Suignard. Unicode Security Mechanisms. http://www.unicode.org/reports/tr36

[TR39#Table 1] Identifier Status and Type Table 1.

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

[TR39#4] Confusable Detection. <https://www.unicode.org/reports/tr39/#Confusable Detection>

[TR39#5.2] Mixed-Scripts Restriction-Level Detection. https://www.unicode.org/reports/tr39/#Restriction_Level_Detection

[TR39#5.4] Optional Detection. https://www.unicode.org/reports/tr39/#Optional_Detection

[TR44] Ken Whistler and Laurențiu Iancu. Unicode Character Database. http://www.unicode.org/reports/tr44

[TR46] Mark Davis and Michel Suignard. Unicode IDNA Compatibility Processing. http://www.unicode.org/reports/tr46