!!!!!!! DO NOT EDIT THIS FILE !!!!!!! # This file is machine-generated by lib/unicore/mktables from the Unicode # database, Version 6.3.0. Any changes made here will be lost!

To change this file, edit lib/unicore/mktables instead.

NAME

perluniprops - Index of Unicode Version 6.3.0 character properties in Perl

DESCRIPTION

This document provides information about the portion of the Unicode database that deals with character properties, that is the portion that is defined on single code points. (*Other information in the Unicode data base* below briefly mentions other data that Unicode provides.)

Perl can provide access to all non-provisional Unicode character properties, though not all are enabled by default. The omitted ones are the Unihan properties (accessible via the CPAN module *Unicode::Unihan*) and certain deprecated or Unicode-internal properties. (An installation may choose to recompile Perl's tables to change this. See *Unicode character properties that are NOT accepted by Perl*.)

For most purposes, access to Unicode properties from the Perl core is through regular expression matches, as described in the next section. For some special purposes, and to access the properties that are not suitable for regular expression matching, all the Unicode character properties that Perl handles are accessible via the standard *Unicode::UCD* module, as described in the section *Properties accessible through Unicode::UCD*.

Perl also provides some additional extensions and short-cut synonyms for Unicode properties.

This document merely lists all available properties and does not attempt to explain what each property really means. There is a brief description of each Perl extension; see *"Other Properties" in perlunicode* for more information on these. There is some detail about Blocks, Scripts, General_Category, and Bidi_Class in *perlunicode*, but to find out about the intricacies of the official Unicode properties, refer to the Unicode standard. A good starting place is *http://www.unicode.org/reports/tr44/*.

Note that you can define your own properties; see *"User-Defined Character Properties" in perlunicode* .

Properties accessible through \p{} and \P{}

The Perl regular expression `\p{}` and `\P{}` constructs give access to most of the Unicode character properties. The table below shows all these constructs, both single and compound forms.

Compound forms consist of two components, separated by an equals sign or a colon. The first component is the property name, and the second component is the particular value of the property to match against, for example, `\p{Script: Greek}` and `\p{Script=Greek}` both mean to match characters whose Script property value is Greek.

Single forms, like `\p{Greek}`, are mostly Perl-defined shortcuts for their equivalent compound forms. The table shows these equivalences. (In our example, `\p{Greek}` is a just a shortcut for `\p{Script=Greek}`.) There are also a few Perl-defined single forms that are not shortcuts for a compound form. One such is `\p{Word}`. These are also listed in the table.

In parsing these constructs, Perl always ignores Upper/lower case differences everywhere within the {braces}. Thus `\p{Greek}` means the same thing as `\p{greek}`. But note that changing the case of the `"p"` or `"P"` before the left brace completely changes the meaning of the construct, from "match" (for `\p{}`) to "doesn't match" (for `\P{}`). Casing in this document is for improved legibility.

Also, white space, hyphens, and underscores are normally ignored everywhere between the {braces}, and hence can be freely added or removed even if the `/x` modifier hasn't been specified on the regular expression. But in the table below a '**T**' at the beginning of an entry means that tighter (stricter)

rules are used for that entry:

Single form (`\p{name}`) tighter rules:

White space, hyphens, and underscores ARE significant except for:

* white space adjacent to a non-word character

* underscores separating digits in numbers

That means, for example, that you can freely add or remove white space adjacent to (but within) the braces without affecting the meaning.

Compound form (`\p{name=value}` or `\p{name:value}`) tighter rules:

The tighter rules given above for the single form apply to everything to the right of the colon or equals; the looser rules still apply to everything to the left.

That means, for example, that you can freely add or remove white space adjacent to (but within) the braces and the colon or equal sign.

Some properties are considered obsolete by Unicode, but still available. There are several varieties of obsolescence:

Stabilized

A property may be stabilized. Such a determination does not indicate that the property should or should not be used; instead it is a declaration that the property will not be maintained nor extended for newly encoded characters. Such properties are marked with an '**S**' in the table.

Deprecated

A property may be deprecated, perhaps because its original intent has been replaced by another property, or because its specification was somehow defective. This means that its use is strongly discouraged, so much so that a warning will be issued if used, unless the regular expression is in the scope of a `no warnings 'deprecated'` statement. A '**D**' flags each such entry in the table, and the entry there for the longest, most descriptive version of the property will give the reason it is deprecated, and perhaps advice. Perl may issue such a warning, even for properties that aren't officially deprecated by Unicode, when there used to be characters or code points that were matched by them, but no longer. This is to warn you that your program may not work like it did on earlier Unicode releases.

A deprecated property may be made unavailable in a future Perl version, so it is best to move away from them.

A deprecated property may also be stabilized, but this fact is not shown.

Obsolete

Properties marked with an '**O**' in the table are considered (plain) obsolete. Generally this designation is given to properties that Unicode once used for internal purposes (but not any longer).

Some Perl extensions are present for backwards compatibility and are discouraged from being used, but are not obsolete. An '**X**' flags each such entry in the table. Future Unicode versions may force some of these extensions to be removed without warning, replaced by another property with the same name that means something different. Use the equivalent shown instead.

Matches in the Block property have shortcuts that begin with "In_". For example, `\p{Block=Latin1}` can be written as `\p{In_Latin1}`. For backward compatibility, if there is no conflict with another shortcut, these may also be written as `\p{Latin1}` or `\p{Is_Latin1}`. But, N.B., there are numerous such conflicting shortcuts. Use of these forms for Block is discouraged, and

are flagged as such, not only because of the potential confusion as to what is meant, but also because a later release of Unicode may preempt the shortcut, and your program would no longer be correct. Use the "In_" form instead to avoid this, or even more clearly, use the compound form, e.g., `\p{blk:latin1}`. See *"Blocks" in perlunicode* for more information about this.

The table below has two columns. The left column contains the `\p{}` constructs to look up, possibly preceded by the flags mentioned above; and the right column contains information about them, like a description, or synonyms. The table shows both the single and compound forms for each property that has them. If the left column is a short name for a property, the right column will give its longer, more descriptive name; and if the left column is the longest name, the right column will show any equivalent shortest name, in both single and compound forms if applicable.

If braces are not needed to specify a property (e.g., `\pL`), the left column contains both forms, with and without braces.

The right column will also caution you if a property means something different than what might normally be expected.

All single forms are Perl extensions; a few compound forms are as well, and are noted as such.

Numbers in (parentheses) indicate the total number of Unicode code points matched by the property. For emphasis, those properties that match no code points at all are listed as well in a separate section following the table.

Most properties match the same code points regardless of whether `"/i"` case-insensitive matching is specified or not. But a few properties are affected. These are shown with the notation (`/i=` *`other_property`*`)` in the second column. Under case-insensitive matching they match the same code pode points as the property *other_property*.

There is no description given for most non-Perl defined properties (See *http://www.unicode.org/reports/tr44/* for that).

For compactness, '*****' is used as a wildcard instead of showing all possible combinations. For example, entries like:

```
\p{Gc: *}                                   \p{General_Category: *}
```

mean that 'Gc' is a synonym for 'General_Category', and anything that is valid for the latter is also valid for the former. Similarly,

```
\p{Is_*}                                    \p{*}
```

means that if and only if, for example, `\p{Foo}` exists, then `\p{Is_Foo}` and `\p{IsFoo}` are also valid and all mean the same thing. And similarly, `\p{Foo=Bar}` means the same as `\p{Is_Foo=Bar}` and `\p{IsFoo=Bar}`. "*" here is restricted to something not beginning with an underscore.

Also, in binary properties, 'Yes', 'T', and 'True' are all synonyms for 'Y'. And 'No', 'F', and 'False' are all synonyms for 'N'. The table shows 'Y*' and 'N*' to indicate this, and doesn't have separate entries for the other possibilities. Note that not all properties which have values 'Yes' and 'No' are binary, and they have all their values spelled out without using this wild card, and a `NOT` clause in their description that highlights their not being binary. These also require the compound form to match them, whereas true binary properties have both single and compound forms available.

Note that all non-essential underscores are removed in the display of the short names below.

Legend summary:

- ***** is a wild-card

- **(\d+)** in the info column gives the number of Unicode code points matched by this property.
- **D** means this is deprecated.
- **O** means this is obsolete.
- **S** means this is stabilized.
- **T** means tighter (stricter) name matching applies.
- **X** means use of this form is discouraged, and may not be stable.

```
   NAME                          INFO

X \p{Aegean_Numbers}       \p{Block=Aegean_Numbers} (64)
T \p{Age: 1.1}             \p{Age=V1_1} (33_979)
T \p{Age: 2.0}             \p{Age=V2_0} (144_521)
T \p{Age: 2.1}             \p{Age=V2_1} (2)
T \p{Age: 3.0}             \p{Age=V3_0} (10_307)
T \p{Age: 3.1}             \p{Age=V3_1} (44_978)
T \p{Age: 3.2}             \p{Age=V3_2} (1016)
T \p{Age: 4.0}             \p{Age=V4_0} (1226)
T \p{Age: 4.1}             \p{Age=V4_1} (1273)
T \p{Age: 5.0}             \p{Age=V5_0} (1369)
T \p{Age: 5.1}             \p{Age=V5_1} (1624)
T \p{Age: 5.2}             \p{Age=V5_2} (6648)
T \p{Age: 6.0}             \p{Age=V6_0} (2088)
T \p{Age: 6.1}             \p{Age=V6_1} (732)
T \p{Age: 6.2}             \p{Age=V6_2} (1)
T \p{Age: 6.3}             \p{Age=V6_3} (5)
  \p{Age: NA}              \p{Age=Unassigned} (864_343 plus all
                             above-Unicode code points)
  \p{Age: Unassigned}      Code point's usage has not been assigned
                             in any Unicode release thus far. (Short:
                             \p{Age=NA}) (864_343 plus all above-
                             Unicode code points)
  \p{Age: V1_1}            Code point's usage introduced in version
                             1.1 (33_979)
  \p{Age: V2_0}            Code point's usage was introduced in
                             version 2.0; See also Property
                             'Present_In' (144_521)
  \p{Age: V2_1}            Code point's usage was introduced in
                             version 2.1; See also Property
                             'Present_In' (2)
  \p{Age: V3_0}            Code point's usage was introduced in
                             version 3.0; See also Property
                             'Present_In' (10_307)
  \p{Age: V3_1}            Code point's usage was introduced in
                             version 3.1; See also Property
                             'Present_In' (44_978)
  \p{Age: V3_2}            Code point's usage was introduced in
                             version 3.2; See also Property
                             'Present_In' (1016)
  \p{Age: V4_0}            Code point's usage was introduced in
                             version 4.0; See also Property
                             'Present_In' (1226)
  \p{Age: V4_1}            Code point's usage was introduced in
                             version 4.1; See also Property
```

```
                                'Present_In' (1273)
    \p{Age: V5_0}           Code point's usage was introduced in
                              version 5.0; See also Property
                              'Present_In' (1369)
    \p{Age: V5_1}           Code point's usage was introduced in
                              version 5.1; See also Property
                              'Present_In' (1624)
    \p{Age: V5_2}           Code point's usage was introduced in
                              version 5.2; See also Property
                              'Present_In' (6648)
    \p{Age: V6_0}           Code point's usage was introduced in
                              version 6.0; See also Property
                              'Present_In' (2088)
    \p{Age: V6_1}           Code point's usage was introduced in
                              version 6.1; See also Property
                              'Present_In' (732)
    \p{Age: V6_2}           Code point's usage was introduced in
                              version 6.2; See also Property
                              'Present_In' (1)
    \p{Age: V6_3}           Code point's usage was introduced in
                              version 6.3; See also Property
                              'Present_In' (5)
    \p{AHex}                \p{PosixXDigit} (= \p{ASCII_Hex_Digit=Y})
                              (22)
    \p{AHex: *}             \p{ASCII_Hex_Digit: *}
  X \p{Alchemical}          \p{Alchemical_Symbols} (= \p{Block=
                              Alchemical_Symbols}) (128)
  X \p{Alchemical_Symbols}  \p{Block=Alchemical_Symbols} (Short:
                              \p{InAlchemical}) (128)
    \p{All}                 All code points, including those above
                              Unicode.  Same as qr/./s (1_114_112 plus
                              all above-Unicode code points)
    \p{Alnum}               Alphabetic and (decimal) Numeric (102_619)
    \p{Alpha}               \p{Alphabetic=Y} (102_159)
    \p{Alpha: *}            \p{Alphabetic: *}
    \p{Alphabetic}          \p{Alpha} (= \p{Alphabetic=Y}) (102_159)
    \p{Alphabetic: N*}      (Short: \p{Alpha=N}, \P{Alpha}) (1_011_953
                              plus all above-Unicode code points)
    \p{Alphabetic: Y*}      (Short: \p{Alpha=Y}, \p{Alpha}) (102_159)
  X \p{Alphabetic_PF}       \p{Alphabetic_Presentation_Forms} (=
                              \p{Block=Alphabetic_Presentation_Forms})
                              (80)
  X \p{Alphabetic_Presentation_Forms} \p{Block=
                              Alphabetic_Presentation_Forms} (Short:
                              \p{InAlphabeticPF}) (80)
  X \p{Ancient_Greek_Music} \p{Ancient_Greek_Musical_Notation} (=
                              \p{Block=
                              Ancient_Greek_Musical_Notation}) (80)
  X \p{Ancient_Greek_Musical_Notation} \p{Block=
                              Ancient_Greek_Musical_Notation} (Short:
                              \p{InAncientGreekMusic}) (80)
  X \p{Ancient_Greek_Numbers} \p{Block=Ancient_Greek_Numbers} (80)
  X \p{Ancient_Symbols}     \p{Block=Ancient_Symbols} (64)
    \p{Any}                 All Unicode code points: [\x{0000}-
                              \x{10FFFF}] (1_114_112)
    \p{Arab}                \p{Arabic} (= \p{Script=Arabic}) (NOT
```

```
                                \p{Block=Arabic}) (1236)
  \p{Arabic}                  \p{Script=Arabic} (Short: \p{Arab}; NOT
                                \p{Block=Arabic}) (1236)
X \p{Arabic_Ext_A}            \p{Arabic_Extended_A} (= \p{Block=
                                Arabic_Extended_A}) (96)
X \p{Arabic_Extended_A}       \p{Block=Arabic_Extended_A} (Short:
                                \p{InArabicExtA}) (96)
X \p{Arabic_Math}             \p{Arabic_Mathematical_Alphabetic_Symbols}
                                (= \p{Block=
                                Arabic_Mathematical_Alphabetic_Symbols})
                                (256)
X \p{Arabic_Mathematical_Alphabetic_Symbols} \p{Block=
                                Arabic_Mathematical_Alphabetic_Symbols}
                                (Short: \p{InArabicMath}) (256)
X \p{Arabic_PF_A}             \p{Arabic_Presentation_Forms_A} (=
                                \p{Block=Arabic_Presentation_Forms_A})
                                (688)
X \p{Arabic_PF_B}             \p{Arabic_Presentation_Forms_B} (=
                                \p{Block=Arabic_Presentation_Forms_B})
                                (144)
X \p{Arabic_Presentation_Forms_A} \p{Block=
                                Arabic_Presentation_Forms_A} (Short:
                                \p{InArabicPFA}) (688)
X \p{Arabic_Presentation_Forms_B} \p{Block=
                                Arabic_Presentation_Forms_B} (Short:
                                \p{InArabicPFB}) (144)
X \p{Arabic_Sup}              \p{Arabic_Supplement} (= \p{Block=
                                Arabic_Supplement}) (48)
X \p{Arabic_Supplement}       \p{Block=Arabic_Supplement} (Short:
                                \p{InArabicSup}) (48)
  \p{Armenian}                \p{Script=Armenian} (Short: \p{Armn}; NOT
                                \p{Block=Armenian}) (91)
  \p{Armi}                    \p{Imperial_Aramaic} (= \p{Script=
                                Imperial_Aramaic}) (NOT \p{Block=
                                Imperial_Aramaic}) (31)
  \p{Armn}                    \p{Armenian} (= \p{Script=Armenian}) (NOT
                                \p{Block=Armenian}) (91)
X \p{Arrows}                  \p{Block=Arrows} (112)
  \p{ASCII}                   \p{Block=Basic_Latin} [[:ASCII:]] (128)
  \p{ASCII_Hex_Digit}         \p{PosixXDigit} (= \p{ASCII_Hex_Digit=Y})
                                (22)
  \p{ASCII_Hex_Digit: N*} (Short: \p{AHex=N}, \P{AHex}) (1_114_090
                                plus all above-Unicode code points)
  \p{ASCII_Hex_Digit: Y*} (Short: \p{AHex=Y}, \p{AHex}) (22)
  \p{Assigned}                All assigned code points (249_703)
  \p{Avestan}                 \p{Script=Avestan} (Short: \p{Avst}; NOT
                                \p{Block=Avestan}) (61)
  \p{Avst}                    \p{Avestan} (= \p{Script=Avestan}) (NOT
                                \p{Block=Avestan}) (61)
  \p{Bali}                    \p{Balinese} (= \p{Script=Balinese}) (NOT
                                \p{Block=Balinese}) (121)
  \p{Balinese}                \p{Script=Balinese} (Short: \p{Bali}; NOT
                                \p{Block=Balinese}) (121)
  \p{Bamu}                    \p{Bamum} (= \p{Script=Bamum}) (NOT
                                \p{Block=Bamum}) (657)
  \p{Bamum}                   \p{Script=Bamum} (Short: \p{Bamu}; NOT
```

```
                                \p{Block=Bamum}) (657)
X \p{Bamum_Sup}                 \p{Bamum_Supplement} (= \p{Block=
                                Bamum_Supplement}) (576)
X \p{Bamum_Supplement}          \p{Block=Bamum_Supplement} (Short:
                                \p{InBamumSup}) (576)
X \p{Basic_Latin}               \p{ASCII} (= \p{Block=Basic_Latin}) (128)
  \p{Batak}                     \p{Script=Batak} (Short: \p{Batk}; NOT
                                \p{Block=Batak}) (56)
  \p{Batk}                      \p{Batak} (= \p{Script=Batak}) (NOT
                                \p{Block=Batak}) (56)
  \p{Bc: *}                     \p{Bidi_Class: *}
  \p{Beng}                      \p{Bengali} (= \p{Script=Bengali}) (NOT
                                \p{Block=Bengali}) (92)
  \p{Bengali}                   \p{Script=Bengali} (Short: \p{Beng}; NOT
                                \p{Block=Bengali}) (92)
  \p{Bidi_C}                    \p{Bidi_Control} (= \p{Bidi_Control=Y})
                                (12)
  \p{Bidi_C: *}                 \p{Bidi_Control: *}
  \p{Bidi_Class: AL}            \p{Bidi_Class=Arabic_Letter} (1438)
  \p{Bidi_Class: AN}            \p{Bidi_Class=Arabic_Number} (49)
  \p{Bidi_Class: Arabic_Letter} (Short: \p{Bc=AL}) (1438)
  \p{Bidi_Class: Arabic_Number} (Short: \p{Bc=AN}) (49)
  \p{Bidi_Class: B}             \p{Bidi_Class=Paragraph_Separator} (7)
  \p{Bidi_Class: BN}            \p{Bidi_Class=Boundary_Neutral} (4012)
  \p{Bidi_Class: Boundary_Neutral} (Short: \p{Bc=BN}) (4012)
  \p{Bidi_Class: Common_Separator} (Short: \p{Bc=CS}) (15)
  \p{Bidi_Class: CS}            \p{Bidi_Class=Common_Separator} (15)
  \p{Bidi_Class: EN}            \p{Bidi_Class=European_Number} (131)
  \p{Bidi_Class: ES}            \p{Bidi_Class=European_Separator} (12)
  \p{Bidi_Class: ET}            \p{Bidi_Class=European_Terminator} (87)
  \p{Bidi_Class: European_Number} (Short: \p{Bc=EN}) (131)
  \p{Bidi_Class: European_Separator} (Short: \p{Bc=ES}) (12)
  \p{Bidi_Class: European_Terminator} (Short: \p{Bc=ET}) (87)
  \p{Bidi_Class: First_Strong_Isolate} (Short: \p{Bc=FSI}) (1)
  \p{Bidi_Class: FSI}           \p{Bidi_Class=First_Strong_Isolate} (1)
  \p{Bidi_Class: L}             \p{Bidi_Class=Left_To_Right} (1_098_508
                                plus all above-Unicode code points)
  \p{Bidi_Class: Left_To_Right} (Short: \p{Bc=L}) (1_098_508 plus
                                all above-Unicode code points)
  \p{Bidi_Class: Left_To_Right_Embedding} (Short: \p{Bc=LRE}) (1)
  \p{Bidi_Class: Left_To_Right_Isolate} (Short: \p{Bc=LRI}) (1)
  \p{Bidi_Class: Left_To_Right_Override} (Short: \p{Bc=LRO}) (1)
  \p{Bidi_Class: LRE}           \p{Bidi_Class=Left_To_Right_Embedding} (1)
  \p{Bidi_Class: LRI}           \p{Bidi_Class=Left_To_Right_Isolate} (1)
  \p{Bidi_Class: LRO}           \p{Bidi_Class=Left_To_Right_Override} (1)
  \p{Bidi_Class: Nonspacing_Mark} (Short: \p{Bc=NSM}) (1291)
  \p{Bidi_Class: NSM}           \p{Bidi_Class=Nonspacing_Mark} (1291)
  \p{Bidi_Class: ON}            \p{Bidi_Class=Other_Neutral} (4447)
  \p{Bidi_Class: Other_Neutral} (Short: \p{Bc=ON}) (4447)
  \p{Bidi_Class: Paragraph_Separator} (Short: \p{Bc=B}) (7)
  \p{Bidi_Class: PDF}           \p{Bidi_Class=Pop_Directional_Format} (1)
  \p{Bidi_Class: PDI}           \p{Bidi_Class=Pop_Directional_Isolate} (1)
  \p{Bidi_Class: Pop_Directional_Format} (Short: \p{Bc=PDF}) (1)
  \p{Bidi_Class: Pop_Directional_Isolate} (Short: \p{Bc=PDI}) (1)
  \p{Bidi_Class: R}             \p{Bidi_Class=Right_To_Left} (4086)
  \p{Bidi_Class: Right_To_Left} (Short: \p{Bc=R}) (4086)
```

```
\p{Bidi_Class: Right_To_Left_Embedding} (Short: \p{Bc=RLE}) (1)
\p{Bidi_Class: Right_To_Left_Isolate} (Short: \p{Bc=RLI}) (1)
\p{Bidi_Class: Right_To_Left_Override} (Short: \p{Bc=RLO}) (1)
\p{Bidi_Class: RLE}     \p{Bidi_Class=Right_To_Left_Embedding} (1)
\p{Bidi_Class: RLI}     \p{Bidi_Class=Right_To_Left_Isolate} (1)
\p{Bidi_Class: RLO}     \p{Bidi_Class=Right_To_Left_Override} (1)
\p{Bidi_Class: S}       \p{Bidi_Class=Segment_Separator} (3)
\p{Bidi_Class: Segment_Separator} (Short: \p{Bc=S}) (3)
\p{Bidi_Class: White_Space} (Short: \p{Bc=WS}) (17)
\p{Bidi_Class: WS}      \p{Bidi_Class=White_Space} (17)
\p{Bidi_Control}        \p{Bidi_Control=Y} (Short: \p{BidiC}) (12)
\p{Bidi_Control: N*}    (Short: \p{BidiC=N}, \P{BidiC}) (1_114_100
                          plus all above-Unicode code points)
\p{Bidi_Control: Y*}    (Short: \p{BidiC=Y}, \p{BidiC}) (12)
\p{Bidi_M}              \p{Bidi_Mirrored} (= \p{Bidi_Mirrored=Y})
                          (545)
\p{Bidi_M: *}           \p{Bidi_Mirrored: *}
\p{Bidi_Mirrored}       \p{Bidi_Mirrored=Y} (Short: \p{BidiM})
                          (545)
\p{Bidi_Mirrored: N*}   (Short: \p{BidiM=N}, \P{BidiM}) (1_113_567
                          plus all above-Unicode code points)
\p{Bidi_Mirrored: Y*}   (Short: \p{BidiM=Y}, \p{BidiM}) (545)
\p{Bidi_Paired_Bracket_Type: C} \p{Bidi_Paired_Bracket_Type=Close}
                          (60)
\p{Bidi_Paired_Bracket_Type: Close} (Short: \p{Bpt=C}) (60)
\p{Bidi_Paired_Bracket_Type: N} \p{Bidi_Paired_Bracket_Type=None}
                          (1_113_992 plus all above-Unicode code
                          points)
\p{Bidi_Paired_Bracket_Type: None} (Short: \p{Bpt=N}) (1_113_992
                          plus all above-Unicode code points)
\p{Bidi_Paired_Bracket_Type: O} \p{Bidi_Paired_Bracket_Type=Open}
                          (60)
\p{Bidi_Paired_Bracket_Type: Open} (Short: \p{Bpt=O}) (60)
\p{Blank}               \h, Horizontal white space (18)
\p{Blk: *}              \p{Block: *}
\p{Block: Aegean_Numbers} (Single: \p{InAegeanNumbers}) (64)
\p{Block: Alchemical}   \p{Block=Alchemical_Symbols} (128)
\p{Block: Alchemical_Symbols} (Short: \p{Blk=Alchemical},
                          \p{InAlchemical}) (128)
\p{Block: Alphabetic_PF} \p{Block=Alphabetic_Presentation_Forms}
                          (80)
\p{Block: Alphabetic_Presentation_Forms} (Short: \p{Blk=
                          AlphabeticPF}, \p{InAlphabeticPF}) (80)
\p{Block: Ancient_Greek_Music} \p{Block=
                          Ancient_Greek_Musical_Notation} (80)
\p{Block: Ancient_Greek_Musical_Notation} (Short: \p{Blk=
                          AncientGreekMusic},
                          \p{InAncientGreekMusic}) (80)
\p{Block: Ancient_Greek_Numbers} (Single:
                          \p{InAncientGreekNumbers}) (80)
\p{Block: Ancient_Symbols} (Single: \p{InAncientSymbols}) (64)
\p{Block: Arabic}       (Single: \p{InArabic}; NOT \p{Arabic} NOR
                          \p{Is_Arabic}) (256)
\p{Block: Arabic_Ext_A} \p{Block=Arabic_Extended_A} (96)
\p{Block: Arabic_Extended_A} (Short: \p{Blk=ArabicExtA},
                          \p{InArabicExtA}) (96)
```

```
\p{Block: Arabic_Math}  \p{Block=
                          Arabic_Mathematical_Alphabetic_Symbols}
                          (256)
\p{Block: Arabic_Mathematical_Alphabetic_Symbols} (Short: \p{Blk=
                          ArabicMath}, \p{InArabicMath}) (256)
\p{Block: Arabic_PF_A}  \p{Block=Arabic_Presentation_Forms_A} (688)
\p{Block: Arabic_PF_B}  \p{Block=Arabic_Presentation_Forms_B} (144)
\p{Block: Arabic_Presentation_Forms_A} (Short: \p{Blk=ArabicPFA},
                          \p{InArabicPFA}) (688)
\p{Block: Arabic_Presentation_Forms_B} (Short: \p{Blk=ArabicPFB},
                          \p{InArabicPFB}) (144)
\p{Block: Arabic_Sup}   \p{Block=Arabic_Supplement} (48)
\p{Block: Arabic_Supplement} (Short: \p{Blk=ArabicSup},
                          \p{InArabicSup}) (48)
\p{Block: Armenian}     (Single: \p{InArmenian}; NOT \p{Armenian}
                          NOR \p{Is_Armenian}) (96)
\p{Block: Arrows}       (Single: \p{InArrows}) (112)
\p{Block: ASCII}        \p{Block=Basic_Latin} (128)
\p{Block: Avestan}      (Single: \p{InAvestan}; NOT \p{Avestan}
                          NOR \p{Is_Avestan}) (64)
\p{Block: Balinese}     (Single: \p{InBalinese}; NOT \p{Balinese}
                          NOR \p{Is_Balinese}) (128)
\p{Block: Bamum}        (Single: \p{InBamum}; NOT \p{Bamum} NOR
                          \p{Is_Bamum}) (96)
\p{Block: Bamum_Sup}    \p{Block=Bamum_Supplement} (576)
\p{Block: Bamum_Supplement} (Short: \p{Blk=BamumSup},
                          \p{InBamumSup}) (576)
\p{Block: Basic_Latin}  (Short: \p{Blk=ASCII}, \p{ASCII}) (128)
\p{Block: Batak}        (Single: \p{InBatak}; NOT \p{Batak} NOR
                          \p{Is_Batak}) (64)
\p{Block: Bengali}      (Single: \p{InBengali}; NOT \p{Bengali}
                          NOR \p{Is_Bengali}) (128)
\p{Block: Block_Elements} (Single: \p{InBlockElements}) (32)
\p{Block: Bopomofo}     (Single: \p{InBopomofo}; NOT \p{Bopomofo}
                          NOR \p{Is_Bopomofo}) (48)
\p{Block: Bopomofo_Ext} \p{Block=Bopomofo_Extended} (32)
\p{Block: Bopomofo_Extended} (Short: \p{Blk=BopomofoExt},
                          \p{InBopomofoExt}) (32)
\p{Block: Box_Drawing}  (Single: \p{InBoxDrawing}) (128)
\p{Block: Brahmi}       (Single: \p{InBrahmi}; NOT \p{Brahmi} NOR
                          \p{Is_Brahmi}) (128)
\p{Block: Braille}      \p{Block=Braille_Patterns} (256)
\p{Block: Braille_Patterns} (Short: \p{Blk=Braille},
                          \p{InBraille}) (256)
\p{Block: Buginese}     (Single: \p{InBuginese}; NOT \p{Buginese}
                          NOR \p{Is_Buginese}) (32)
\p{Block: Buhid}        (Single: \p{InBuhid}; NOT \p{Buhid} NOR
                          \p{Is_Buhid}) (32)
\p{Block: Byzantine_Music} \p{Block=Byzantine_Musical_Symbols}
                          (256)
\p{Block: Byzantine_Musical_Symbols} (Short: \p{Blk=
                          ByzantineMusic}, \p{InByzantineMusic})
                          (256)
\p{Block: Canadian_Syllabics} \p{Block=
                          Unified_Canadian_Aboriginal_Syllabics}
                          (640)
```

```
\p{Block: Carian}       (Single: \p{InCarian}; NOT \p{Carian} NOR
                          \p{Is_Carian}) (64)
\p{Block: Chakma}       (Single: \p{InChakma}; NOT \p{Chakma} NOR
                          \p{Is_Chakma}) (80)
\p{Block: Cham}         (Single: \p{InCham}; NOT \p{Cham} NOR
                          \p{Is_Cham}) (96)
\p{Block: Cherokee}     (Single: \p{InCherokee}; NOT \p{Cherokee}
                          NOR \p{Is_Cherokee}) (96)
\p{Block: CJK}          \p{Block=CJK_Unified_Ideographs} (20_992)
\p{Block: CJK_Compat}   \p{Block=CJK_Compatibility} (256)
\p{Block: CJK_Compat_Forms} \p{Block=CJK_Compatibility_Forms} (32)
\p{Block: CJK_Compat_Ideographs} \p{Block=
                          CJK_Compatibility_Ideographs} (512)
\p{Block: CJK_Compat_Ideographs_Sup} \p{Block=
                          CJK_Compatibility_Ideographs_Supplement}
                          (544)
\p{Block: CJK_Compatibility} (Short: \p{Blk=CJKCompat},
                          \p{InCJKCompat}) (256)
\p{Block: CJK_Compatibility_Forms} (Short: \p{Blk=CJKCompatForms},
                          \p{InCJKCompatForms}) (32)
\p{Block: CJK_Compatibility_Ideographs} (Short: \p{Blk=
                          CJKCompatIdeographs},
                          \p{InCJKCompatIdeographs}) (512)
\p{Block: CJK_Compatibility_Ideographs_Supplement} (Short: \p{Blk=
                          CJKCompatIdeographsSup},
                          \p{InCJKCompatIdeographsSup}) (544)
\p{Block: CJK_Ext_A}    \p{Block=
                          CJK_Unified_Ideographs_Extension_A}
                          (6592)
\p{Block: CJK_Ext_B}    \p{Block=
                          CJK_Unified_Ideographs_Extension_B}
                          (42_720)
\p{Block: CJK_Ext_C}    \p{Block=
                          CJK_Unified_Ideographs_Extension_C}
                          (4160)
\p{Block: CJK_Ext_D}    \p{Block=
                          CJK_Unified_Ideographs_Extension_D} (224)
\p{Block: CJK_Radicals_Sup} \p{Block=CJK_Radicals_Supplement} (128)
\p{Block: CJK_Radicals_Supplement} (Short: \p{Blk=CJKRadicalsSup},
                          \p{InCJKRadicalsSup}) (128)
\p{Block: CJK_Strokes}  (Single: \p{InCJKStrokes}) (48)
\p{Block: CJK_Symbols}  \p{Block=CJK_Symbols_And_Punctuation} (64)
\p{Block: CJK_Symbols_And_Punctuation} (Short: \p{Blk=CJKSymbols},
                          \p{InCJKSymbols}) (64)
\p{Block: CJK_Unified_Ideographs} (Short: \p{Blk=CJK}, \p{InCJK})
                          (20_992)
\p{Block: CJK_Unified_Ideographs_Extension_A} (Short: \p{Blk=
                          CJKExtA}, \p{InCJKExtA}) (6592)
\p{Block: CJK_Unified_Ideographs_Extension_B} (Short: \p{Blk=
                          CJKExtB}, \p{InCJKExtB}) (42_720)
\p{Block: CJK_Unified_Ideographs_Extension_C} (Short: \p{Blk=
                          CJKExtC}, \p{InCJKExtC}) (4160)
\p{Block: CJK_Unified_Ideographs_Extension_D} (Short: \p{Blk=
                          CJKExtD}, \p{InCJKExtD}) (224)
\p{Block: Combining_Diacritical_Marks} (Short: \p{Blk=
                          Diacriticals}, \p{InDiacriticals}) (112)
```

```
\p{Block: Combining_Diacritical_Marks_For_Symbols} (Short: \p{Blk=
                        DiacriticalsForSymbols},
                        \p{InDiacriticalsForSymbols}) (48)
\p{Block: Combining_Diacritical_Marks_Supplement} (Short: \p{Blk=
                        DiacriticalsSup}, \p{InDiacriticalsSup})
                        (64)
\p{Block: Combining_Half_Marks} (Short: \p{Blk=HalfMarks},
                        \p{InHalfMarks}) (16)
\p{Block: Combining_Marks_For_Symbols} \p{Block=
                        Combining_Diacritical_Marks_For_Symbols}
                        (48)
\p{Block: Common_Indic_Number_Forms} (Short: \p{Blk=
                        IndicNumberForms},
                        \p{InIndicNumberForms}) (16)
\p{Block: Compat_Jamo}  \p{Block=Hangul_Compatibility_Jamo} (96)
\p{Block: Control_Pictures} (Single: \p{InControlPictures}) (64)
\p{Block: Coptic}       (Single: \p{InCoptic}; NOT \p{Coptic} NOR
                        \p{Is_Coptic}) (128)
\p{Block: Counting_Rod} \p{Block=Counting_Rod_Numerals} (32)
\p{Block: Counting_Rod_Numerals} (Short: \p{Blk=CountingRod},
                        \p{InCountingRod}) (32)
\p{Block: Cuneiform}    (Single: \p{InCuneiform}; NOT
                        \p{Cuneiform} NOR \p{Is_Cuneiform})
                        (1024)
\p{Block: Cuneiform_Numbers} \p{Block=
                        Cuneiform_Numbers_And_Punctuation} (128)
\p{Block: Cuneiform_Numbers_And_Punctuation} (Short: \p{Blk=
                        CuneiformNumbers},
                        \p{InCuneiformNumbers}) (128)
\p{Block: Currency_Symbols} (Single: \p{InCurrencySymbols}) (48)
\p{Block: Cypriot_Syllabary} (Single: \p{InCypriotSyllabary}) (64)
\p{Block: Cyrillic}     (Single: \p{InCyrillic}; NOT \p{Cyrillic}
                        NOR \p{Is_Cyrillic}) (256)
\p{Block: Cyrillic_Ext_A} \p{Block=Cyrillic_Extended_A} (32)
\p{Block: Cyrillic_Ext_B} \p{Block=Cyrillic_Extended_B} (96)
\p{Block: Cyrillic_Extended_A} (Short: \p{Blk=CyrillicExtA},
                        \p{InCyrillicExtA}) (32)
\p{Block: Cyrillic_Extended_B} (Short: \p{Blk=CyrillicExtB},
                        \p{InCyrillicExtB}) (96)
\p{Block: Cyrillic_Sup} \p{Block=Cyrillic_Supplement} (48)
\p{Block: Cyrillic_Supplement} (Short: \p{Blk=CyrillicSup},
                        \p{InCyrillicSup}) (48)
\p{Block: Cyrillic_Supplementary} \p{Block=Cyrillic_Supplement}
                        (48)
\p{Block: Deseret}      (Single: \p{InDeseret}) (80)
\p{Block: Devanagari}   (Single: \p{InDevanagari}; NOT
                        \p{Devanagari} NOR \p{Is_Devanagari})
                        (128)
\p{Block: Devanagari_Ext} \p{Block=Devanagari_Extended} (32)
\p{Block: Devanagari_Extended} (Short: \p{Blk=DevanagariExt},
                        \p{InDevanagariExt}) (32)
\p{Block: Diacriticals} \p{Block=Combining_Diacritical_Marks} (112)
\p{Block: Diacriticals_For_Symbols} \p{Block=
                        Combining_Diacritical_Marks_For_Symbols}
                        (48)
\p{Block: Diacriticals_Sup} \p{Block=
```

```
                            Combining_Diacritical_Marks_Supplement}
                            (64)
\p{Block: Dingbats}     (Single: \p{InDingbats}) (192)
\p{Block: Domino}       \p{Block=Domino_Tiles} (112)
\p{Block: Domino_Tiles} (Short: \p{Blk=Domino}, \p{InDomino}) (112)
\p{Block: Egyptian_Hieroglyphs} (Single:
                            \p{InEgyptianHieroglyphs}; NOT
                            \p{Egyptian_Hieroglyphs} NOR
                            \p{Is_Egyptian_Hieroglyphs}) (1072)
\p{Block: Emoticons}    (Single: \p{InEmoticons}) (80)
\p{Block: Enclosed_Alphanum} \p{Block=Enclosed_Alphanumerics} (160)
\p{Block: Enclosed_Alphanum_Sup} \p{Block=
                            Enclosed_Alphanumeric_Supplement} (256)
\p{Block: Enclosed_Alphanumeric_Supplement} (Short: \p{Blk=
                            EnclosedAlphanumSup},
                            \p{InEnclosedAlphanumSup}) (256)
\p{Block: Enclosed_Alphanumerics} (Short: \p{Blk=
                            EnclosedAlphanum},
                            \p{InEnclosedAlphanum}) (160)
\p{Block: Enclosed_CJK} \p{Block=Enclosed_CJK_Letters_And_Months}
                            (256)
\p{Block: Enclosed_CJK_Letters_And_Months} (Short: \p{Blk=
                            EnclosedCJK}, \p{InEnclosedCJK}) (256)
\p{Block: Enclosed_Ideographic_Sup} \p{Block=
                            Enclosed_Ideographic_Supplement} (256)
\p{Block: Enclosed_Ideographic_Supplement} (Short: \p{Blk=
                            EnclosedIdeographicSup},
                            \p{InEnclosedIdeographicSup}) (256)
\p{Block: Ethiopic}     (Single: \p{InEthiopic}; NOT \p{Ethiopic}
                            NOR \p{Is_Ethiopic}) (384)
\p{Block: Ethiopic_Ext} \p{Block=Ethiopic_Extended} (96)
\p{Block: Ethiopic_Ext_A} \p{Block=Ethiopic_Extended_A} (48)
\p{Block: Ethiopic_Extended} (Short: \p{Blk=EthiopicExt},
                            \p{InEthiopicExt}) (96)
\p{Block: Ethiopic_Extended_A} (Short: \p{Blk=EthiopicExtA},
                            \p{InEthiopicExtA}) (48)
\p{Block: Ethiopic_Sup} \p{Block=Ethiopic_Supplement} (32)
\p{Block: Ethiopic_Supplement} (Short: \p{Blk=EthiopicSup},
                            \p{InEthiopicSup}) (32)
\p{Block: General_Punctuation} (Short: \p{Blk=Punctuation},
                            \p{InPunctuation}; NOT \p{Punct} NOR
                            \p{Is_Punctuation}) (112)
\p{Block: Geometric_Shapes} (Single: \p{InGeometricShapes}) (96)
\p{Block: Georgian}     (Single: \p{InGeorgian}; NOT \p{Georgian}
                            NOR \p{Is_Georgian}) (96)
\p{Block: Georgian_Sup} \p{Block=Georgian_Supplement} (48)
\p{Block: Georgian_Supplement} (Short: \p{Blk=GeorgianSup},
                            \p{InGeorgianSup}) (48)
\p{Block: Glagolitic}   (Single: \p{InGlagolitic}; NOT
                            \p{Glagolitic} NOR \p{Is_Glagolitic})
                            (96)
\p{Block: Gothic}       (Single: \p{InGothic}; NOT \p{Gothic} NOR
                            \p{Is_Gothic}) (32)
\p{Block: Greek}        \p{Block=Greek_And_Coptic} (NOT \p{Greek}
                            NOR \p{Is_Greek}) (144)
\p{Block: Greek_And_Coptic} (Short: \p{Blk=Greek}, \p{InGreek};
```

```
                              NOT \p{Greek} NOR \p{Is_Greek}) (144)
\p{Block: Greek_Ext}     \p{Block=Greek_Extended} (256)
\p{Block: Greek_Extended} (Short: \p{Blk=GreekExt},
                              \p{InGreekExt}) (256)
\p{Block: Gujarati}      (Single: \p{InGujarati}; NOT \p{Gujarati}
                              NOR \p{Is_Gujarati}) (128)
\p{Block: Gurmukhi}      (Single: \p{InGurmukhi}; NOT \p{Gurmukhi}
                              NOR \p{Is_Gurmukhi}) (128)
\p{Block: Half_And_Full_Forms} \p{Block=
                              Halfwidth_And_Fullwidth_Forms} (240)
\p{Block: Half_Marks}    \p{Block=Combining_Half_Marks} (16)
\p{Block: Halfwidth_And_Fullwidth_Forms} (Short: \p{Blk=
                              HalfAndFullForms},
                              \p{InHalfAndFullForms}) (240)
\p{Block: Hangul}        \p{Block=Hangul_Syllables} (NOT \p{Hangul}
                              NOR \p{Is_Hangul}) (11_184)
\p{Block: Hangul_Compatibility_Jamo} (Short: \p{Blk=CompatJamo},
                              \p{InCompatJamo}) (96)
\p{Block: Hangul_Jamo}  (Short: \p{Blk=Jamo}, \p{InJamo}) (256)
\p{Block: Hangul_Jamo_Extended_A} (Short: \p{Blk=JamoExtA},
                              \p{InJamoExtA}) (32)
\p{Block: Hangul_Jamo_Extended_B} (Short: \p{Blk=JamoExtB},
                              \p{InJamoExtB}) (80)
\p{Block: Hangul_Syllables} (Short: \p{Blk=Hangul}, \p{InHangul};
                              NOT \p{Hangul} NOR \p{Is_Hangul})
                              (11_184)
\p{Block: Hanunoo}       (Single: \p{InHanunoo}; NOT \p{Hanunoo}
                              NOR \p{Is_Hanunoo}) (32)
\p{Block: Hebrew}        (Single: \p{InHebrew}; NOT \p{Hebrew} NOR
                              \p{Is_Hebrew}) (112)
\p{Block: High_Private_Use_Surrogates} (Short: \p{Blk=
                              HighPUSurrogates},
                              \p{InHighPUSurrogates}) (128)
\p{Block: High_PU_Surrogates} \p{Block=
                              High_Private_Use_Surrogates} (128)
\p{Block: High_Surrogates} (Single: \p{InHighSurrogates}) (896)
\p{Block: Hiragana}      (Single: \p{InHiragana}; NOT \p{Hiragana}
                              NOR \p{Is_Hiragana}) (96)
\p{Block: IDC}           \p{Block=
                              Ideographic_Description_Characters} (NOT
                              \p{ID_Continue} NOR \p{Is_IDC}) (16)
\p{Block: Ideographic_Description_Characters} (Short: \p{Blk=IDC},
                              \p{InIDC}; NOT \p{ID_Continue} NOR
                              \p{Is_IDC}) (16)
\p{Block: Imperial_Aramaic} (Single: \p{InImperialAramaic}; NOT
                              \p{Imperial_Aramaic} NOR
                              \p{Is_Imperial_Aramaic}) (32)
\p{Block: Indic_Number_Forms} \p{Block=Common_Indic_Number_Forms}
                              (16)
\p{Block: Inscriptional_Pahlavi} (Single:
                              \p{InInscriptionalPahlavi}; NOT
                              \p{Inscriptional_Pahlavi} NOR
                              \p{Is_Inscriptional_Pahlavi}) (32)
\p{Block: Inscriptional_Parthian} (Single:
                              \p{InInscriptionalParthian}; NOT
                              \p{Inscriptional_Parthian} NOR
```

```
                            \p{Is_Inscriptional_Parthian}) (32)
\p{Block: IPA_Ext}      \p{Block=IPA_Extensions} (96)
\p{Block: IPA_Extensions} (Short: \p{Blk=IPAExt}, \p{InIPAExt})
                            (96)
\p{Block: Jamo}         \p{Block=Hangul_Jamo} (256)
\p{Block: Jamo_Ext_A}   \p{Block=Hangul_Jamo_Extended_A} (32)
\p{Block: Jamo_Ext_B}   \p{Block=Hangul_Jamo_Extended_B} (80)
\p{Block: Javanese}     (Single: \p{InJavanese}; NOT \p{Javanese}
                            NOR \p{Is_Javanese}) (96)
\p{Block: Kaithi}       (Single: \p{InKaithi}; NOT \p{Kaithi} NOR
                            \p{Is_Kaithi}) (80)
\p{Block: Kana_Sup}     \p{Block=Kana_Supplement} (256)
\p{Block: Kana_Supplement} (Short: \p{Blk=KanaSup}, \p{InKanaSup})
                            (256)
\p{Block: Kanbun}       (Single: \p{InKanbun}) (16)
\p{Block: Kangxi}       \p{Block=Kangxi_Radicals} (224)
\p{Block: Kangxi_Radicals} (Short: \p{Blk=Kangxi}, \p{InKangxi})
                            (224)
\p{Block: Kannada}      (Single: \p{InKannada}; NOT \p{Kannada}
                            NOR \p{Is_Kannada}) (128)
\p{Block: Katakana}     (Single: \p{InKatakana}; NOT \p{Katakana}
                            NOR \p{Is_Katakana}) (96)
\p{Block: Katakana_Ext} \p{Block=Katakana_Phonetic_Extensions} (16)
\p{Block: Katakana_Phonetic_Extensions} (Short: \p{Blk=
                            KatakanaExt}, \p{InKatakanaExt}) (16)
\p{Block: Kayah_Li}     (Single: \p{InKayahLi}) (48)
\p{Block: Kharoshthi}   (Single: \p{InKharoshthi}; NOT
                            \p{Kharoshthi} NOR \p{Is_Kharoshthi})
                            (96)
\p{Block: Khmer}        (Single: \p{InKhmer}; NOT \p{Khmer} NOR
                            \p{Is_Khmer}) (128)
\p{Block: Khmer_Symbols} (Single: \p{InKhmerSymbols}) (32)
\p{Block: Lao}          (Single: \p{InLao}; NOT \p{Lao} NOR
                            \p{Is_Lao}) (128)
\p{Block: Latin_1}      \p{Block=Latin_1_Supplement} (128)
\p{Block: Latin_1_Sup}  \p{Block=Latin_1_Supplement} (128)
\p{Block: Latin_1_Supplement} (Short: \p{Blk=Latin1},
                            \p{InLatin1}) (128)
\p{Block: Latin_Ext_A}  \p{Block=Latin_Extended_A} (128)
\p{Block: Latin_Ext_Additional} \p{Block=
                            Latin_Extended_Additional} (256)
\p{Block: Latin_Ext_B}  \p{Block=Latin_Extended_B} (208)
\p{Block: Latin_Ext_C}  \p{Block=Latin_Extended_C} (32)
\p{Block: Latin_Ext_D}  \p{Block=Latin_Extended_D} (224)
\p{Block: Latin_Extended_A} (Short: \p{Blk=LatinExtA},
                            \p{InLatinExtA}) (128)
\p{Block: Latin_Extended_Additional} (Short: \p{Blk=
                            LatinExtAdditional},
                            \p{InLatinExtAdditional}) (256)
\p{Block: Latin_Extended_B} (Short: \p{Blk=LatinExtB},
                            \p{InLatinExtB}) (208)
\p{Block: Latin_Extended_C} (Short: \p{Blk=LatinExtC},
                            \p{InLatinExtC}) (32)
\p{Block: Latin_Extended_D} (Short: \p{Blk=LatinExtD},
                            \p{InLatinExtD}) (224)
\p{Block: Lepcha}       (Single: \p{InLepcha}; NOT \p{Lepcha} NOR
```

```
                            \p{Is_Lepcha}) (80)
\p{Block: Letterlike_Symbols} (Single: \p{InLetterlikeSymbols})
                            (80)
\p{Block: Limbu}        (Single: \p{InLimbu}; NOT \p{Limbu} NOR
                            \p{Is_Limbu}) (80)
\p{Block: Linear_B_Ideograms} (Single: \p{InLinearBIdeograms})
                            (128)
\p{Block: Linear_B_Syllabary} (Single: \p{InLinearBSyllabary})
                            (128)
\p{Block: Lisu}         (Single: \p{InLisu}) (48)
\p{Block: Low_Surrogates} (Single: \p{InLowSurrogates}) (1024)
\p{Block: Lycian}       (Single: \p{InLycian}; NOT \p{Lycian} NOR
                            \p{Is_Lycian}) (32)
\p{Block: Lydian}       (Single: \p{InLydian}; NOT \p{Lydian} NOR
                            \p{Is_Lydian}) (32)
\p{Block: Mahjong}      \p{Block=Mahjong_Tiles} (48)
\p{Block: Mahjong_Tiles} (Short: \p{Blk=Mahjong}, \p{InMahjong})
                            (48)
\p{Block: Malayalam}    (Single: \p{InMalayalam}; NOT
                            \p{Malayalam} NOR \p{Is_Malayalam}) (128)
\p{Block: Mandaic}      (Single: \p{InMandaic}; NOT \p{Mandaic}
                            NOR \p{Is_Mandaic}) (32)
\p{Block: Math_Alphanum} \p{Block=
                            Mathematical_Alphanumeric_Symbols} (1024)
\p{Block: Math_Operators} \p{Block=Mathematical_Operators} (256)
\p{Block: Mathematical_Alphanumeric_Symbols} (Short: \p{Blk=
                            MathAlphanum}, \p{InMathAlphanum}) (1024)
\p{Block: Mathematical_Operators} (Short: \p{Blk=MathOperators},
                            \p{InMathOperators}) (256)
\p{Block: Meetei_Mayek} (Single: \p{InMeeteiMayek}; NOT
                            \p{Meetei_Mayek} NOR
                            \p{Is_Meetei_Mayek}) (64)
\p{Block: Meetei_Mayek_Ext} \p{Block=Meetei_Mayek_Extensions} (32)
\p{Block: Meetei_Mayek_Extensions} (Short: \p{Blk=MeeteiMayekExt},
                            \p{InMeeteiMayekExt}) (32)
\p{Block: Meroitic_Cursive} (Single: \p{InMeroiticCursive}; NOT
                            \p{Meroitic_Cursive} NOR
                            \p{Is_Meroitic_Cursive}) (96)
\p{Block: Meroitic_Hieroglyphs} (Single:
                            \p{InMeroiticHieroglyphs}) (32)
\p{Block: Miao}         (Single: \p{InMiao}; NOT \p{Miao} NOR
                            \p{Is_Miao}) (160)
\p{Block: Misc_Arrows}  \p{Block=Miscellaneous_Symbols_And_Arrows}
                            (256)
\p{Block: Misc_Math_Symbols_A} \p{Block=
                            Miscellaneous_Mathematical_Symbols_A}
                            (48)
\p{Block: Misc_Math_Symbols_B} \p{Block=
                            Miscellaneous_Mathematical_Symbols_B}
                            (128)
\p{Block: Misc_Pictographs} \p{Block=
                            Miscellaneous_Symbols_And_Pictographs}
                            (768)
\p{Block: Misc_Symbols} \p{Block=Miscellaneous_Symbols} (256)
\p{Block: Misc_Technical} \p{Block=Miscellaneous_Technical} (256)
\p{Block: Miscellaneous_Mathematical_Symbols_A} (Short: \p{Blk=
```

```
                            MiscMathSymbolsA},
                            \p{InMiscMathSymbolsA}) (48)
\p{Block: Miscellaneous_Mathematical_Symbols_B} (Short: \p{Blk=
                            MiscMathSymbolsB},
                            \p{InMiscMathSymbolsB}) (128)
\p{Block: Miscellaneous_Symbols} (Short: \p{Blk=MiscSymbols},
                            \p{InMiscSymbols}) (256)
\p{Block: Miscellaneous_Symbols_And_Arrows} (Short: \p{Blk=
                            MiscArrows}, \p{InMiscArrows}) (256)
\p{Block: Miscellaneous_Symbols_And_Pictographs} (Short: \p{Blk=
                            MiscPictographs}, \p{InMiscPictographs})
                            (768)
\p{Block: Miscellaneous_Technical} (Short: \p{Blk=MiscTechnical},
                            \p{InMiscTechnical}) (256)
\p{Block: Modifier_Letters} \p{Block=Spacing_Modifier_Letters} (80)
\p{Block: Modifier_Tone_Letters} (Single:
                            \p{InModifierToneLetters}) (32)
\p{Block: Mongolian}    (Single: \p{InMongolian}; NOT
                            \p{Mongolian} NOR \p{Is_Mongolian}) (176)
\p{Block: Music}        \p{Block=Musical_Symbols} (256)
\p{Block: Musical_Symbols} (Short: \p{Blk=Music}, \p{InMusic})
                            (256)
\p{Block: Myanmar}      (Single: \p{InMyanmar}; NOT \p{Myanmar}
                            NOR \p{Is_Myanmar}) (160)
\p{Block: Myanmar_Ext_A} \p{Block=Myanmar_Extended_A} (32)
\p{Block: Myanmar_Extended_A} (Short: \p{Blk=MyanmarExtA},
                            \p{InMyanmarExtA}) (32)
\p{Block: NB}           \p{Block=No_Block} (860_672 plus all
                            above-Unicode code points)
\p{Block: New_Tai_Lue}  (Single: \p{InNewTaiLue}; NOT
                            \p{New_Tai_Lue} NOR \p{Is_New_Tai_Lue})
                            (96)
\p{Block: NKo}          (Single: \p{InNKo}; NOT \p{Nko} NOR
                            \p{Is_NKo}) (64)
\p{Block: No_Block}     (Short: \p{Blk=NB}, \p{InNB}) (860_672
                            plus all above-Unicode code points)
\p{Block: Number_Forms} (Single: \p{InNumberForms}) (64)
\p{Block: OCR}          \p{Block=Optical_Character_Recognition}
                            (32)
\p{Block: Ogham}        (Single: \p{InOgham}; NOT \p{Ogham} NOR
                            \p{Is_Ogham}) (32)
\p{Block: Ol_Chiki}     (Single: \p{InOlChiki}) (48)
\p{Block: Old_Italic}   (Single: \p{InOldItalic}; NOT
                            \p{Old_Italic} NOR \p{Is_Old_Italic})
                            (48)
\p{Block: Old_Persian}  (Single: \p{InOldPersian}; NOT
                            \p{Old_Persian} NOR \p{Is_Old_Persian})
                            (64)
\p{Block: Old_South_Arabian} (Single: \p{InOldSouthArabian}) (32)
\p{Block: Old_Turkic}   (Single: \p{InOldTurkic}; NOT
                            \p{Old_Turkic} NOR \p{Is_Old_Turkic})
                            (80)
\p{Block: Optical_Character_Recognition} (Short: \p{Blk=OCR},
                            \p{InOCR}) (32)
\p{Block: Oriya}        (Single: \p{InOriya}; NOT \p{Oriya} NOR
                            \p{Is_Oriya}) (128)
```

```
\p{Block: Osmanya}      (Single: \p{InOsmanya}; NOT \p{Osmanya}
                          NOR \p{Is_Osmanya}) (48)
\p{Block: Phags_Pa}     (Single: \p{InPhagsPa}; NOT \p{Phags_Pa}
                          NOR \p{Is_Phags_Pa}) (64)
\p{Block: Phaistos}     \p{Block=Phaistos_Disc} (48)
\p{Block: Phaistos_Disc} (Short: \p{Blk=Phaistos}, \p{InPhaistos})
                          (48)
\p{Block: Phoenician}   (Single: \p{InPhoenician}; NOT
                          \p{Phoenician} NOR \p{Is_Phoenician})
                          (32)
\p{Block: Phonetic_Ext} \p{Block=Phonetic_Extensions} (128)
\p{Block: Phonetic_Ext_Sup} \p{Block=
                          Phonetic_Extensions_Supplement} (64)
\p{Block: Phonetic_Extensions} (Short: \p{Blk=PhoneticExt},
                          \p{InPhoneticExt}) (128)
\p{Block: Phonetic_Extensions_Supplement} (Short: \p{Blk=
                          PhoneticExtSup}, \p{InPhoneticExtSup})
                          (64)
\p{Block: Playing_Cards} (Single: \p{InPlayingCards}) (96)
\p{Block: Private_Use}  \p{Block=Private_Use_Area} (NOT
                          \p{Private_Use} NOR \p{Is_Private_Use})
                          (6400)
\p{Block: Private_Use_Area} (Short: \p{Blk=PUA}, \p{InPUA}; NOT
                          \p{Private_Use} NOR \p{Is_Private_Use})
                          (6400)
\p{Block: PUA}          \p{Block=Private_Use_Area} (NOT
                          \p{Private_Use} NOR \p{Is_Private_Use})
                          (6400)
\p{Block: Punctuation}  \p{Block=General_Punctuation} (NOT
                          \p{Punct} NOR \p{Is_Punctuation}) (112)
\p{Block: Rejang}       (Single: \p{InRejang}; NOT \p{Rejang} NOR
                          \p{Is_Rejang}) (48)
\p{Block: Rumi}         \p{Block=Rumi_Numeral_Symbols} (32)
\p{Block: Rumi_Numeral_Symbols} (Short: \p{Blk=Rumi}, \p{InRumi})
                          (32)
\p{Block: Runic}        (Single: \p{InRunic}; NOT \p{Runic} NOR
                          \p{Is_Runic}) (96)
\p{Block: Samaritan}    (Single: \p{InSamaritan}; NOT
                          \p{Samaritan} NOR \p{Is_Samaritan}) (64)
\p{Block: Saurashtra}   (Single: \p{InSaurashtra}; NOT
                          \p{Saurashtra} NOR \p{Is_Saurashtra})
                          (96)
\p{Block: Sharada}      (Single: \p{InSharada}; NOT \p{Sharada}
                          NOR \p{Is_Sharada}) (96)
\p{Block: Shavian}      (Single: \p{InShavian}) (48)
\p{Block: Sinhala}      (Single: \p{InSinhala}; NOT \p{Sinhala}
                          NOR \p{Is_Sinhala}) (128)
\p{Block: Small_Form_Variants} (Short: \p{Blk=SmallForms},
                          \p{InSmallForms}) (32)
\p{Block: Small_Forms}  \p{Block=Small_Form_Variants} (32)
\p{Block: Sora_Sompeng} (Single: \p{InSoraSompeng}; NOT
                          \p{Sora_Sompeng} NOR
                          \p{Is_Sora_Sompeng}) (48)
\p{Block: Spacing_Modifier_Letters} (Short: \p{Blk=
                          ModifierLetters}, \p{InModifierLetters})
                          (80)
```

```
\p{Block: Specials}     (Single: \p{InSpecials}) (16)
\p{Block: Sundanese}    (Single: \p{InSundanese}; NOT
                          \p{Sundanese} NOR \p{Is_Sundanese}) (64)
\p{Block: Sundanese_Sup} \p{Block=Sundanese_Supplement} (16)
\p{Block: Sundanese_Supplement} (Short: \p{Blk=SundaneseSup},
                          \p{InSundaneseSup}) (16)
\p{Block: Sup_Arrows_A} \p{Block=Supplemental_Arrows_A} (16)
\p{Block: Sup_Arrows_B} \p{Block=Supplemental_Arrows_B} (128)
\p{Block: Sup_Math_Operators} \p{Block=
                          Supplemental_Mathematical_Operators}
                          (256)
\p{Block: Sup_PUA_A}    \p{Block=Supplementary_Private_Use_Area_A}
                          (65_536)
\p{Block: Sup_PUA_B}    \p{Block=Supplementary_Private_Use_Area_B}
                          (65_536)
\p{Block: Sup_Punctuation} \p{Block=Supplemental_Punctuation} (128)
\p{Block: Super_And_Sub} \p{Block=Superscripts_And_Subscripts} (48)
\p{Block: Superscripts_And_Subscripts} (Short: \p{Blk=
                          SuperAndSub}, \p{InSuperAndSub}) (48)
\p{Block: Supplemental_Arrows_A} (Short: \p{Blk=SupArrowsA},
                          \p{InSupArrowsA}) (16)
\p{Block: Supplemental_Arrows_B} (Short: \p{Blk=SupArrowsB},
                          \p{InSupArrowsB}) (128)
\p{Block: Supplemental_Mathematical_Operators} (Short: \p{Blk=
                          SupMathOperators},
                          \p{InSupMathOperators}) (256)
\p{Block: Supplemental_Punctuation} (Short: \p{Blk=
                          SupPunctuation}, \p{InSupPunctuation})
                          (128)
\p{Block: Supplementary_Private_Use_Area_A} (Short: \p{Blk=
                          SupPUAA}, \p{InSupPUAA}) (65_536)
\p{Block: Supplementary_Private_Use_Area_B} (Short: \p{Blk=
                          SupPUAB}, \p{InSupPUAB}) (65_536)
\p{Block: Syloti_Nagri} (Single: \p{InSylotiNagri}; NOT
                          \p{Syloti_Nagri} NOR
                          \p{Is_Syloti_Nagri}) (48)
\p{Block: Syriac}       (Single: \p{InSyriac}; NOT \p{Syriac} NOR
                          \p{Is_Syriac}) (80)
\p{Block: Tagalog}      (Single: \p{InTagalog}; NOT \p{Tagalog}
                          NOR \p{Is_Tagalog}) (32)
\p{Block: Tagbanwa}     (Single: \p{InTagbanwa}; NOT \p{Tagbanwa}
                          NOR \p{Is_Tagbanwa}) (32)
\p{Block: Tags}         (Single: \p{InTags}) (128)
\p{Block: Tai_Le}       (Single: \p{InTaiLe}; NOT \p{Tai_Le} NOR
                          \p{Is_Tai_Le}) (48)
\p{Block: Tai_Tham}     (Single: \p{InTaiTham}; NOT \p{Tai_Tham}
                          NOR \p{Is_Tai_Tham}) (144)
\p{Block: Tai_Viet}     (Single: \p{InTaiViet}; NOT \p{Tai_Viet}
                          NOR \p{Is_Tai_Viet}) (96)
\p{Block: Tai_Xuan_Jing} \p{Block=Tai_Xuan_Jing_Symbols} (96)
\p{Block: Tai_Xuan_Jing_Symbols} (Short: \p{Blk=TaiXuanJing},
                          \p{InTaiXuanJing}) (96)
\p{Block: Takri}        (Single: \p{InTakri}; NOT \p{Takri} NOR
                          \p{Is_Takri}) (80)
\p{Block: Tamil}        (Single: \p{InTamil}; NOT \p{Tamil} NOR
                          \p{Is_Tamil}) (128)
```

```
    \p{Block: Telugu}       (Single: \p{InTelugu}; NOT \p{Telugu} NOR
                              \p{Is_Telugu}) (128)
    \p{Block: Thaana}       (Single: \p{InThaana}; NOT \p{Thaana} NOR
                              \p{Is_Thaana}) (64)
    \p{Block: Thai}         (Single: \p{InThai}; NOT \p{Thai} NOR
                              \p{Is_Thai}) (128)
    \p{Block: Tibetan}      (Single: \p{InTibetan}; NOT \p{Tibetan}
                              NOR \p{Is_Tibetan}) (256)
    \p{Block: Tifinagh}     (Single: \p{InTifinagh}; NOT \p{Tifinagh}
                              NOR \p{Is_Tifinagh}) (80)
    \p{Block: Transport_And_Map} \p{Block=Transport_And_Map_Symbols}
                              (128)
    \p{Block: Transport_And_Map_Symbols} (Short: \p{Blk=
                              TransportAndMap}, \p{InTransportAndMap})
                              (128)
    \p{Block: UCAS}         \p{Block=
                              Unified_Canadian_Aboriginal_Syllabics}
                              (640)
    \p{Block: UCAS_Ext}     \p{Block=
                              Unified_Canadian_Aboriginal_Syllabics_-
                              Extended} (80)
    \p{Block: Ugaritic}     (Single: \p{InUgaritic}; NOT \p{Ugaritic}
                              NOR \p{Is_Ugaritic}) (32)
    \p{Block: Unified_Canadian_Aboriginal_Syllabics} (Short: \p{Blk=
                              UCAS}, \p{InUCAS}) (640)
    \p{Block: Unified_Canadian_Aboriginal_Syllabics_Extended} (Short:
                              \p{Blk=UCASExt}, \p{InUCASExt}) (80)
    \p{Block: Vai}          (Single: \p{InVai}; NOT \p{Vai} NOR
                              \p{Is_Vai}) (320)
    \p{Block: Variation_Selectors} (Short: \p{Blk=VS}, \p{InVS}; NOT
                              \p{Variation_Selector} NOR \p{Is_VS})
                              (16)
    \p{Block: Variation_Selectors_Supplement} (Short: \p{Blk=VSSup},
                              \p{InVSSup}) (240)
    \p{Block: Vedic_Ext}    \p{Block=Vedic_Extensions} (48)
    \p{Block: Vedic_Extensions} (Short: \p{Blk=VedicExt},
                              \p{InVedicExt}) (48)
    \p{Block: Vertical_Forms} (Single: \p{InVerticalForms}) (16)
    \p{Block: VS}           \p{Block=Variation_Selectors} (NOT
                              \p{Variation_Selector} NOR \p{Is_VS})
                              (16)
    \p{Block: VS_Sup}       \p{Block=Variation_Selectors_Supplement}
                              (240)
    \p{Block: Yi_Radicals}  (Single: \p{InYiRadicals}) (64)
    \p{Block: Yi_Syllables} (Single: \p{InYiSyllables}) (1168)
    \p{Block: Yijing}       \p{Block=Yijing_Hexagram_Symbols} (64)
    \p{Block: Yijing_Hexagram_Symbols} (Short: \p{Blk=Yijing},
                              \p{InYijing}) (64)
  X \p{Block_Elements}      \p{Block=Block_Elements} (32)
    \p{Bopo}                \p{Bopomofo} (= \p{Script=Bopomofo}) (NOT
                              \p{Block=Bopomofo}) (70)
    \p{Bopomofo}            \p{Script=Bopomofo} (Short: \p{Bopo}; NOT
                              \p{Block=Bopomofo}) (70)
  X \p{Bopomofo_Ext}        \p{Bopomofo_Extended} (= \p{Block=
                              Bopomofo_Extended}) (32)
  X \p{Bopomofo_Extended}   \p{Block=Bopomofo_Extended} (Short:
```

```
                              \p{InBopomofoExt}) (32)
X \p{Box_Drawing}           \p{Block=Box_Drawing} (128)
  \p{Bpt: *}                \p{Bidi_Paired_Bracket_Type: *}
  \p{Brah}                  \p{Brahmi} (= \p{Script=Brahmi}) (NOT
                              \p{Block=Brahmi}) (108)
  \p{Brahmi}                \p{Script=Brahmi} (Short: \p{Brah}; NOT
                              \p{Block=Brahmi}) (108)
  \p{Brai}                  \p{Braille} (= \p{Script=Braille}) (256)
  \p{Braille}               \p{Script=Braille} (Short: \p{Brai}) (256)
X \p{Braille_Patterns}      \p{Block=Braille_Patterns} (Short:
                              \p{InBraille}) (256)
  \p{Bugi}                  \p{Buginese} (= \p{Script=Buginese}) (NOT
                              \p{Block=Buginese}) (30)
  \p{Buginese}              \p{Script=Buginese} (Short: \p{Bugi}; NOT
                              \p{Block=Buginese}) (30)
  \p{Buhd}                  \p{Buhid} (= \p{Script=Buhid}) (NOT
                              \p{Block=Buhid}) (20)
  \p{Buhid}                 \p{Script=Buhid} (Short: \p{Buhd}; NOT
                              \p{Block=Buhid}) (20)
X \p{Byzantine_Music}       \p{Byzantine_Musical_Symbols} (= \p{Block=
                              Byzantine_Musical_Symbols}) (256)
X \p{Byzantine_Musical_Symbols} \p{Block=Byzantine_Musical_Symbols}
                              (Short: \p{InByzantineMusic}) (256)
  \p{C} \pC                 \p{Other} (= \p{General_Category=Other})
                              (1_004_135 plus all above-Unicode code
                              points)
  \p{Cakm}                  \p{Chakma} (= \p{Script=Chakma}) (NOT
                              \p{Block=Chakma}) (67)
  \p{Canadian_Aboriginal} \p{Script=Canadian_Aboriginal} (Short:
                              \p{Cans}) (710)
X \p{Canadian_Syllabics}  \p{Unified_Canadian_Aboriginal_Syllabics}
                              (= \p{Block=
                              Unified_Canadian_Aboriginal_Syllabics})
                              (640)
T \p{Canonical_Combining_Class: 0} \p{Canonical_Combining_Class=
                              Not_Reordered} (1_113_459 plus all
                              above-Unicode code points)
T \p{Canonical_Combining_Class: 1} \p{Canonical_Combining_Class=
                              Overlay} (26)
T \p{Canonical_Combining_Class: 7} \p{Canonical_Combining_Class=
                              Nukta} (13)
T \p{Canonical_Combining_Class: 8} \p{Canonical_Combining_Class=
                              Kana_Voicing} (2)
T \p{Canonical_Combining_Class: 9} \p{Canonical_Combining_Class=
                              Virama} (37)
T \p{Canonical_Combining_Class: 10} \p{Canonical_Combining_Class=
                              CCC10} (1)
T \p{Canonical_Combining_Class: 11} \p{Canonical_Combining_Class=
                              CCC11} (1)
T \p{Canonical_Combining_Class: 12} \p{Canonical_Combining_Class=
                              CCC12} (1)
T \p{Canonical_Combining_Class: 13} \p{Canonical_Combining_Class=
                              CCC13} (1)
T \p{Canonical_Combining_Class: 14} \p{Canonical_Combining_Class=
                              CCC14} (1)
T \p{Canonical_Combining_Class: 15} \p{Canonical_Combining_Class=
```

```
                                CCC15} (1)
T \p{Canonical_Combining_Class: 16} \p{Canonical_Combining_Class=
                                CCC16} (1)
T \p{Canonical_Combining_Class: 17} \p{Canonical_Combining_Class=
                                CCC17} (1)
T \p{Canonical_Combining_Class: 18} \p{Canonical_Combining_Class=
                                CCC18} (2)
T \p{Canonical_Combining_Class: 19} \p{Canonical_Combining_Class=
                                CCC19} (2)
T \p{Canonical_Combining_Class: 20} \p{Canonical_Combining_Class=
                                CCC20} (1)
T \p{Canonical_Combining_Class: 21} \p{Canonical_Combining_Class=
                                CCC21} (1)
T \p{Canonical_Combining_Class: 22} \p{Canonical_Combining_Class=
                                CCC22} (1)
T \p{Canonical_Combining_Class: 23} \p{Canonical_Combining_Class=
                                CCC23} (1)
T \p{Canonical_Combining_Class: 24} \p{Canonical_Combining_Class=
                                CCC24} (1)
T \p{Canonical_Combining_Class: 25} \p{Canonical_Combining_Class=
                                CCC25} (1)
T \p{Canonical_Combining_Class: 26} \p{Canonical_Combining_Class=
                                CCC26} (1)
T \p{Canonical_Combining_Class: 27} \p{Canonical_Combining_Class=
                                CCC27} (2)
T \p{Canonical_Combining_Class: 28} \p{Canonical_Combining_Class=
                                CCC28} (2)
T \p{Canonical_Combining_Class: 29} \p{Canonical_Combining_Class=
                                CCC29} (2)
T \p{Canonical_Combining_Class: 30} \p{Canonical_Combining_Class=
                                CCC30} (2)
T \p{Canonical_Combining_Class: 31} \p{Canonical_Combining_Class=
                                CCC31} (2)
T \p{Canonical_Combining_Class: 32} \p{Canonical_Combining_Class=
                                CCC32} (2)
T \p{Canonical_Combining_Class: 33} \p{Canonical_Combining_Class=
                                CCC33} (1)
T \p{Canonical_Combining_Class: 34} \p{Canonical_Combining_Class=
                                CCC34} (1)
T \p{Canonical_Combining_Class: 35} \p{Canonical_Combining_Class=
                                CCC35} (1)
T \p{Canonical_Combining_Class: 36} \p{Canonical_Combining_Class=
                                CCC36} (1)
T \p{Canonical_Combining_Class: 84} \p{Canonical_Combining_Class=
                                CCC84} (1)
T \p{Canonical_Combining_Class: 91} \p{Canonical_Combining_Class=
                                CCC91} (1)
T \p{Canonical_Combining_Class: 103} \p{Canonical_Combining_Class=
                                CCC103} (2)
T \p{Canonical_Combining_Class: 107} \p{Canonical_Combining_Class=
                                CCC107} (4)
T \p{Canonical_Combining_Class: 118} \p{Canonical_Combining_Class=
                                CCC118} (2)
T \p{Canonical_Combining_Class: 122} \p{Canonical_Combining_Class=
                                CCC122} (4)
T \p{Canonical_Combining_Class: 129} \p{Canonical_Combining_Class=
```

```
                                CCC129} (1)
T \p{Canonical_Combining_Class: 130} \p{Canonical_Combining_Class=
                                CCC130} (6)
T \p{Canonical_Combining_Class: 132} \p{Canonical_Combining_Class=
                                CCC132} (1)
T \p{Canonical_Combining_Class: 133} \p{Canonical_Combining_Class=
                                CCC133} (0)
T \p{Canonical_Combining_Class: 200} \p{Canonical_Combining_Class=
                                Attached_Below_Left} (0)
T \p{Canonical_Combining_Class: 202} \p{Canonical_Combining_Class=
                                Attached_Below} (5)
T \p{Canonical_Combining_Class: 214} \p{Canonical_Combining_Class=
                                Attached_Above} (1)
T \p{Canonical_Combining_Class: 216} \p{Canonical_Combining_Class=
                                Attached_Above_Right} (9)
T \p{Canonical_Combining_Class: 218} \p{Canonical_Combining_Class=
                                Below_Left} (1)
T \p{Canonical_Combining_Class: 220} \p{Canonical_Combining_Class=
                                Below} (129)
T \p{Canonical_Combining_Class: 222} \p{Canonical_Combining_Class=
                                Below_Right} (4)
T \p{Canonical_Combining_Class: 224} \p{Canonical_Combining_Class=
                                Left} (2)
T \p{Canonical_Combining_Class: 226} \p{Canonical_Combining_Class=
                                Right} (1)
T \p{Canonical_Combining_Class: 228} \p{Canonical_Combining_Class=
                                Above_Left} (3)
T \p{Canonical_Combining_Class: 230} \p{Canonical_Combining_Class=
                                Above} (349)
T \p{Canonical_Combining_Class: 232} \p{Canonical_Combining_Class=
                                Above_Right} (4)
T \p{Canonical_Combining_Class: 233} \p{Canonical_Combining_Class=
                                Double_Below} (4)
T \p{Canonical_Combining_Class: 234} \p{Canonical_Combining_Class=
                                Double_Above} (5)
T \p{Canonical_Combining_Class: 240} \p{Canonical_Combining_Class=
                                Iota_Subscript} (1)
  \p{Canonical_Combining_Class: A} \p{Canonical_Combining_Class=
                                Above} (349)
  \p{Canonical_Combining_Class: Above} (Short: \p{Ccc=A}) (349)
  \p{Canonical_Combining_Class: Above_Left} (Short: \p{Ccc=AL}) (3)
  \p{Canonical_Combining_Class: Above_Right} (Short: \p{Ccc=AR}) (4)
  \p{Canonical_Combining_Class: AL} \p{Canonical_Combining_Class=
                                Above_Left} (3)
  \p{Canonical_Combining_Class: AR} \p{Canonical_Combining_Class=
                                Above_Right} (4)
  \p{Canonical_Combining_Class: ATA} \p{Canonical_Combining_Class=
                                Attached_Above} (1)
  \p{Canonical_Combining_Class: ATAR} \p{Canonical_Combining_Class=
                                Attached_Above_Right} (9)
  \p{Canonical_Combining_Class: ATB} \p{Canonical_Combining_Class=
                                Attached_Below} (5)
  \p{Canonical_Combining_Class: ATBL} \p{Canonical_Combining_Class=
                                Attached_Below_Left} (0)
  \p{Canonical_Combining_Class: Attached_Above} (Short: \p{Ccc=ATA})
                                (1)
```

```
\p{Canonical_Combining_Class: Attached_Above_Right} (Short:
                          \p{Ccc=ATAR}) (9)
\p{Canonical_Combining_Class: Attached_Below} (Short: \p{Ccc=ATB})
                          (5)
\p{Canonical_Combining_Class: Attached_Below_Left} (Short: \p{Ccc=
                          ATBL}) (0)
\p{Canonical_Combining_Class: B} \p{Canonical_Combining_Class=
                          Below} (129)
\p{Canonical_Combining_Class: Below} (Short: \p{Ccc=B}) (129)
\p{Canonical_Combining_Class: Below_Left} (Short: \p{Ccc=BL}) (1)
\p{Canonical_Combining_Class: Below_Right} (Short: \p{Ccc=BR}) (4)
\p{Canonical_Combining_Class: BL} \p{Canonical_Combining_Class=
                          Below_Left} (1)
\p{Canonical_Combining_Class: BR} \p{Canonical_Combining_Class=
                          Below_Right} (4)
\p{Canonical_Combining_Class: CCC10} (Short: \p{Ccc=CCC10}) (1)
\p{Canonical_Combining_Class: CCC103} (Short: \p{Ccc=CCC103}) (2)
\p{Canonical_Combining_Class: CCC107} (Short: \p{Ccc=CCC107}) (4)
\p{Canonical_Combining_Class: CCC11} (Short: \p{Ccc=CCC11}) (1)
\p{Canonical_Combining_Class: CCC118} (Short: \p{Ccc=CCC118}) (2)
\p{Canonical_Combining_Class: CCC12} (Short: \p{Ccc=CCC12}) (1)
\p{Canonical_Combining_Class: CCC122} (Short: \p{Ccc=CCC122}) (4)
\p{Canonical_Combining_Class: CCC129} (Short: \p{Ccc=CCC129}) (1)
\p{Canonical_Combining_Class: CCC13} (Short: \p{Ccc=CCC13}) (1)
\p{Canonical_Combining_Class: CCC130} (Short: \p{Ccc=CCC130}) (6)
\p{Canonical_Combining_Class: CCC132} (Short: \p{Ccc=CCC132}) (1)
\p{Canonical_Combining_Class: CCC133} (Short: \p{Ccc=CCC133}) (0)
\p{Canonical_Combining_Class: CCC14} (Short: \p{Ccc=CCC14}) (1)
\p{Canonical_Combining_Class: CCC15} (Short: \p{Ccc=CCC15}) (1)
\p{Canonical_Combining_Class: CCC16} (Short: \p{Ccc=CCC16}) (1)
\p{Canonical_Combining_Class: CCC17} (Short: \p{Ccc=CCC17}) (1)
\p{Canonical_Combining_Class: CCC18} (Short: \p{Ccc=CCC18}) (2)
\p{Canonical_Combining_Class: CCC19} (Short: \p{Ccc=CCC19}) (2)
\p{Canonical_Combining_Class: CCC20} (Short: \p{Ccc=CCC20}) (1)
\p{Canonical_Combining_Class: CCC21} (Short: \p{Ccc=CCC21}) (1)
\p{Canonical_Combining_Class: CCC22} (Short: \p{Ccc=CCC22}) (1)
\p{Canonical_Combining_Class: CCC23} (Short: \p{Ccc=CCC23}) (1)
\p{Canonical_Combining_Class: CCC24} (Short: \p{Ccc=CCC24}) (1)
\p{Canonical_Combining_Class: CCC25} (Short: \p{Ccc=CCC25}) (1)
\p{Canonical_Combining_Class: CCC26} (Short: \p{Ccc=CCC26}) (1)
\p{Canonical_Combining_Class: CCC27} (Short: \p{Ccc=CCC27}) (2)
\p{Canonical_Combining_Class: CCC28} (Short: \p{Ccc=CCC28}) (2)
\p{Canonical_Combining_Class: CCC29} (Short: \p{Ccc=CCC29}) (2)
\p{Canonical_Combining_Class: CCC30} (Short: \p{Ccc=CCC30}) (2)
\p{Canonical_Combining_Class: CCC31} (Short: \p{Ccc=CCC31}) (2)
\p{Canonical_Combining_Class: CCC32} (Short: \p{Ccc=CCC32}) (2)
\p{Canonical_Combining_Class: CCC33} (Short: \p{Ccc=CCC33}) (1)
\p{Canonical_Combining_Class: CCC34} (Short: \p{Ccc=CCC34}) (1)
\p{Canonical_Combining_Class: CCC35} (Short: \p{Ccc=CCC35}) (1)
\p{Canonical_Combining_Class: CCC36} (Short: \p{Ccc=CCC36}) (1)
\p{Canonical_Combining_Class: CCC84} (Short: \p{Ccc=CCC84}) (1)
\p{Canonical_Combining_Class: CCC91} (Short: \p{Ccc=CCC91}) (1)
\p{Canonical_Combining_Class: DA} \p{Canonical_Combining_Class=
                          Double_Above} (5)
\p{Canonical_Combining_Class: DB} \p{Canonical_Combining_Class=
                          Double_Below} (4)
```

```
\p{Canonical_Combining_Class: Double_Above} (Short: \p{Ccc=DA}) (5)
\p{Canonical_Combining_Class: Double_Below} (Short: \p{Ccc=DB}) (4)
\p{Canonical_Combining_Class: Iota_Subscript} (Short: \p{Ccc=IS})
                          (1)
\p{Canonical_Combining_Class: IS} \p{Canonical_Combining_Class=
                          Iota_Subscript} (1)
\p{Canonical_Combining_Class: Kana_Voicing} (Short: \p{Ccc=KV}) (2)
\p{Canonical_Combining_Class: KV} \p{Canonical_Combining_Class=
                          Kana_Voicing} (2)
\p{Canonical_Combining_Class: L} \p{Canonical_Combining_Class=
                          Left} (2)
\p{Canonical_Combining_Class: Left} (Short: \p{Ccc=L}) (2)
\p{Canonical_Combining_Class: NK} \p{Canonical_Combining_Class=
                          Nukta} (13)
\p{Canonical_Combining_Class: Not_Reordered} (Short: \p{Ccc=NR})
                          (1_113_459 plus all above-Unicode code
                          points)
\p{Canonical_Combining_Class: NR} \p{Canonical_Combining_Class=
                          Not_Reordered} (1_113_459 plus all
                          above-Unicode code points)
\p{Canonical_Combining_Class: Nukta} (Short: \p{Ccc=NK}) (13)
\p{Canonical_Combining_Class: OV} \p{Canonical_Combining_Class=
                          Overlay} (26)
\p{Canonical_Combining_Class: Overlay} (Short: \p{Ccc=OV}) (26)
\p{Canonical_Combining_Class: R} \p{Canonical_Combining_Class=
                          Right} (1)
\p{Canonical_Combining_Class: Right} (Short: \p{Ccc=R}) (1)
\p{Canonical_Combining_Class: Virama} (Short: \p{Ccc=VR}) (37)
\p{Canonical_Combining_Class: VR} \p{Canonical_Combining_Class=
                          Virama} (37)
\p{Cans}                \p{Canadian_Aboriginal} (= \p{Script=
                          Canadian_Aboriginal}) (710)
\p{Cari}                \p{Carian} (= \p{Script=Carian}) (NOT
                          \p{Block=Carian}) (49)
\p{Carian}              \p{Script=Carian} (Short: \p{Cari}; NOT
                          \p{Block=Carian}) (49)
\p{Case_Ignorable}      \p{Case_Ignorable=Y} (Short: \p{CI}) (1806)
\p{Case_Ignorable: N*}  (Short: \p{CI=N}, \P{CI}) (1_112_306 plus
                          all above-Unicode code points)
\p{Case_Ignorable: Y*}  (Short: \p{CI=Y}, \p{CI}) (1806)
\p{Cased}               \p{Cased=Y} (3448)
\p{Cased: N*}           (Single: \P{Cased}) (1_110_664 plus all
                          above-Unicode code points)
\p{Cased: Y*}           (Single: \p{Cased}) (3448)
\p{Cased_Letter}        \p{General_Category=Cased_Letter} (Short:
                          \p{LC}) (3223)
\p{Category: *}         \p{General_Category: *}
\p{Cc}                  \p{Cntrl} (= \p{General_Category=Control})
                          (65)
\p{Ccc: *}              \p{Canonical_Combining_Class: *}
\p{CE}                  \p{Composition_Exclusion} (=
                          \p{Composition_Exclusion=Y}) (81)
\p{CE: *}               \p{Composition_Exclusion: *}
\p{Cf}                  \p{Format} (= \p{General_Category=Format})
                          (145)
\p{Chakma}              \p{Script=Chakma} (Short: \p{Cakm}; NOT
```

```
                              \p{Block=Chakma}) (67)
    \p{Cham}                \p{Script=Cham} (NOT \p{Block=Cham}) (83)
    \p{Changes_When_Casefolded} \p{Changes_When_Casefolded=Y} (Short:
                              \p{CWCF}) (1107)
    \p{Changes_When_Casefolded: N*} (Short: \p{CWCF=N}, \P{CWCF})
                              (1_113_005 plus all above-Unicode code
                              points)
    \p{Changes_When_Casefolded: Y*} (Short: \p{CWCF=Y}, \p{CWCF})
                              (1107)
    \p{Changes_When_Casemapped} \p{Changes_When_Casemapped=Y} (Short:
                              \p{CWCM}) (2138)
    \p{Changes_When_Casemapped: N*} (Short: \p{CWCM=N}, \P{CWCM})
                              (1_111_974 plus all above-Unicode code
                              points)
    \p{Changes_When_Casemapped: Y*} (Short: \p{CWCM=Y}, \p{CWCM})
                              (2138)
    \p{Changes_When_Lowercased} \p{Changes_When_Lowercased=Y} (Short:
                              \p{CWL}) (1043)
    \p{Changes_When_Lowercased: N*} (Short: \p{CWL=N}, \P{CWL})
                              (1_113_069 plus all above-Unicode code
                              points)
    \p{Changes_When_Lowercased: Y*} (Short: \p{CWL=Y}, \p{CWL}) (1043)
    \p{Changes_When_NFKC_Casefolded} \p{Changes_When_NFKC_Casefolded=
                              Y} (Short: \p{CWKCF}) (9946)
    \p{Changes_When_NFKC_Casefolded: N*} (Short: \p{CWKCF=N},
                              \P{CWKCF}) (1_104_166 plus all above-
                              Unicode code points)
    \p{Changes_When_NFKC_Casefolded: Y*} (Short: \p{CWKCF=Y},
                              \p{CWKCF}) (9946)
    \p{Changes_When_Titlecased} \p{Changes_When_Titlecased=Y} (Short:
                              \p{CWT}) (1099)
    \p{Changes_When_Titlecased: N*} (Short: \p{CWT=N}, \P{CWT})
                              (1_113_013 plus all above-Unicode code
                              points)
    \p{Changes_When_Titlecased: Y*} (Short: \p{CWT=Y}, \p{CWT}) (1099)
    \p{Changes_When_Uppercased} \p{Changes_When_Uppercased=Y} (Short:
                              \p{CWU}) (1126)
    \p{Changes_When_Uppercased: N*} (Short: \p{CWU=N}, \P{CWU})
                              (1_112_986 plus all above-Unicode code
                              points)
    \p{Changes_When_Uppercased: Y*} (Short: \p{CWU=Y}, \p{CWU}) (1126)
    \p{Cher}                \p{Cherokee} (= \p{Script=Cherokee}) (NOT
                              \p{Block=Cherokee}) (85)
    \p{Cherokee}            \p{Script=Cherokee} (Short: \p{Cher}; NOT
                              \p{Block=Cherokee}) (85)
    \p{CI}                  \p{Case_Ignorable} (= \p{Case_Ignorable=
                              Y}) (1806)
    \p{CI: *}               \p{Case_Ignorable: *}
  X \p{CJK}                 \p{CJK_Unified_Ideographs} (= \p{Block=
                              CJK_Unified_Ideographs}) (20_992)
  X \p{CJK_Compat}          \p{CJK_Compatibility} (= \p{Block=
                              CJK_Compatibility}) (256)
  X \p{CJK_Compat_Forms}    \p{CJK_Compatibility_Forms} (= \p{Block=
                              CJK_Compatibility_Forms}) (32)
  X \p{CJK_Compat_Ideographs} \p{CJK_Compatibility_Ideographs} (=
                              \p{Block=CJK_Compatibility_Ideographs})
```

```
                                (512)
X \p{CJK_Compat_Ideographs_Sup}
                                \p{CJK_Compatibility_Ideographs_-
                                Supplement} (= \p{Block=
                                CJK_Compatibility_Ideographs_-
                                Supplement}) (544)
X \p{CJK_Compatibility}    \p{Block=CJK_Compatibility} (Short:
                                \p{InCJKCompat}) (256)
X \p{CJK_Compatibility_Forms} \p{Block=CJK_Compatibility_Forms}
                                (Short: \p{InCJKCompatForms}) (32)
X \p{CJK_Compatibility_Ideographs} \p{Block=
                                CJK_Compatibility_Ideographs} (Short:
                                \p{InCJKCompatIdeographs}) (512)
X \p{CJK_Compatibility_Ideographs_Supplement} \p{Block=
                                CJK_Compatibility_Ideographs_Supplement}
                                (Short: \p{InCJKCompatIdeographsSup})
                                (544)
X \p{CJK_Ext_A}            \p{CJK_Unified_Ideographs_Extension_A} (=
                                \p{Block=
                                CJK_Unified_Ideographs_Extension_A})
                                (6592)
X \p{CJK_Ext_B}            \p{CJK_Unified_Ideographs_Extension_B} (=
                                \p{Block=
                                CJK_Unified_Ideographs_Extension_B})
                                (42_720)
X \p{CJK_Ext_C}            \p{CJK_Unified_Ideographs_Extension_C} (=
                                \p{Block=
                                CJK_Unified_Ideographs_Extension_C})
                                (4160)
X \p{CJK_Ext_D}            \p{CJK_Unified_Ideographs_Extension_D} (=
                                \p{Block=
                                CJK_Unified_Ideographs_Extension_D})
                                (224)
X \p{CJK_Radicals_Sup}     \p{CJK_Radicals_Supplement} (= \p{Block=
                                CJK_Radicals_Supplement}) (128)
X \p{CJK_Radicals_Supplement} \p{Block=CJK_Radicals_Supplement}
                                (Short: \p{InCJKRadicalsSup}) (128)
X \p{CJK_Strokes}          \p{Block=CJK_Strokes} (48)
X \p{CJK_Symbols}          \p{CJK_Symbols_And_Punctuation} (=
                                \p{Block=CJK_Symbols_And_Punctuation})
                                (64)
X \p{CJK_Symbols_And_Punctuation} \p{Block=
                                CJK_Symbols_And_Punctuation} (Short:
                                \p{InCJKSymbols}) (64)
X \p{CJK_Unified_Ideographs} \p{Block=CJK_Unified_Ideographs}
                                (Short: \p{InCJK}) (20_992)
X \p{CJK_Unified_Ideographs_Extension_A} \p{Block=
                                CJK_Unified_Ideographs_Extension_A}
                                (Short: \p{InCJKExtA}) (6592)
X \p{CJK_Unified_Ideographs_Extension_B} \p{Block=
                                CJK_Unified_Ideographs_Extension_B}
                                (Short: \p{InCJKExtB}) (42_720)
X \p{CJK_Unified_Ideographs_Extension_C} \p{Block=
                                CJK_Unified_Ideographs_Extension_C}
                                (Short: \p{InCJKExtC}) (4160)
X \p{CJK_Unified_Ideographs_Extension_D} \p{Block=
```

```
                              CJK_Unified_Ideographs_Extension_D}
                              (Short: \p{InCJKExtD}) (224)
  \p{Close_Punctuation}     \p{General_Category=Close_Punctuation}
                              (Short: \p{Pe}) (73)
  \p{Cn}                    \p{Unassigned} (= \p{General_Category=
                              Unassigned}) (864_409 plus all above-
                              Unicode code points)
  \p{Cntrl}                 \p{General_Category=Control} Control
                              characters (Short: \p{Cc}) (65)
  \p{Co}                    \p{Private_Use} (= \p{General_Category=
                              Private_Use}) (NOT \p{Private_Use_Area})
                              (137_468)
X \p{Combining_Diacritical_Marks} \p{Block=
                              Combining_Diacritical_Marks} (Short:
                              \p{InDiacriticals}) (112)
X \p{Combining_Diacritical_Marks_For_Symbols} \p{Block=
                              Combining_Diacritical_Marks_For_Symbols}
                              (Short: \p{InDiacriticalsForSymbols})
                              (48)
X \p{Combining_Diacritical_Marks_Supplement} \p{Block=
                              Combining_Diacritical_Marks_Supplement}
                              (Short: \p{InDiacriticalsSup}) (64)
X \p{Combining_Half_Marks} \p{Block=Combining_Half_Marks} (Short:
                              \p{InHalfMarks}) (16)
  \p{Combining_Mark}        \p{Mark} (= \p{General_Category=Mark})
                              (1645)
X \p{Combining_Marks_For_Symbols}
                              \p{Combining_Diacritical_Marks_For_-
                              Symbols} (= \p{Block=
                              Combining_Diacritical_Marks_For_-
                              Symbols}) (48)
  \p{Common}                \p{Script=Common} (Short: \p{Zyyy}) (6418)
X \p{Common_Indic_Number_Forms} \p{Block=Common_Indic_Number_Forms}
                              (Short: \p{InIndicNumberForms}) (16)
  \p{Comp_Ex}               \p{Full_Composition_Exclusion} (=
                              \p{Full_Composition_Exclusion=Y}) (1120)
  \p{Comp_Ex: *}            \p{Full_Composition_Exclusion: *}
X \p{Compat_Jamo}           \p{Hangul_Compatibility_Jamo} (= \p{Block=
                              Hangul_Compatibility_Jamo}) (96)
  \p{Composition_Exclusion} \p{Composition_Exclusion=Y} (Short:
                              \p{CE}) (81)
  \p{Composition_Exclusion: N*} (Short: \p{CE=N}, \P{CE}) (1_114_031
                              plus all above-Unicode code points)
  \p{Composition_Exclusion: Y*} (Short: \p{CE=Y}, \p{CE}) (81)
  \p{Connector_Punctuation} \p{General_Category=
                              Connector_Punctuation} (Short: \p{Pc})
                              (10)
  \p{Control}               \p{Cntrl} (= \p{General_Category=Control})
                              (65)
X \p{Control_Pictures}      \p{Block=Control_Pictures} (64)
  \p{Copt}                  \p{Coptic} (= \p{Script=Coptic}) (NOT
                              \p{Block=Coptic}) (137)
  \p{Coptic}                \p{Script=Coptic} (Short: \p{Copt}; NOT
                              \p{Block=Coptic}) (137)
X \p{Counting_Rod}          \p{Counting_Rod_Numerals} (= \p{Block=
                              Counting_Rod_Numerals}) (32)
```

```
X \p{Counting_Rod_Numerals} \p{Block=Counting_Rod_Numerals} (Short:
                            \p{InCountingRod}) (32)
  \p{Cprt}                \p{Cypriot} (= \p{Script=Cypriot}) (55)
  \p{Cs}                  \p{Surrogate} (= \p{General_Category=
                            Surrogate}) (2048)
  \p{Cuneiform}           \p{Script=Cuneiform} (Short: \p{Xsux}; NOT
                            \p{Block=Cuneiform}) (982)
X \p{Cuneiform_Numbers}   \p{Cuneiform_Numbers_And_Punctuation} (=
                            \p{Block=
                            Cuneiform_Numbers_And_Punctuation}) (128)
X \p{Cuneiform_Numbers_And_Punctuation} \p{Block=
                            Cuneiform_Numbers_And_Punctuation}
                            (Short: \p{InCuneiformNumbers}) (128)
  \p{Currency_Symbol}     \p{General_Category=Currency_Symbol}
                            (Short: \p{Sc}) (49)
X \p{Currency_Symbols}    \p{Block=Currency_Symbols} (48)
  \p{CWCF}                \p{Changes_When_Casefolded} (=
                            \p{Changes_When_Casefolded=Y}) (1107)
  \p{CWCF: *}             \p{Changes_When_Casefolded: *}
  \p{CWCM}                \p{Changes_When_Casemapped} (=
                            \p{Changes_When_Casemapped=Y}) (2138)
  \p{CWCM: *}             \p{Changes_When_Casemapped: *}
  \p{CWKCF}               \p{Changes_When_NFKC_Casefolded} (=
                            \p{Changes_When_NFKC_Casefolded=Y})
                            (9946)
  \p{CWKCF: *}            \p{Changes_When_NFKC_Casefolded: *}
  \p{CWL}                 \p{Changes_When_Lowercased} (=
                            \p{Changes_When_Lowercased=Y}) (1043)
  \p{CWL: *}              \p{Changes_When_Lowercased: *}
  \p{CWT}                 \p{Changes_When_Titlecased} (=
                            \p{Changes_When_Titlecased=Y}) (1099)
  \p{CWT: *}              \p{Changes_When_Titlecased: *}
  \p{CWU}                 \p{Changes_When_Uppercased} (=
                            \p{Changes_When_Uppercased=Y}) (1126)
  \p{CWU: *}              \p{Changes_When_Uppercased: *}
  \p{Cypriot}             \p{Script=Cypriot} (Short: \p{Cprt}) (55)
X \p{Cypriot_Syllabary}   \p{Block=Cypriot_Syllabary} (64)
  \p{Cyrillic}            \p{Script=Cyrillic} (Short: \p{Cyrl}; NOT
                            \p{Block=Cyrillic}) (417)
X \p{Cyrillic_Ext_A}      \p{Cyrillic_Extended_A} (= \p{Block=
                            Cyrillic_Extended_A}) (32)
X \p{Cyrillic_Ext_B}      \p{Cyrillic_Extended_B} (= \p{Block=
                            Cyrillic_Extended_B}) (96)
X \p{Cyrillic_Extended_A} \p{Block=Cyrillic_Extended_A} (Short:
                            \p{InCyrillicExtA}) (32)
X \p{Cyrillic_Extended_B} \p{Block=Cyrillic_Extended_B} (Short:
                            \p{InCyrillicExtB}) (96)
X \p{Cyrillic_Sup}        \p{Cyrillic_Supplement} (= \p{Block=
                            Cyrillic_Supplement}) (48)
X \p{Cyrillic_Supplement} \p{Block=Cyrillic_Supplement} (Short:
                            \p{InCyrillicSup}) (48)
X \p{Cyrillic_Supplementary} \p{Cyrillic_Supplement} (= \p{Block=
                            Cyrillic_Supplement}) (48)
  \p{Cyrl}                \p{Cyrillic} (= \p{Script=Cyrillic}) (NOT
                            \p{Block=Cyrillic}) (417)
  \p{Dash}                \p{Dash=Y} (27)
```

```
\p{Dash: N*}            (Single: \P{Dash}) (1_114_085 plus all
                          above-Unicode code points)
\p{Dash: Y*}            (Single: \p{Dash}) (27)
\p{Dash_Punctuation}    \p{General_Category=Dash_Punctuation}
                          (Short: \p{Pd}) (23)
\p{Decimal_Number}      \p{Digit} (= \p{General_Category=
                          Decimal_Number}) (460)
\p{Decomposition_Type: Can} \p{Decomposition_Type=Canonical}
                          (13_225)
\p{Decomposition_Type: Canonical} (Short: \p{Dt=Can}) (13_225)
\p{Decomposition_Type: Circle} (Short: \p{Dt=Enc}) (240)
\p{Decomposition_Type: Com} \p{Decomposition_Type=Compat} (720)
\p{Decomposition_Type: Compat} (Short: \p{Dt=Com}) (720)
\p{Decomposition_Type: Enc} \p{Decomposition_Type=Circle} (240)
\p{Decomposition_Type: Fin} \p{Decomposition_Type=Final} (240)
\p{Decomposition_Type: Final} (Short: \p{Dt=Fin}) (240)
\p{Decomposition_Type: Font} (Short: \p{Dt=Font}) (1184)
\p{Decomposition_Type: Fra} \p{Decomposition_Type=Fraction} (20)
\p{Decomposition_Type: Fraction} (Short: \p{Dt=Fra}) (20)
\p{Decomposition_Type: Init} \p{Decomposition_Type=Initial} (171)
\p{Decomposition_Type: Initial} (Short: \p{Dt=Init}) (171)
\p{Decomposition_Type: Iso} \p{Decomposition_Type=Isolated} (238)
\p{Decomposition_Type: Isolated} (Short: \p{Dt=Iso}) (238)
\p{Decomposition_Type: Med} \p{Decomposition_Type=Medial} (82)
\p{Decomposition_Type: Medial} (Short: \p{Dt=Med}) (82)
\p{Decomposition_Type: Nar} \p{Decomposition_Type=Narrow} (122)
\p{Decomposition_Type: Narrow} (Short: \p{Dt=Nar}) (122)
\p{Decomposition_Type: Nb} \p{Decomposition_Type=Nobreak} (5)
\p{Decomposition_Type: Nobreak} (Short: \p{Dt=Nb}) (5)
\p{Decomposition_Type: Non_Canon} \p{Decomposition_Type=
                          Non_Canonical} (Perl extension) (3655)
\p{Decomposition_Type: Non_Canonical} Union of all non-canonical
                          decompositions (Short: \p{Dt=NonCanon})
                          (Perl extension) (3655)
\p{Decomposition_Type: None} (Short: \p{Dt=None}) (1_097_232 plus
                          all above-Unicode code points)
\p{Decomposition_Type: Small} (Short: \p{Dt=Sml}) (26)
\p{Decomposition_Type: Sml} \p{Decomposition_Type=Small} (26)
\p{Decomposition_Type: Sqr} \p{Decomposition_Type=Square} (284)
\p{Decomposition_Type: Square} (Short: \p{Dt=Sqr}) (284)
\p{Decomposition_Type: Sub} (Short: \p{Dt=Sub}) (38)
\p{Decomposition_Type: Sup} \p{Decomposition_Type=Super} (146)
\p{Decomposition_Type: Super} (Short: \p{Dt=Sup}) (146)
\p{Decomposition_Type: Vert} \p{Decomposition_Type=Vertical} (35)
\p{Decomposition_Type: Vertical} (Short: \p{Dt=Vert}) (35)
\p{Decomposition_Type: Wide} (Short: \p{Dt=Wide}) (104)
\p{Default_Ignorable_Code_Point} \p{Default_Ignorable_Code_Point=
                          Y} (Short: \p{DI}) (4169)
\p{Default_Ignorable_Code_Point: N*} (Short: \p{DI=N}, \P{DI})
                          (1_109_943 plus all above-Unicode code
                          points)
\p{Default_Ignorable_Code_Point: Y*} (Short: \p{DI=Y}, \p{DI})
                          (4169)
\p{Dep}                 \p{Deprecated} (= \p{Deprecated=Y}) (111)
\p{Dep: *}              \p{Deprecated: *}
\p{Deprecated}          \p{Deprecated=Y} (Short: \p{Dep}) (111)
```

```
    \p{Deprecated: N*}      (Short: \p{Dep=N}, \P{Dep}) (1_114_001
                              plus all above-Unicode code points)
    \p{Deprecated: Y*}      (Short: \p{Dep=Y}, \p{Dep}) (111)
    \p{Deseret}             \p{Script=Deseret} (Short: \p{Dsrt}) (80)
    \p{Deva}                \p{Devanagari} (= \p{Script=Devanagari})
                              (NOT \p{Block=Devanagari}) (151)
    \p{Devanagari}          \p{Script=Devanagari} (Short: \p{Deva};
                              NOT \p{Block=Devanagari}) (151)
  X \p{Devanagari_Ext}      \p{Devanagari_Extended} (= \p{Block=
                              Devanagari_Extended}) (32)
  X \p{Devanagari_Extended} \p{Block=Devanagari_Extended} (Short:
                              \p{InDevanagariExt}) (32)
    \p{DI}                  \p{Default_Ignorable_Code_Point} (=
                              \p{Default_Ignorable_Code_Point=Y})
                              (4169)
    \p{DI: *}               \p{Default_Ignorable_Code_Point: *}
    \p{Dia}                 \p{Diacritic} (= \p{Diacritic=Y}) (693)
    \p{Dia: *}              \p{Diacritic: *}
    \p{Diacritic}           \p{Diacritic=Y} (Short: \p{Dia}) (693)
    \p{Diacritic: N*}       (Short: \p{Dia=N}, \P{Dia}) (1_113_419
                              plus all above-Unicode code points)
    \p{Diacritic: Y*}       (Short: \p{Dia=Y}, \p{Dia}) (693)
  X \p{Diacriticals}        \p{Combining_Diacritical_Marks} (=
                              \p{Block=Combining_Diacritical_Marks})
                              (112)
  X \p{Diacriticals_For_Symbols}
                              \p{Combining_Diacritical_Marks_For_-
                              Symbols} (= \p{Block=
                              Combining_Diacritical_Marks_For_-
                              Symbols}) (48)
  X \p{Diacriticals_Sup}    \p{Combining_Diacritical_Marks_Supplement}
                              (= \p{Block=
                              Combining_Diacritical_Marks_Supplement})
                              (64)
    \p{Digit}               \p{General_Category=Decimal_Number} [0-9]
                              + all other decimal digits (Short:
                              \p{Nd}) (460)
  X \p{Dingbats}            \p{Block=Dingbats} (192)
  X \p{Domino}              \p{Domino_Tiles} (= \p{Block=
                              Domino_Tiles}) (112)
  X \p{Domino_Tiles}        \p{Block=Domino_Tiles} (Short:
                              \p{InDomino}) (112)
    \p{Dsrt}                \p{Deseret} (= \p{Script=Deseret}) (80)
    \p{Dt: *}               \p{Decomposition_Type: *}
    \p{Ea: *}               \p{East_Asian_Width: *}
    \p{East_Asian_Width: A} \p{East_Asian_Width=Ambiguous} (138_746)
    \p{East_Asian_Width: Ambiguous} (Short: \p{Ea=A}) (138_746)
    \p{East_Asian_Width: F} \p{East_Asian_Width=Fullwidth} (104)
    \p{East_Asian_Width: Fullwidth} (Short: \p{Ea=F}) (104)
    \p{East_Asian_Width: H} \p{East_Asian_Width=Halfwidth} (123)
    \p{East_Asian_Width: Halfwidth} (Short: \p{Ea=H}) (123)
    \p{East_Asian_Width: N} \p{East_Asian_Width=Neutral} (801_894 plus
                              all above-Unicode code points)
    \p{East_Asian_Width: Na} \p{East_Asian_Width=Narrow} (111)
    \p{East_Asian_Width: Narrow} (Short: \p{Ea=Na}) (111)
    \p{East_Asian_Width: Neutral} (Short: \p{Ea=N}) (801_894 plus all
```

```
                              above-Unicode code points)
  \p{East_Asian_Width: W} \p{East_Asian_Width=Wide} (173_134)
  \p{East_Asian_Width: Wide} (Short: \p{Ea=W}) (173_134)
  \p{Egyp}                \p{Egyptian_Hieroglyphs} (= \p{Script=
                              Egyptian_Hieroglyphs}) (NOT \p{Block=
                              Egyptian_Hieroglyphs}) (1071)
  \p{Egyptian_Hieroglyphs} \p{Script=Egyptian_Hieroglyphs} (Short:
                              \p{Egyp}; NOT \p{Block=
                              Egyptian_Hieroglyphs}) (1071)
X \p{Emoticons}           \p{Block=Emoticons} (80)
X \p{Enclosed_Alphanum}   \p{Enclosed_Alphanumerics} (= \p{Block=
                              Enclosed_Alphanumerics}) (160)
X \p{Enclosed_Alphanum_Sup} \p{Enclosed_Alphanumeric_Supplement} (=
                              \p{Block=
                              Enclosed_Alphanumeric_Supplement}) (256)
X \p{Enclosed_Alphanumeric_Supplement} \p{Block=
                              Enclosed_Alphanumeric_Supplement}
                              (Short: \p{InEnclosedAlphanumSup}) (256)
X \p{Enclosed_Alphanumerics} \p{Block=Enclosed_Alphanumerics}
                              (Short: \p{InEnclosedAlphanum}) (160)
X \p{Enclosed_CJK}        \p{Enclosed_CJK_Letters_And_Months} (=
                              \p{Block=
                              Enclosed_CJK_Letters_And_Months}) (256)
X \p{Enclosed_CJK_Letters_And_Months} \p{Block=
                              Enclosed_CJK_Letters_And_Months} (Short:
                              \p{InEnclosedCJK}) (256)
X \p{Enclosed_Ideographic_Sup} \p{Enclosed_Ideographic_Supplement}
                              (= \p{Block=
                              Enclosed_Ideographic_Supplement}) (256)
X \p{Enclosed_Ideographic_Supplement} \p{Block=
                              Enclosed_Ideographic_Supplement} (Short:
                              \p{InEnclosedIdeographicSup}) (256)
  \p{Enclosing_Mark}      \p{General_Category=Enclosing_Mark}
                              (Short: \p{Me}) (12)
  \p{Ethi}                \p{Ethiopic} (= \p{Script=Ethiopic}) (NOT
                              \p{Block=Ethiopic}) (495)
  \p{Ethiopic}            \p{Script=Ethiopic} (Short: \p{Ethi}; NOT
                              \p{Block=Ethiopic}) (495)
X \p{Ethiopic_Ext}        \p{Ethiopic_Extended} (= \p{Block=
                              Ethiopic_Extended}) (96)
X \p{Ethiopic_Ext_A}      \p{Ethiopic_Extended_A} (= \p{Block=
                              Ethiopic_Extended_A}) (48)
X \p{Ethiopic_Extended}   \p{Block=Ethiopic_Extended} (Short:
                              \p{InEthiopicExt}) (96)
X \p{Ethiopic_Extended_A} \p{Block=Ethiopic_Extended_A} (Short:
                              \p{InEthiopicExtA}) (48)
X \p{Ethiopic_Sup}        \p{Ethiopic_Supplement} (= \p{Block=
                              Ethiopic_Supplement}) (32)
X \p{Ethiopic_Supplement} \p{Block=Ethiopic_Supplement} (Short:
                              \p{InEthiopicSup}) (32)
  \p{Ext}                 \p{Extender} (= \p{Extender=Y}) (31)
  \p{Ext: *}              \p{Extender: *}
  \p{Extender}            \p{Extender=Y} (Short: \p{Ext}) (31)
  \p{Extender: N*}        (Short: \p{Ext=N}, \P{Ext}) (1_114_081
                              plus all above-Unicode code points)
  \p{Extender: Y*}        (Short: \p{Ext=Y}, \p{Ext}) (31)
```

```
    \p{Final_Punctuation}   \p{General_Category=Final_Punctuation}
                              (Short: \p{Pf}) (10)
    \p{Format}              \p{General_Category=Format} (Short:
                              \p{Cf}) (145)
    \p{Full_Composition_Exclusion} \p{Full_Composition_Exclusion=Y}
                              (Short: \p{CompEx}) (1120)
    \p{Full_Composition_Exclusion: N*} (Short: \p{CompEx=N},
                              \P{CompEx}) (1_112_992 plus all above-
                              Unicode code points)
    \p{Full_Composition_Exclusion: Y*} (Short: \p{CompEx=Y},
                              \p{CompEx}) (1120)
    \p{Gc: *}               \p{General_Category: *}
    \p{GCB: *}              \p{Grapheme_Cluster_Break: *}
    \p{General_Category: C} \p{General_Category=Other} (1_004_135 plus
                              all above-Unicode code points)
    \p{General_Category: Cased_Letter} [\p{Ll}\p{Lu}\p{Lt}] (Short:
                              \p{Gc=LC}, \p{LC}) (3223)
    \p{General_Category: Cc} \p{General_Category=Control} (65)
    \p{General_Category: Cf} \p{General_Category=Format} (145)
    \p{General_Category: Close_Punctuation} (Short: \p{Gc=Pe}, \p{Pe})
                              (73)
    \p{General_Category: Cn} \p{General_Category=Unassigned} (864_409
                              plus all above-Unicode code points)
    \p{General_Category: Cntrl} \p{General_Category=Control} (65)
    \p{General_Category: Co} \p{General_Category=Private_Use} (137_468)
    \p{General_Category: Combining_Mark} \p{General_Category=Mark}
                              (1645)
    \p{General_Category: Connector_Punctuation} (Short: \p{Gc=Pc},
                              \p{Pc}) (10)
    \p{General_Category: Control} (Short: \p{Gc=Cc}, \p{Cc}) (65)
    \p{General_Category: Cs} \p{General_Category=Surrogate} (2048)
    \p{General_Category: Currency_Symbol} (Short: \p{Gc=Sc}, \p{Sc})
                              (49)
    \p{General_Category: Dash_Punctuation} (Short: \p{Gc=Pd}, \p{Pd})
                              (23)
    \p{General_Category: Decimal_Number} (Short: \p{Gc=Nd}, \p{Nd})
                              (460)
    \p{General_Category: Digit} \p{General_Category=Decimal_Number}
                              (460)
    \p{General_Category: Enclosing_Mark} (Short: \p{Gc=Me}, \p{Me})
                              (12)
    \p{General_Category: Final_Punctuation} (Short: \p{Gc=Pf}, \p{Pf})
                              (10)
    \p{General_Category: Format} (Short: \p{Gc=Cf}, \p{Cf}) (145)
    \p{General_Category: Initial_Punctuation} (Short: \p{Gc=Pi},
                              \p{Pi}) (12)
    \p{General_Category: L} \p{General_Category=Letter} (101_013)
  X \p{General_Category: L&} \p{General_Category=Cased_Letter} (3223)
  X \p{General_Category: L_} \p{General_Category=Cased_Letter} Note
                              the trailing '_' matters in spite of
                              loose matching rules. (3223)
    \p{General_Category: LC} \p{General_Category=Cased_Letter} (3223)
    \p{General_Category: Letter} (Short: \p{Gc=L}, \p{L}) (101_013)
    \p{General_Category: Letter_Number} (Short: \p{Gc=Nl}, \p{Nl})
                              (224)
    \p{General_Category: Line_Separator} (Short: \p{Gc=Zl}, \p{Zl}) (1)
```

```
\p{General_Category: Ll} \p{General_Category=Lowercase_Letter}
                         (/i= General_Category=Cased_Letter)
                         (1751)
\p{General_Category: Lm} \p{General_Category=Modifier_Letter} (237)
\p{General_Category: Lo} \p{General_Category=Other_Letter} (97_553)
\p{General_Category: Lowercase_Letter} (Short: \p{Gc=Ll}, \p{Ll};
                         /i= General_Category=Cased_Letter) (1751)
\p{General_Category: Lt} \p{General_Category=Titlecase_Letter}
                         (/i= General_Category=Cased_Letter) (31)
\p{General_Category: Lu} \p{General_Category=Uppercase_Letter}
                         (/i= General_Category=Cased_Letter)
                         (1441)
\p{General_Category: M} \p{General_Category=Mark} (1645)
\p{General_Category: Mark} (Short: \p{Gc=M}, \p{M}) (1645)
\p{General_Category: Math_Symbol} (Short: \p{Gc=Sm}, \p{Sm}) (948)
\p{General_Category: Mc} \p{General_Category=Spacing_Mark} (352)
\p{General_Category: Me} \p{General_Category=Enclosing_Mark} (12)
\p{General_Category: Mn} \p{General_Category=Nonspacing_Mark}
                         (1281)
\p{General_Category: Modifier_Letter} (Short: \p{Gc=Lm}, \p{Lm})
                         (237)
\p{General_Category: Modifier_Symbol} (Short: \p{Gc=Sk}, \p{Sk})
                         (115)
\p{General_Category: N} \p{General_Category=Number} (1148)
\p{General_Category: Nd} \p{General_Category=Decimal_Number} (460)
\p{General_Category: Nl} \p{General_Category=Letter_Number} (224)
\p{General_Category: No} \p{General_Category=Other_Number} (464)
\p{General_Category: Nonspacing_Mark} (Short: \p{Gc=Mn}, \p{Mn})
                         (1281)
\p{General_Category: Number} (Short: \p{Gc=N}, \p{N}) (1148)
\p{General_Category: Open_Punctuation} (Short: \p{Gc=Ps}, \p{Ps})
                         (74)
\p{General_Category: Other} (Short: \p{Gc=C}, \p{C}) (1_004_135
                         plus all above-Unicode code points)
\p{General_Category: Other_Letter} (Short: \p{Gc=Lo}, \p{Lo})
                         (97_553)
\p{General_Category: Other_Number} (Short: \p{Gc=No}, \p{No}) (464)
\p{General_Category: Other_Punctuation} (Short: \p{Gc=Po}, \p{Po})
                         (434)
\p{General_Category: Other_Symbol} (Short: \p{Gc=So}, \p{So})
                         (4404)
\p{General_Category: P} \p{General_Category=Punctuation} (636)
\p{General_Category: Paragraph_Separator} (Short: \p{Gc=Zp},
                         \p{Zp}) (1)
\p{General_Category: Pc} \p{General_Category=
                         Connector_Punctuation} (10)
\p{General_Category: Pd} \p{General_Category=Dash_Punctuation} (23)
\p{General_Category: Pe} \p{General_Category=Close_Punctuation}
                         (73)
\p{General_Category: Pf} \p{General_Category=Final_Punctuation}
                         (10)
\p{General_Category: Pi} \p{General_Category=Initial_Punctuation}
                         (12)
\p{General_Category: Po} \p{General_Category=Other_Punctuation}
                         (434)
\p{General_Category: Private_Use} (Short: \p{Gc=Co}, \p{Co})
```

```
                                (137_468)
    \p{General_Category: Ps} \p{General_Category=Open_Punctuation} (74)
    \p{General_Category: Punct} \p{General_Category=Punctuation} (636)
    \p{General_Category: Punctuation} (Short: \p{Gc=P}, \p{P}) (636)
    \p{General_Category: S} \p{General_Category=Symbol} (5516)
    \p{General_Category: Sc} \p{General_Category=Currency_Symbol} (49)
    \p{General_Category: Separator} (Short: \p{Gc=Z}, \p{Z}) (19)
    \p{General_Category: Sk} \p{General_Category=Modifier_Symbol} (115)
    \p{General_Category: Sm} \p{General_Category=Math_Symbol} (948)
    \p{General_Category: So} \p{General_Category=Other_Symbol} (4404)
    \p{General_Category: Space_Separator} (Short: \p{Gc=Zs}, \p{Zs})
                                (17)
    \p{General_Category: Spacing_Mark} (Short: \p{Gc=Mc}, \p{Mc}) (352)
    \p{General_Category: Surrogate} (Short: \p{Gc=Cs}, \p{Cs}) (2048)
    \p{General_Category: Symbol} (Short: \p{Gc=S}, \p{S}) (5516)
    \p{General_Category: Titlecase_Letter} (Short: \p{Gc=Lt}, \p{Lt};
                                /i= General_Category=Cased_Letter) (31)
    \p{General_Category: Unassigned} (Short: \p{Gc=Cn}, \p{Cn})
                                (864_409 plus all above-Unicode code
                                points)
    \p{General_Category: Uppercase_Letter} (Short: \p{Gc=Lu}, \p{Lu};
                                /i= General_Category=Cased_Letter) (1441)
    \p{General_Category: Z} \p{General_Category=Separator} (19)
    \p{General_Category: Zl} \p{General_Category=Line_Separator} (1)
    \p{General_Category: Zp} \p{General_Category=Paragraph_Separator}
                                (1)
    \p{General_Category: Zs} \p{General_Category=Space_Separator} (17)
X \p{General_Punctuation} \p{Block=General_Punctuation} (Short:
                                \p{InPunctuation}) (112)
X \p{Geometric_Shapes}    \p{Block=Geometric_Shapes} (96)
  \p{Geor}                \p{Georgian} (= \p{Script=Georgian}) (NOT
                            \p{Block=Georgian}) (127)
  \p{Georgian}            \p{Script=Georgian} (Short: \p{Geor}; NOT
                            \p{Block=Georgian}) (127)
X \p{Georgian_Sup}        \p{Georgian_Supplement} (= \p{Block=
                            Georgian_Supplement}) (48)
X \p{Georgian_Supplement} \p{Block=Georgian_Supplement} (Short:
                            \p{InGeorgianSup}) (48)
  \p{Glag}                \p{Glagolitic} (= \p{Script=Glagolitic})
                            (NOT \p{Block=Glagolitic}) (94)
  \p{Glagolitic}          \p{Script=Glagolitic} (Short: \p{Glag};
                            NOT \p{Block=Glagolitic}) (94)
  \p{Goth}                \p{Gothic} (= \p{Script=Gothic}) (NOT
                            \p{Block=Gothic}) (27)
  \p{Gothic}              \p{Script=Gothic} (Short: \p{Goth}; NOT
                            \p{Block=Gothic}) (27)
  \p{Gr_Base}             \p{Grapheme_Base} (= \p{Grapheme_Base=Y})
                            (108_659)
  \p{Gr_Base: *}          \p{Grapheme_Base: *}
  \p{Gr_Ext}              \p{Grapheme_Extend} (= \p{Grapheme_Extend=
                            Y}) (1318)
  \p{Gr_Ext: *}           \p{Grapheme_Extend: *}
  \p{Graph}               Characters that are graphical (247_571)
  \p{Grapheme_Base}       \p{Grapheme_Base=Y} (Short: \p{GrBase})
                            (108_659)
  \p{Grapheme_Base: N*}   (Short: \p{GrBase=N}, \P{GrBase})
```

```
                              (1_005_453 plus all above-Unicode code
                              points)
    \p{Grapheme_Base: Y*}   (Short: \p{GrBase=Y}, \p{GrBase}) (108_659)
    \p{Grapheme_Cluster_Break: CN} \p{Grapheme_Cluster_Break=Control}
                              (6025)
    \p{Grapheme_Cluster_Break: Control} (Short: \p{GCB=CN}) (6025)
    \p{Grapheme_Cluster_Break: CR} (Short: \p{GCB=CR}) (1)
    \p{Grapheme_Cluster_Break: EX} \p{Grapheme_Cluster_Break=Extend}
                              (1318)
    \p{Grapheme_Cluster_Break: Extend} (Short: \p{GCB=EX}) (1318)
    \p{Grapheme_Cluster_Break: L} (Short: \p{GCB=L}) (125)
    \p{Grapheme_Cluster_Break: LF} (Short: \p{GCB=LF}) (1)
    \p{Grapheme_Cluster_Break: LV} (Short: \p{GCB=LV}) (399)
    \p{Grapheme_Cluster_Break: LVT} (Short: \p{GCB=LVT}) (10_773)
    \p{Grapheme_Cluster_Break: Other} (Short: \p{GCB=XX}) (1_094_922
                              plus all above-Unicode code points)
    \p{Grapheme_Cluster_Break: PP} \p{Grapheme_Cluster_Break=Prepend}
                              (0)
    \p{Grapheme_Cluster_Break: Prepend} (Short: \p{GCB=PP}) (0)
    \p{Grapheme_Cluster_Break: Regional_Indicator} (Short: \p{GCB=RI})
                              (26)
    \p{Grapheme_Cluster_Break: RI} \p{Grapheme_Cluster_Break=
                              Regional_Indicator} (26)
    \p{Grapheme_Cluster_Break: SM} \p{Grapheme_Cluster_Break=
                              SpacingMark} (290)
    \p{Grapheme_Cluster_Break: SpacingMark} (Short: \p{GCB=SM}) (290)
    \p{Grapheme_Cluster_Break: T} (Short: \p{GCB=T}) (137)
    \p{Grapheme_Cluster_Break: V} (Short: \p{GCB=V}) (95)
    \p{Grapheme_Cluster_Break: XX} \p{Grapheme_Cluster_Break=Other}
                              (1_094_922 plus all above-Unicode code
                              points)
    \p{Grapheme_Extend}     \p{Grapheme_Extend=Y} (Short: \p{GrExt})
                              (1318)
    \p{Grapheme_Extend: N*} (Short: \p{GrExt=N}, \P{GrExt}) (1_112_794
                              plus all above-Unicode code points)
    \p{Grapheme_Extend: Y*} (Short: \p{GrExt=Y}, \p{GrExt}) (1318)
    \p{Greek}               \p{Script=Greek} (Short: \p{Grek}; NOT
                              \p{Greek_And_Coptic}) (511)
  X \p{Greek_And_Coptic}    \p{Block=Greek_And_Coptic} (Short:
                              \p{InGreek}) (144)
  X \p{Greek_Ext}           \p{Greek_Extended} (= \p{Block=
                              Greek_Extended}) (256)
  X \p{Greek_Extended}      \p{Block=Greek_Extended} (Short:
                              \p{InGreekExt}) (256)
    \p{Grek}                \p{Greek} (= \p{Script=Greek}) (NOT
                              \p{Greek_And_Coptic}) (511)
    \p{Gujarati}            \p{Script=Gujarati} (Short: \p{Gujr}; NOT
                              \p{Block=Gujarati}) (84)
    \p{Gujr}                \p{Gujarati} (= \p{Script=Gujarati}) (NOT
                              \p{Block=Gujarati}) (84)
    \p{Gurmukhi}            \p{Script=Gurmukhi} (Short: \p{Guru}; NOT
                              \p{Block=Gurmukhi}) (79)
    \p{Guru}                \p{Gurmukhi} (= \p{Script=Gurmukhi}) (NOT
                              \p{Block=Gurmukhi}) (79)
  X \p{Half_And_Full_Forms} \p{Halfwidth_And_Fullwidth_Forms} (=
                              \p{Block=Halfwidth_And_Fullwidth_Forms})
```

```
                                (240)
X \p{Half_Marks}           \p{Combining_Half_Marks} (= \p{Block=
                                Combining_Half_Marks}) (16)
X \p{Halfwidth_And_Fullwidth_Forms} \p{Block=
                                Halfwidth_And_Fullwidth_Forms} (Short:
                                \p{InHalfAndFullForms}) (240)
  \p{Han}                  \p{Script=Han} (75_963)
  \p{Hang}                 \p{Hangul} (= \p{Script=Hangul}) (NOT
                                \p{Hangul_Syllables}) (11_739)
  \p{Hangul}               \p{Script=Hangul} (Short: \p{Hang}; NOT
                                \p{Hangul_Syllables}) (11_739)
X \p{Hangul_Compatibility_Jamo} \p{Block=Hangul_Compatibility_Jamo}
                                (Short: \p{InCompatJamo}) (96)
X \p{Hangul_Jamo}          \p{Block=Hangul_Jamo} (Short: \p{InJamo})
                                (256)
X \p{Hangul_Jamo_Extended_A} \p{Block=Hangul_Jamo_Extended_A}
                                (Short: \p{InJamoExtA}) (32)
X \p{Hangul_Jamo_Extended_B} \p{Block=Hangul_Jamo_Extended_B}
                                (Short: \p{InJamoExtB}) (80)
  \p{Hangul_Syllable_Type: L} \p{Hangul_Syllable_Type=Leading_Jamo}
                                (125)
  \p{Hangul_Syllable_Type: Leading_Jamo} (Short: \p{Hst=L}) (125)
  \p{Hangul_Syllable_Type: LV} \p{Hangul_Syllable_Type=LV_Syllable}
                                (399)
  \p{Hangul_Syllable_Type: LV_Syllable} (Short: \p{Hst=LV}) (399)
  \p{Hangul_Syllable_Type: LVT} \p{Hangul_Syllable_Type=
                                LVT_Syllable} (10_773)
  \p{Hangul_Syllable_Type: LVT_Syllable} (Short: \p{Hst=LVT})
                                (10_773)
  \p{Hangul_Syllable_Type: NA} \p{Hangul_Syllable_Type=
                                Not_Applicable} (1_102_583 plus all
                                above-Unicode code points)
  \p{Hangul_Syllable_Type: Not_Applicable} (Short: \p{Hst=NA})
                                (1_102_583 plus all above-Unicode code
                                points)
  \p{Hangul_Syllable_Type: T} \p{Hangul_Syllable_Type=Trailing_Jamo}
                                (137)
  \p{Hangul_Syllable_Type: Trailing_Jamo} (Short: \p{Hst=T}) (137)
  \p{Hangul_Syllable_Type: V} \p{Hangul_Syllable_Type=Vowel_Jamo}
                                (95)
  \p{Hangul_Syllable_Type: Vowel_Jamo} (Short: \p{Hst=V}) (95)
X \p{Hangul_Syllables}     \p{Block=Hangul_Syllables} (Short:
                                \p{InHangul}) (11_184)
  \p{Hani}                 \p{Han} (= \p{Script=Han}) (75_963)
  \p{Hano}                 \p{Hanunoo} (= \p{Script=Hanunoo}) (NOT
                                \p{Block=Hanunoo}) (21)
  \p{Hanunoo}              \p{Script=Hanunoo} (Short: \p{Hano}; NOT
                                \p{Block=Hanunoo}) (21)
  \p{Hebr}                 \p{Hebrew} (= \p{Script=Hebrew}) (NOT
                                \p{Block=Hebrew}) (133)
  \p{Hebrew}               \p{Script=Hebrew} (Short: \p{Hebr}; NOT
                                \p{Block=Hebrew}) (133)
  \p{Hex}                  \p{XDigit} (= \p{Hex_Digit=Y}) (44)
  \p{Hex: *}               \p{Hex_Digit: *}
  \p{Hex_Digit}            \p{XDigit} (= \p{Hex_Digit=Y}) (44)
  \p{Hex_Digit: N*}        (Short: \p{Hex=N}, \P{Hex}) (1_114_068
```

```
                                plus all above-Unicode code points)
  \p{Hex_Digit: Y*}       (Short: \p{Hex=Y}, \p{Hex}) (44)
X \p{High_Private_Use_Surrogates} \p{Block=
                                High_Private_Use_Surrogates} (Short:
                                \p{InHighPUSurrogates}) (128)
X \p{High_PU_Surrogates}  \p{High_Private_Use_Surrogates} (=
                                \p{Block=High_Private_Use_Surrogates})
                                (128)
X \p{High_Surrogates}     \p{Block=High_Surrogates} (896)
  \p{Hira}                \p{Hiragana} (= \p{Script=Hiragana}) (NOT
                                \p{Block=Hiragana}) (91)
  \p{Hiragana}            \p{Script=Hiragana} (Short: \p{Hira}; NOT
                                \p{Block=Hiragana}) (91)
  \p{HorizSpace}          \p{Blank} (18)
  \p{Hst: *}              \p{Hangul_Syllable_Type: *}
D \p{Hyphen}              \p{Hyphen=Y} (11)
D \p{Hyphen: N*}          Supplanted by Line_Break property values;
                                see www.unicode.org/reports/tr14
                                (Single: \P{Hyphen}) (1_114_101 plus all
                                above-Unicode code points)
D \p{Hyphen: Y*}          Supplanted by Line_Break property values;
                                see www.unicode.org/reports/tr14
                                (Single: \p{Hyphen}) (11)
  \p{ID_Continue}         \p{ID_Continue=Y} (Short: \p{IDC}; NOT
                                \p{Ideographic_Description_Characters})
                                (103_355)
  \p{ID_Continue: N*}     (Short: \p{IDC=N}, \P{IDC}) (1_010_757
                                plus all above-Unicode code points)
  \p{ID_Continue: Y*}     (Short: \p{IDC=Y}, \p{IDC}) (103_355)
  \p{ID_Start}            \p{ID_Start=Y} (Short: \p{IDS}) (101_240)
  \p{ID_Start: N*}        (Short: \p{IDS=N}, \P{IDS}) (1_012_872
                                plus all above-Unicode code points)
  \p{ID_Start: Y*}        (Short: \p{IDS=Y}, \p{IDS}) (101_240)
  \p{IDC}                 \p{ID_Continue} (= \p{ID_Continue=Y}) (NOT
                                \p{Ideographic_Description_Characters})
                                (103_355)
  \p{IDC: *}              \p{ID_Continue: *}
  \p{Ideo}                \p{Ideographic} (= \p{Ideographic=Y})
                                (75_633)
  \p{Ideo: *}             \p{Ideographic: *}
  \p{Ideographic}         \p{Ideographic=Y} (Short: \p{Ideo})
                                (75_633)
  \p{Ideographic: N*}     (Short: \p{Ideo=N}, \P{Ideo}) (1_038_479
                                plus all above-Unicode code points)
  \p{Ideographic: Y*}     (Short: \p{Ideo=Y}, \p{Ideo}) (75_633)
X \p{Ideographic_Description_Characters} \p{Block=
                                Ideographic_Description_Characters}
                                (Short: \p{InIDC}) (16)
  \p{IDS}                 \p{ID_Start} (= \p{ID_Start=Y}) (101_240)
  \p{IDS: *}              \p{ID_Start: *}
  \p{IDS_Binary_Operator} \p{IDS_Binary_Operator=Y} (Short:
                                \p{IDSB}) (10)
  \p{IDS_Binary_Operator: N*} (Short: \p{IDSB=N}, \P{IDSB})
                                (1_114_102 plus all above-Unicode code
                                points)
  \p{IDS_Binary_Operator: Y*} (Short: \p{IDSB=Y}, \p{IDSB}) (10)
```

```
    \p{IDS_Trinary_Operator} \p{IDS_Trinary_Operator=Y} (Short:
                              \p{IDST}) (2)
    \p{IDS_Trinary_Operator: N*} (Short: \p{IDST=N}, \P{IDST})
                              (1_114_110 plus all above-Unicode code
                              points)
    \p{IDS_Trinary_Operator: Y*} (Short: \p{IDST=Y}, \p{IDST}) (2)
    \p{IDSB}                 \p{IDS_Binary_Operator} (=
                              \p{IDS_Binary_Operator=Y}) (10)
    \p{IDSB: *}              \p{IDS_Binary_Operator: *}
    \p{IDST}                 \p{IDS_Trinary_Operator} (=
                              \p{IDS_Trinary_Operator=Y}) (2)
    \p{IDST: *}              \p{IDS_Trinary_Operator: *}
    \p{Imperial_Aramaic}     \p{Script=Imperial_Aramaic} (Short:
                              \p{Armi}; NOT \p{Block=
                              Imperial_Aramaic}) (31)
    \p{In: *}                \p{Present_In: *} (Perl extension)
    \p{In_*}                 \p{Block: *}
  X \p{Indic_Number_Forms}   \p{Common_Indic_Number_Forms} (= \p{Block=
                              Common_Indic_Number_Forms}) (16)
    \p{Inherited}            \p{Script=Inherited} (Short: \p{Zinh})
                              (523)
    \p{Initial_Punctuation} \p{General_Category=Initial_Punctuation}
                              (Short: \p{Pi}) (12)
    \p{Inscriptional_Pahlavi} \p{Script=Inscriptional_Pahlavi} (Short:
                              \p{Phli}; NOT \p{Block=
                              Inscriptional_Pahlavi}) (27)
    \p{Inscriptional_Parthian} \p{Script=Inscriptional_Parthian}
                              (Short: \p{Prti}; NOT \p{Block=
                              Inscriptional_Parthian}) (30)
  X \p{IPA_Ext}              \p{IPA_Extensions} (= \p{Block=
                              IPA_Extensions}) (96)
  X \p{IPA_Extensions}       \p{Block=IPA_Extensions} (Short:
                              \p{InIPAExt}) (96)
    \p{Is_*}                 \p{*} (Any exceptions are individually
                              noted beginning with the word NOT.) If
                              an entry has flag(s) at its beginning,
                              like "D", the "Is_" form has the same
                              flag(s)
    \p{Ital}                 \p{Old_Italic} (= \p{Script=Old_Italic})
                              (NOT \p{Block=Old_Italic}) (35)
  X \p{Jamo}                 \p{Hangul_Jamo} (= \p{Block=Hangul_Jamo})
                              (256)
  X \p{Jamo_Ext_A}           \p{Hangul_Jamo_Extended_A} (= \p{Block=
                              Hangul_Jamo_Extended_A}) (32)
  X \p{Jamo_Ext_B}           \p{Hangul_Jamo_Extended_B} (= \p{Block=
                              Hangul_Jamo_Extended_B}) (80)
    \p{Java}                 \p{Javanese} (= \p{Script=Javanese}) (NOT
                              \p{Block=Javanese}) (90)
    \p{Javanese}             \p{Script=Javanese} (Short: \p{Java}; NOT
                              \p{Block=Javanese}) (90)
    \p{Jg: *}                \p{Joining_Group: *}
    \p{Join_C}               \p{Join_Control} (= \p{Join_Control=Y}) (2)
    \p{Join_C: *}            \p{Join_Control: *}
    \p{Join_Control}         \p{Join_Control=Y} (Short: \p{JoinC}) (2)
    \p{Join_Control: N*}     (Short: \p{JoinC=N}, \P{JoinC}) (1_114_110
                              plus all above-Unicode code points)
```

```
\p{Join_Control: Y*}    (Short: \p{JoinC=Y}, \p{JoinC}) (2)
\p{Joining_Group: Ain}  (Short: \p{Jg=Ain}) (7)
\p{Joining_Group: Alaph} (Short: \p{Jg=Alaph}) (1)
\p{Joining_Group: Alef} (Short: \p{Jg=Alef}) (10)
\p{Joining_Group: Beh}  (Short: \p{Jg=Beh}) (20)
\p{Joining_Group: Beth} (Short: \p{Jg=Beth}) (2)
\p{Joining_Group: Burushaski_Yeh_Barree} (Short: \p{Jg=
                        BurushaskiYehBarree}) (2)
\p{Joining_Group: Dal}  (Short: \p{Jg=Dal}) (14)
\p{Joining_Group: Dalath_Rish} (Short: \p{Jg=DalathRish}) (4)
\p{Joining_Group: E}    (Short: \p{Jg=E}) (1)
\p{Joining_Group: Farsi_Yeh} (Short: \p{Jg=FarsiYeh}) (7)
\p{Joining_Group: Fe}   (Short: \p{Jg=Fe}) (1)
\p{Joining_Group: Feh}  (Short: \p{Jg=Feh}) (10)
\p{Joining_Group: Final_Semkath} (Short: \p{Jg=FinalSemkath}) (1)
\p{Joining_Group: Gaf}  (Short: \p{Jg=Gaf}) (13)
\p{Joining_Group: Gamal} (Short: \p{Jg=Gamal}) (3)
\p{Joining_Group: Hah}  (Short: \p{Jg=Hah}) (18)
\p{Joining_Group: Hamza_On_Heh_Goal} (Short: \p{Jg=
                        HamzaOnHehGoal}) (1)
\p{Joining_Group: He}   (Short: \p{Jg=He}) (1)
\p{Joining_Group: Heh}  (Short: \p{Jg=Heh}) (1)
\p{Joining_Group: Heh_Goal} (Short: \p{Jg=HehGoal}) (2)
\p{Joining_Group: Heth} (Short: \p{Jg=Heth}) (1)
\p{Joining_Group: Kaf}  (Short: \p{Jg=Kaf}) (5)
\p{Joining_Group: Kaph} (Short: \p{Jg=Kaph}) (1)
\p{Joining_Group: Khaph} (Short: \p{Jg=Khaph}) (1)
\p{Joining_Group: Knotted_Heh} (Short: \p{Jg=KnottedHeh}) (2)
\p{Joining_Group: Lam}  (Short: \p{Jg=Lam}) (7)
\p{Joining_Group: Lamadh} (Short: \p{Jg=Lamadh}) (1)
\p{Joining_Group: Meem} (Short: \p{Jg=Meem}) (4)
\p{Joining_Group: Mim}  (Short: \p{Jg=Mim}) (1)
\p{Joining_Group: No_Joining_Group} (Short: \p{Jg=NoJoiningGroup})
                        (1_113_870 plus all above-Unicode code
                        points)
\p{Joining_Group: Noon} (Short: \p{Jg=Noon}) (8)
\p{Joining_Group: Nun}  (Short: \p{Jg=Nun}) (1)
\p{Joining_Group: Nya}  (Short: \p{Jg=Nya}) (1)
\p{Joining_Group: Pe}   (Short: \p{Jg=Pe}) (1)
\p{Joining_Group: Qaf}  (Short: \p{Jg=Qaf}) (5)
\p{Joining_Group: Qaph} (Short: \p{Jg=Qaph}) (1)
\p{Joining_Group: Reh}  (Short: \p{Jg=Reh}) (17)
\p{Joining_Group: Reversed_Pe} (Short: \p{Jg=ReversedPe}) (1)
\p{Joining_Group: Rohingya_Yeh} (Short: \p{Jg=RohingyaYeh}) (1)
\p{Joining_Group: Sad}  (Short: \p{Jg=Sad}) (5)
\p{Joining_Group: Sadhe} (Short: \p{Jg=Sadhe}) (1)
\p{Joining_Group: Seen} (Short: \p{Jg=Seen}) (11)
\p{Joining_Group: Semkath} (Short: \p{Jg=Semkath}) (1)
\p{Joining_Group: Shin} (Short: \p{Jg=Shin}) (1)
\p{Joining_Group: Swash_Kaf} (Short: \p{Jg=SwashKaf}) (1)
\p{Joining_Group: Syriac_Waw} (Short: \p{Jg=SyriacWaw}) (1)
\p{Joining_Group: Tah}  (Short: \p{Jg=Tah}) (4)
\p{Joining_Group: Taw}  (Short: \p{Jg=Taw}) (1)
\p{Joining_Group: Teh_Marbuta} (Short: \p{Jg=TehMarbuta}) (3)
\p{Joining_Group: Teh_Marbuta_Goal} \p{Joining_Group=
                        Hamza_On_Heh_Goal} (1)
```

```
    \p{Joining_Group: Teth} (Short: \p{Jg=Teth}) (2)
    \p{Joining_Group: Waw}  (Short: \p{Jg=Waw}) (16)
    \p{Joining_Group: Yeh}  (Short: \p{Jg=Yeh}) (10)
    \p{Joining_Group: Yeh_Barree} (Short: \p{Jg=YehBarree}) (2)
    \p{Joining_Group: Yeh_With_Tail} (Short: \p{Jg=YehWithTail}) (1)
    \p{Joining_Group: Yudh} (Short: \p{Jg=Yudh}) (1)
    \p{Joining_Group: Yudh_He} (Short: \p{Jg=YudhHe}) (1)
    \p{Joining_Group: Zain} (Short: \p{Jg=Zain}) (1)
    \p{Joining_Group: Zhain} (Short: \p{Jg=Zhain}) (1)
    \p{Joining_Type: C}     \p{Joining_Type=Join_Causing} (4)
    \p{Joining_Type: D}     \p{Joining_Type=Dual_Joining} (389)
    \p{Joining_Type: Dual_Joining} (Short: \p{Jt=D}) (389)
    \p{Joining_Type: Join_Causing} (Short: \p{Jt=C}) (4)
    \p{Joining_Type: L}     \p{Joining_Type=Left_Joining} (1)
    \p{Joining_Type: Left_Joining} (Short: \p{Jt=L}) (1)
    \p{Joining_Type: Non_Joining} (Short: \p{Jt=U}) (1_112_211 plus
                              all above-Unicode code points)
    \p{Joining_Type: R}     \p{Joining_Type=Right_Joining} (82)
    \p{Joining_Type: Right_Joining} (Short: \p{Jt=R}) (82)
    \p{Joining_Type: T}     \p{Joining_Type=Transparent} (1425)
    \p{Joining_Type: Transparent} (Short: \p{Jt=T}) (1425)
    \p{Joining_Type: U}     \p{Joining_Type=Non_Joining} (1_112_211
                              plus all above-Unicode code points)
    \p{Jt: *}               \p{Joining_Type: *}
    \p{Kaithi}              \p{Script=Kaithi} (Short: \p{Kthi}; NOT
                              \p{Block=Kaithi}) (66)
    \p{Kali}                \p{Kayah_Li} (= \p{Script=Kayah_Li}) (48)
    \p{Kana}                \p{Katakana} (= \p{Script=Katakana}) (NOT
                              \p{Block=Katakana}) (300)
X   \p{Kana_Sup}            \p{Kana_Supplement} (= \p{Block=
                              Kana_Supplement}) (256)
X   \p{Kana_Supplement}     \p{Block=Kana_Supplement} (Short:
                              \p{InKanaSup}) (256)
X   \p{Kanbun}              \p{Block=Kanbun} (16)
X   \p{Kangxi}              \p{Kangxi_Radicals} (= \p{Block=
                              Kangxi_Radicals}) (224)
X   \p{Kangxi_Radicals}     \p{Block=Kangxi_Radicals} (Short:
                              \p{InKangxi}) (224)
    \p{Kannada}             \p{Script=Kannada} (Short: \p{Knda}; NOT
                              \p{Block=Kannada}) (86)
    \p{Katakana}            \p{Script=Katakana} (Short: \p{Kana}; NOT
                              \p{Block=Katakana}) (300)
X   \p{Katakana_Ext}        \p{Katakana_Phonetic_Extensions} (=
                              \p{Block=Katakana_Phonetic_Extensions})
                              (16)
X   \p{Katakana_Phonetic_Extensions} \p{Block=
                              Katakana_Phonetic_Extensions} (Short:
                              \p{InKatakanaExt}) (16)
    \p{Kayah_Li}            \p{Script=Kayah_Li} (Short: \p{Kali}) (48)
    \p{Khar}                \p{Kharoshthi} (= \p{Script=Kharoshthi})
                              (NOT \p{Block=Kharoshthi}) (65)
    \p{Kharoshthi}          \p{Script=Kharoshthi} (Short: \p{Khar};
                              NOT \p{Block=Kharoshthi}) (65)
    \p{Khmer}               \p{Script=Khmer} (Short: \p{Khmr}; NOT
                              \p{Block=Khmer}) (146)
X   \p{Khmer_Symbols}       \p{Block=Khmer_Symbols} (32)
```

```
    \p{Khmr}                \p{Khmer} (= \p{Script=Khmer}) (NOT
                              \p{Block=Khmer}) (146)
    \p{Knda}                \p{Kannada} (= \p{Script=Kannada}) (NOT
                              \p{Block=Kannada}) (86)
    \p{Kthi}                \p{Kaithi} (= \p{Script=Kaithi}) (NOT
                              \p{Block=Kaithi}) (66)
    \p{L} \pL               \p{Letter} (= \p{General_Category=Letter})
                              (101_013)
  X \p{L&}                  \p{Cased_Letter} (= \p{General_Category=
                              Cased_Letter}) (3223)
  X \p{L_}                  \p{Cased_Letter} (= \p{General_Category=
                              Cased_Letter}) Note the trailing '_'
                              matters in spite of loose matching
                              rules. (3223)
    \p{Lana}                \p{Tai_Tham} (= \p{Script=Tai_Tham}) (NOT
                              \p{Block=Tai_Tham}) (127)
    \p{Lao}                 \p{Script=Lao} (NOT \p{Block=Lao}) (67)
    \p{Laoo}                \p{Lao} (= \p{Script=Lao}) (NOT \p{Block=
                              Lao}) (67)
    \p{Latin}               \p{Script=Latin} (Short: \p{Latn}) (1272)
  X \p{Latin_1}             \p{Latin_1_Supplement} (= \p{Block=
                              Latin_1_Supplement}) (128)
  X \p{Latin_1_Sup}         \p{Latin_1_Supplement} (= \p{Block=
                              Latin_1_Supplement}) (128)
  X \p{Latin_1_Supplement}  \p{Block=Latin_1_Supplement} (Short:
                              \p{InLatin1}) (128)
  X \p{Latin_Ext_A}         \p{Latin_Extended_A} (= \p{Block=
                              Latin_Extended_A}) (128)
  X \p{Latin_Ext_Additional} \p{Latin_Extended_Additional} (=
                              \p{Block=Latin_Extended_Additional})
                              (256)
  X \p{Latin_Ext_B}         \p{Latin_Extended_B} (= \p{Block=
                              Latin_Extended_B}) (208)
  X \p{Latin_Ext_C}         \p{Latin_Extended_C} (= \p{Block=
                              Latin_Extended_C}) (32)
  X \p{Latin_Ext_D}         \p{Latin_Extended_D} (= \p{Block=
                              Latin_Extended_D}) (224)
  X \p{Latin_Extended_A}    \p{Block=Latin_Extended_A} (Short:
                              \p{InLatinExtA}) (128)
  X \p{Latin_Extended_Additional} \p{Block=Latin_Extended_Additional}
                              (Short: \p{InLatinExtAdditional}) (256)
  X \p{Latin_Extended_B}    \p{Block=Latin_Extended_B} (Short:
                              \p{InLatinExtB}) (208)
  X \p{Latin_Extended_C}    \p{Block=Latin_Extended_C} (Short:
                              \p{InLatinExtC}) (32)
  X \p{Latin_Extended_D}    \p{Block=Latin_Extended_D} (Short:
                              \p{InLatinExtD}) (224)
    \p{Latn}                \p{Latin} (= \p{Script=Latin}) (1272)
    \p{Lb: *}               \p{Line_Break: *}
    \p{LC}                  \p{Cased_Letter} (= \p{General_Category=
                              Cased_Letter}) (3223)
    \p{Lepc}                \p{Lepcha} (= \p{Script=Lepcha}) (NOT
                              \p{Block=Lepcha}) (74)
    \p{Lepcha}              \p{Script=Lepcha} (Short: \p{Lepc}; NOT
                              \p{Block=Lepcha}) (74)
    \p{Letter}              \p{General_Category=Letter} (Short: \p{L})
```

```
                              (101_013)
  \p{Letter_Number}         \p{General_Category=Letter_Number} (Short:
                              \p{Nl}) (224)
X \p{Letterlike_Symbols}    \p{Block=Letterlike_Symbols} (80)
  \p{Limb}                  \p{Limbu} (= \p{Script=Limbu}) (NOT
                              \p{Block=Limbu}) (66)
  \p{Limbu}                 \p{Script=Limbu} (Short: \p{Limb}; NOT
                              \p{Block=Limbu}) (66)
  \p{Linb}                  \p{Linear_B} (= \p{Script=Linear_B}) (211)
  \p{Line_Break: AI}        \p{Line_Break=Ambiguous} (687)
  \p{Line_Break: AL}        \p{Line_Break=Alphabetic} (15_355)
  \p{Line_Break: Alphabetic} (Short: \p{Lb=AL}) (15_355)
  \p{Line_Break: Ambiguous} (Short: \p{Lb=AI}) (687)
  \p{Line_Break: B2}        \p{Line_Break=Break_Both} (3)
  \p{Line_Break: BA}        \p{Line_Break=Break_After} (152)
  \p{Line_Break: BB}        \p{Line_Break=Break_Before} (19)
  \p{Line_Break: BK}        \p{Line_Break=Mandatory_Break} (4)
  \p{Line_Break: Break_After} (Short: \p{Lb=BA}) (152)
  \p{Line_Break: Break_Before} (Short: \p{Lb=BB}) (19)
  \p{Line_Break: Break_Both} (Short: \p{Lb=B2}) (3)
  \p{Line_Break: Break_Symbols} (Short: \p{Lb=SY}) (1)
  \p{Line_Break: Carriage_Return} (Short: \p{Lb=CR}) (1)
  \p{Line_Break: CB}        \p{Line_Break=Contingent_Break} (1)
  \p{Line_Break: CJ}        \p{Line_Break=
                              Conditional_Japanese_Starter} (51)
  \p{Line_Break: CL}        \p{Line_Break=Close_Punctuation} (87)
  \p{Line_Break: Close_Parenthesis} (Short: \p{Lb=CP}) (2)
  \p{Line_Break: Close_Punctuation} (Short: \p{Lb=CL}) (87)
  \p{Line_Break: CM}        \p{Line_Break=Combining_Mark} (1634)
  \p{Line_Break: Combining_Mark} (Short: \p{Lb=CM}) (1634)
  \p{Line_Break: Complex_Context} (Short: \p{Lb=SA}) (665)
  \p{Line_Break: Conditional_Japanese_Starter} (Short: \p{Lb=CJ})
                              (51)
  \p{Line_Break: Contingent_Break} (Short: \p{Lb=CB}) (1)
  \p{Line_Break: CP}        \p{Line_Break=Close_Parenthesis} (2)
  \p{Line_Break: CR}        \p{Line_Break=Carriage_Return} (1)
  \p{Line_Break: EX}        \p{Line_Break=Exclamation} (34)
  \p{Line_Break: Exclamation} (Short: \p{Lb=EX}) (34)
  \p{Line_Break: GL}        \p{Line_Break=Glue} (18)
  \p{Line_Break: Glue}      (Short: \p{Lb=GL}) (18)
  \p{Line_Break: H2}        (Short: \p{Lb=H2}) (399)
  \p{Line_Break: H3}        (Short: \p{Lb=H3}) (10_773)
  \p{Line_Break: Hebrew_Letter} (Short: \p{Lb=HL}) (74)
  \p{Line_Break: HL}        \p{Line_Break=Hebrew_Letter} (74)
  \p{Line_Break: HY}        \p{Line_Break=Hyphen} (1)
  \p{Line_Break: Hyphen}    (Short: \p{Lb=HY}) (1)
  \p{Line_Break: ID}        \p{Line_Break=Ideographic} (162_698)
  \p{Line_Break: Ideographic} (Short: \p{Lb=ID}) (162_698)
  \p{Line_Break: IN}        \p{Line_Break=Inseparable} (4)
  \p{Line_Break: Infix_Numeric} (Short: \p{Lb=IS}) (13)
  \p{Line_Break: Inseparable} (Short: \p{Lb=IN}) (4)
  \p{Line_Break: Inseperable} \p{Line_Break=Inseparable} (4)
  \p{Line_Break: IS}        \p{Line_Break=Infix_Numeric} (13)
  \p{Line_Break: JL}        (Short: \p{Lb=JL}) (125)
  \p{Line_Break: JT}        (Short: \p{Lb=JT}) (137)
  \p{Line_Break: JV}        (Short: \p{Lb=JV}) (95)
```

```
  \p{Line_Break: LF}      \p{Line_Break=Line_Feed} (1)
  \p{Line_Break: Line_Feed} (Short: \p{Lb=LF}) (1)
  \p{Line_Break: Mandatory_Break} (Short: \p{Lb=BK}) (4)
  \p{Line_Break: Next_Line} (Short: \p{Lb=NL}) (1)
  \p{Line_Break: NL}      \p{Line_Break=Next_Line} (1)
  \p{Line_Break: Nonstarter} (Short: \p{Lb=NS}) (26)
  \p{Line_Break: NS}      \p{Line_Break=Nonstarter} (26)
  \p{Line_Break: NU}      \p{Line_Break=Numeric} (452)
  \p{Line_Break: Numeric} (Short: \p{Lb=NU}) (452)
  \p{Line_Break: OP}      \p{Line_Break=Open_Punctuation} (81)
  \p{Line_Break: Open_Punctuation} (Short: \p{Lb=OP}) (81)
  \p{Line_Break: PO}      \p{Line_Break=Postfix_Numeric} (28)
  \p{Line_Break: Postfix_Numeric} (Short: \p{Lb=PO}) (28)
  \p{Line_Break: PR}      \p{Line_Break=Prefix_Numeric} (67)
  \p{Line_Break: Prefix_Numeric} (Short: \p{Lb=PR}) (67)
  \p{Line_Break: QU}      \p{Line_Break=Quotation} (34)
  \p{Line_Break: Quotation} (Short: \p{Lb=QU}) (34)
  \p{Line_Break: Regional_Indicator} (Short: \p{Lb=RI}) (26)
  \p{Line_Break: RI}      \p{Line_Break=Regional_Indicator} (26)
  \p{Line_Break: SA}      \p{Line_Break=Complex_Context} (665)
D \p{Line_Break: SG}      \p{Line_Break=Surrogate} (2048)
  \p{Line_Break: SP}      \p{Line_Break=Space} (1)
  \p{Line_Break: Space}   (Short: \p{Lb=SP}) (1)
D \p{Line_Break: Surrogate} Deprecated by Unicode because surrogates
                            should never appear in well-formed text,
                            and therefore shouldn't be the basis for
                            line breaking (Short: \p{Lb=SG}) (2048)
  \p{Line_Break: SY}      \p{Line_Break=Break_Symbols} (1)
  \p{Line_Break: Unknown} (Short: \p{Lb=XX}) (918_311 plus all
                            above-Unicode code points)
  \p{Line_Break: WJ}      \p{Line_Break=Word_Joiner} (2)
  \p{Line_Break: Word_Joiner} (Short: \p{Lb=WJ}) (2)
  \p{Line_Break: XX}      \p{Line_Break=Unknown} (918_311 plus all
                            above-Unicode code points)
  \p{Line_Break: ZW}      \p{Line_Break=ZWSpace} (1)
  \p{Line_Break: ZWSpace} (Short: \p{Lb=ZW}) (1)
  \p{Line_Separator}      \p{General_Category=Line_Separator}
                            (Short: \p{Zl}) (1)
  \p{Linear_B}            \p{Script=Linear_B} (Short: \p{Linb}) (211)
X \p{Linear_B_Ideograms}  \p{Block=Linear_B_Ideograms} (128)
X \p{Linear_B_Syllabary}  \p{Block=Linear_B_Syllabary} (128)
  \p{Lisu}                \p{Script=Lisu} (48)
  \p{Ll}                  \p{Lowercase_Letter} (=
                            \p{General_Category=Lowercase_Letter})
                            (/i= General_Category=Cased_Letter)
                            (1751)
  \p{Lm}                  \p{Modifier_Letter} (=
                            \p{General_Category=Modifier_Letter})
                            (237)
  \p{Lo}                  \p{Other_Letter} (= \p{General_Category=
                            Other_Letter}) (97_553)
  \p{LOE}                 \p{Logical_Order_Exception} (=
                            \p{Logical_Order_Exception=Y}) (15)
  \p{LOE: *}              \p{Logical_Order_Exception: *}
  \p{Logical_Order_Exception} \p{Logical_Order_Exception=Y} (Short:
                            \p{LOE}) (15)
```

```
  \p{Logical_Order_Exception: N*} (Short: \p{LOE=N}, \P{LOE})
                              (1_114_097 plus all above-Unicode code
                              points)
  \p{Logical_Order_Exception: Y*} (Short: \p{LOE=Y}, \p{LOE}) (15)
X \p{Low_Surrogates}        \p{Block=Low_Surrogates} (1024)
  \p{Lower}                 \p{Lowercase=Y} (/i= Cased=Yes) (1934)
  \p{Lower: *}              \p{Lowercase: *}
  \p{Lowercase}             \p{Lower} (= \p{Lowercase=Y}) (/i= Cased=
                              Yes) (1934)
  \p{Lowercase: N*}         (Short: \p{Lower=N}, \P{Lower}; /i= Cased=
                              No) (1_112_178 plus all above-Unicode
                              code points)
  \p{Lowercase: Y*}         (Short: \p{Lower=Y}, \p{Lower}; /i= Cased=
                              Yes) (1934)
  \p{Lowercase_Letter}      \p{General_Category=Lowercase_Letter}
                              (Short: \p{Ll}; /i= General_Category=
                              Cased_Letter) (1751)
  \p{Lt}                    \p{Titlecase_Letter} (=
                              \p{General_Category=Titlecase_Letter})
                              (/i= General_Category=Cased_Letter) (31)
  \p{Lu}                    \p{Uppercase_Letter} (=
                              \p{General_Category=Uppercase_Letter})
                              (/i= General_Category=Cased_Letter)
                              (1441)
  \p{Lyci}                  \p{Lycian} (= \p{Script=Lycian}) (NOT
                              \p{Block=Lycian}) (29)
  \p{Lycian}                \p{Script=Lycian} (Short: \p{Lyci}; NOT
                              \p{Block=Lycian}) (29)
  \p{Lydi}                  \p{Lydian} (= \p{Script=Lydian}) (NOT
                              \p{Block=Lydian}) (27)
  \p{Lydian}                \p{Script=Lydian} (Short: \p{Lydi}; NOT
                              \p{Block=Lydian}) (27)
  \p{M} \pM                 \p{Mark} (= \p{General_Category=Mark})
                              (1645)
X \p{Mahjong}               \p{Mahjong_Tiles} (= \p{Block=
                              Mahjong_Tiles}) (48)
X \p{Mahjong_Tiles}         \p{Block=Mahjong_Tiles} (Short:
                              \p{InMahjong}) (48)
  \p{Malayalam}             \p{Script=Malayalam} (Short: \p{Mlym}; NOT
                              \p{Block=Malayalam}) (98)
  \p{Mand}                  \p{Mandaic} (= \p{Script=Mandaic}) (NOT
                              \p{Block=Mandaic}) (29)
  \p{Mandaic}               \p{Script=Mandaic} (Short: \p{Mand}; NOT
                              \p{Block=Mandaic}) (29)
  \p{Mark}                  \p{General_Category=Mark} (Short: \p{M})
                              (1645)
  \p{Math}                  \p{Math=Y} (2310)
  \p{Math: N*}              (Single: \P{Math}) (1_111_802 plus all
                              above-Unicode code points)
  \p{Math: Y*}              (Single: \p{Math}) (2310)
X \p{Math_Alphanum}         \p{Mathematical_Alphanumeric_Symbols} (=
                              \p{Block=
                              Mathematical_Alphanumeric_Symbols})
                              (1024)
X \p{Math_Operators}        \p{Mathematical_Operators} (= \p{Block=
                              Mathematical_Operators}) (256)
```

```
  \p{Math_Symbol}         \p{General_Category=Math_Symbol} (Short:
                            \p{Sm}) (948)
X \p{Mathematical_Alphanumeric_Symbols} \p{Block=
                            Mathematical_Alphanumeric_Symbols}
                            (Short: \p{InMathAlphanum}) (1024)
X \p{Mathematical_Operators} \p{Block=Mathematical_Operators}
                            (Short: \p{InMathOperators}) (256)
  \p{Mc}                  \p{Spacing_Mark} (= \p{General_Category=
                            Spacing_Mark}) (352)
  \p{Me}                  \p{Enclosing_Mark} (= \p{General_Category=
                            Enclosing_Mark}) (12)
  \p{Meetei_Mayek}        \p{Script=Meetei_Mayek} (Short: \p{Mtei};
                            NOT \p{Block=Meetei_Mayek}) (79)
X \p{Meetei_Mayek_Ext}    \p{Meetei_Mayek_Extensions} (= \p{Block=
                            Meetei_Mayek_Extensions}) (32)
X \p{Meetei_Mayek_Extensions} \p{Block=Meetei_Mayek_Extensions}
                            (Short: \p{InMeeteiMayekExt}) (32)
  \p{Merc}                \p{Meroitic_Cursive} (= \p{Script=
                            Meroitic_Cursive}) (NOT \p{Block=
                            Meroitic_Cursive}) (26)
  \p{Mero}                \p{Meroitic_Hieroglyphs} (= \p{Script=
                            Meroitic_Hieroglyphs}) (32)
  \p{Meroitic_Cursive}    \p{Script=Meroitic_Cursive} (Short:
                            \p{Merc}; NOT \p{Block=
                            Meroitic_Cursive}) (26)
  \p{Meroitic_Hieroglyphs} \p{Script=Meroitic_Hieroglyphs} (Short:
                            \p{Mero}) (32)
  \p{Miao}                \p{Script=Miao} (NOT \p{Block=Miao}) (133)
X \p{Misc_Arrows}         \p{Miscellaneous_Symbols_And_Arrows} (=
                            \p{Block=
                            Miscellaneous_Symbols_And_Arrows}) (256)
X \p{Misc_Math_Symbols_A} \p{Miscellaneous_Mathematical_Symbols_A}
                            (= \p{Block=
                            Miscellaneous_Mathematical_Symbols_A})
                            (48)
X \p{Misc_Math_Symbols_B} \p{Miscellaneous_Mathematical_Symbols_B}
                            (= \p{Block=
                            Miscellaneous_Mathematical_Symbols_B})
                            (128)
X \p{Misc_Pictographs}    \p{Miscellaneous_Symbols_And_Pictographs}
                            (= \p{Block=
                            Miscellaneous_Symbols_And_Pictographs})
                            (768)
X \p{Misc_Symbols}        \p{Miscellaneous_Symbols} (= \p{Block=
                            Miscellaneous_Symbols}) (256)
X \p{Misc_Technical}      \p{Miscellaneous_Technical} (= \p{Block=
                            Miscellaneous_Technical}) (256)
X \p{Miscellaneous_Mathematical_Symbols_A} \p{Block=
                            Miscellaneous_Mathematical_Symbols_A}
                            (Short: \p{InMiscMathSymbolsA}) (48)
X \p{Miscellaneous_Mathematical_Symbols_B} \p{Block=
                            Miscellaneous_Mathematical_Symbols_B}
                            (Short: \p{InMiscMathSymbolsB}) (128)
X \p{Miscellaneous_Symbols} \p{Block=Miscellaneous_Symbols} (Short:
                            \p{InMiscSymbols}) (256)
X \p{Miscellaneous_Symbols_And_Arrows} \p{Block=
```

```
                              Miscellaneous_Symbols_And_Arrows}
                              (Short: \p{InMiscArrows}) (256)
X \p{Miscellaneous_Symbols_And_Pictographs} \p{Block=
                              Miscellaneous_Symbols_And_Pictographs}
                              (Short: \p{InMiscPictographs}) (768)
X \p{Miscellaneous_Technical} \p{Block=Miscellaneous_Technical}
                              (Short: \p{InMiscTechnical}) (256)
  \p{Mlym}                  \p{Malayalam} (= \p{Script=Malayalam})
                              (NOT \p{Block=Malayalam}) (98)
  \p{Mn}                    \p{Nonspacing_Mark} (=
                              \p{General_Category=Nonspacing_Mark})
                              (1281)
  \p{Modifier_Letter}       \p{General_Category=Modifier_Letter}
                              (Short: \p{Lm}) (237)
X \p{Modifier_Letters}      \p{Spacing_Modifier_Letters} (= \p{Block=
                              Spacing_Modifier_Letters}) (80)
  \p{Modifier_Symbol}       \p{General_Category=Modifier_Symbol}
                              (Short: \p{Sk}) (115)
X \p{Modifier_Tone_Letters} \p{Block=Modifier_Tone_Letters} (32)
  \p{Mong}                  \p{Mongolian} (= \p{Script=Mongolian})
                              (NOT \p{Block=Mongolian}) (153)
  \p{Mongolian}             \p{Script=Mongolian} (Short: \p{Mong}; NOT
                              \p{Block=Mongolian}) (153)
  \p{Mtei}                  \p{Meetei_Mayek} (= \p{Script=
                              Meetei_Mayek}) (NOT \p{Block=
                              Meetei_Mayek}) (79)
X \p{Music}                 \p{Musical_Symbols} (= \p{Block=
                              Musical_Symbols}) (256)
X \p{Musical_Symbols}       \p{Block=Musical_Symbols} (Short:
                              \p{InMusic}) (256)
  \p{Myanmar}               \p{Script=Myanmar} (Short: \p{Mymr}; NOT
                              \p{Block=Myanmar}) (188)
X \p{Myanmar_Ext_A}         \p{Myanmar_Extended_A} (= \p{Block=
                              Myanmar_Extended_A}) (32)
X \p{Myanmar_Extended_A}    \p{Block=Myanmar_Extended_A} (Short:
                              \p{InMyanmarExtA}) (32)
  \p{Mymr}                  \p{Myanmar} (= \p{Script=Myanmar}) (NOT
                              \p{Block=Myanmar}) (188)
  \p{N} \pN                 \p{Number} (= \p{General_Category=Number})
                              (1148)
X \p{NB}                    \p{No_Block} (= \p{Block=No_Block})
                              (860_672 plus all above-Unicode code
                              points)
  \p{NChar}                 \p{Noncharacter_Code_Point} (=
                              \p{Noncharacter_Code_Point=Y}) (66)
  \p{NChar: *}              \p{Noncharacter_Code_Point: *}
  \p{Nd}                    \p{Digit} (= \p{General_Category=
                              Decimal_Number}) (460)
  \p{New_Tai_Lue}           \p{Script=New_Tai_Lue} (Short: \p{Talu};
                              NOT \p{Block=New_Tai_Lue}) (83)
  \p{NFC_QC: *}             \p{NFC_Quick_Check: *}
  \p{NFC_Quick_Check: M}    \p{NFC_Quick_Check=Maybe} (104)
  \p{NFC_Quick_Check: Maybe} (Short: \p{NFCQC=M}) (104)
  \p{NFC_Quick_Check: N}    \p{NFC_Quick_Check=No} (NOT
                              \P{NFC_Quick_Check} NOR \P{NFC_QC})
                              (1120)
```

```
\p{NFC_Quick_Check: No} (Short: \p{NFCQC=N}; NOT
                          \P{NFC_Quick_Check} NOR \P{NFC_QC})
                          (1120)
\p{NFC_Quick_Check: Y}  \p{NFC_Quick_Check=Yes} (NOT
                          \p{NFC_Quick_Check} NOR \p{NFC_QC})
                          (1_112_888 plus all above-Unicode code
                          points)
\p{NFC_Quick_Check: Yes} (Short: \p{NFCQC=Y}; NOT
                          \p{NFC_Quick_Check} NOR \p{NFC_QC})
                          (1_112_888 plus all above-Unicode code
                          points)
\p{NFD_QC: *}           \p{NFD_Quick_Check: *}
\p{NFD_Quick_Check: N}  \p{NFD_Quick_Check=No} (NOT
                          \P{NFD_Quick_Check} NOR \P{NFD_QC})
                          (13_225)
\p{NFD_Quick_Check: No} (Short: \p{NFDQC=N}; NOT
                          \P{NFD_Quick_Check} NOR \P{NFD_QC})
                          (13_225)
\p{NFD_Quick_Check: Y}  \p{NFD_Quick_Check=Yes} (NOT
                          \p{NFD_Quick_Check} NOR \p{NFD_QC})
                          (1_100_887 plus all above-Unicode code
                          points)
\p{NFD_Quick_Check: Yes} (Short: \p{NFDQC=Y}; NOT
                          \p{NFD_Quick_Check} NOR \p{NFD_QC})
                          (1_100_887 plus all above-Unicode code
                          points)
\p{NFKC_QC: *}          \p{NFKC_Quick_Check: *}
\p{NFKC_Quick_Check: M} \p{NFKC_Quick_Check=Maybe} (104)
\p{NFKC_Quick_Check: Maybe} (Short: \p{NFKCQC=M}) (104)
\p{NFKC_Quick_Check: N} \p{NFKC_Quick_Check=No} (NOT
                          \P{NFKC_Quick_Check} NOR \P{NFKC_QC})
                          (4787)
\p{NFKC_Quick_Check: No} (Short: \p{NFKCQC=N}; NOT
                          \P{NFKC_Quick_Check} NOR \P{NFKC_QC})
                          (4787)
\p{NFKC_Quick_Check: Y} \p{NFKC_Quick_Check=Yes} (NOT
                          \p{NFKC_Quick_Check} NOR \p{NFKC_QC})
                          (1_109_221 plus all above-Unicode code
                          points)
\p{NFKC_Quick_Check: Yes} (Short: \p{NFKCQC=Y}; NOT
                          \p{NFKC_Quick_Check} NOR \p{NFKC_QC})
                          (1_109_221 plus all above-Unicode code
                          points)
\p{NFKD_QC: *}          \p{NFKD_Quick_Check: *}
\p{NFKD_Quick_Check: N} \p{NFKD_Quick_Check=No} (NOT
                          \P{NFKD_Quick_Check} NOR \P{NFKD_QC})
                          (16_880)
\p{NFKD_Quick_Check: No} (Short: \p{NFKDQC=N}; NOT
                          \P{NFKD_Quick_Check} NOR \P{NFKD_QC})
                          (16_880)
\p{NFKD_Quick_Check: Y} \p{NFKD_Quick_Check=Yes} (NOT
                          \p{NFKD_Quick_Check} NOR \p{NFKD_QC})
                          (1_097_232 plus all above-Unicode code
                          points)
\p{NFKD_Quick_Check: Yes} (Short: \p{NFKDQC=Y}; NOT
                          \p{NFKD_Quick_Check} NOR \p{NFKD_QC})
```

```
                                (1_097_232 plus all above-Unicode code
                                points)
  \p{Nko}                     \p{Script=Nko} (NOT \p{NKo}) (59)
  \p{Nkoo}                    \p{Nko} (= \p{Script=Nko}) (NOT \p{NKo})
                                (59)
  \p{Nl}                      \p{Letter_Number} (= \p{General_Category=
                                Letter_Number}) (224)
  \p{No}                      \p{Other_Number} (= \p{General_Category=
                                Other_Number}) (464)
X \p{No_Block}                \p{Block=No_Block} (Short: \p{InNB})
                                (860_672 plus all above-Unicode code
                                points)
  \p{Noncharacter_Code_Point} \p{Noncharacter_Code_Point=Y} (Short:
                                \p{NChar}) (66)
  \p{Noncharacter_Code_Point: N*} (Short: \p{NChar=N}, \P{NChar})
                                (1_114_046 plus all above-Unicode code
                                points)
  \p{Noncharacter_Code_Point: Y*} (Short: \p{NChar=Y}, \p{NChar})
                                (66)
  \p{Nonspacing_Mark}         \p{General_Category=Nonspacing_Mark}
                                (Short: \p{Mn}) (1281)
  \p{Nt: *}                   \p{Numeric_Type: *}
  \p{Number}                  \p{General_Category=Number} (Short: \p{N})
                                (1148)
X \p{Number_Forms}            \p{Block=Number_Forms} (64)
  \p{Numeric_Type: De}        \p{Numeric_Type=Decimal} (460)
  \p{Numeric_Type: Decimal} (Short: \p{Nt=De}) (460)
  \p{Numeric_Type: Di}        \p{Numeric_Type=Digit} (128)
  \p{Numeric_Type: Digit} (Short: \p{Nt=Di}) (128)
  \p{Numeric_Type: None}  (Short: \p{Nt=None}) (1_112_883 plus all
                                above-Unicode code points)
  \p{Numeric_Type: Nu}        \p{Numeric_Type=Numeric} (641)
  \p{Numeric_Type: Numeric} (Short: \p{Nt=Nu}) (641)
T \p{Numeric_Value: -1/2} (Short: \p{Nv=-1/2}) (1)
T \p{Numeric_Value: 0}    (Short: \p{Nv=0}) (60)
T \p{Numeric_Value: 1/16} (Short: \p{Nv=1/16}) (3)
T \p{Numeric_Value: 1/10} (Short: \p{Nv=1/10}) (1)
T \p{Numeric_Value: 1/9}  (Short: \p{Nv=1/9}) (1)
T \p{Numeric_Value: 1/8}  (Short: \p{Nv=1/8}) (5)
T \p{Numeric_Value: 1/7}  (Short: \p{Nv=1/7}) (1)
T \p{Numeric_Value: 1/6}  (Short: \p{Nv=1/6}) (2)
T \p{Numeric_Value: 3/16} (Short: \p{Nv=3/16}) (3)
T \p{Numeric_Value: 1/5}  (Short: \p{Nv=1/5}) (1)
T \p{Numeric_Value: 1/4}  (Short: \p{Nv=1/4}) (9)
T \p{Numeric_Value: 1/3}  (Short: \p{Nv=1/3}) (4)
T \p{Numeric_Value: 3/8}  (Short: \p{Nv=3/8}) (1)
T \p{Numeric_Value: 2/5}  (Short: \p{Nv=2/5}) (1)
T \p{Numeric_Value: 1/2}  (Short: \p{Nv=1/2}) (10)
T \p{Numeric_Value: 3/5}  (Short: \p{Nv=3/5}) (1)
T \p{Numeric_Value: 5/8}  (Short: \p{Nv=5/8}) (1)
T \p{Numeric_Value: 2/3}  (Short: \p{Nv=2/3}) (5)
T \p{Numeric_Value: 3/4}  (Short: \p{Nv=3/4}) (6)
T \p{Numeric_Value: 4/5}  (Short: \p{Nv=4/5}) (1)
T \p{Numeric_Value: 5/6}  (Short: \p{Nv=5/6}) (2)
T \p{Numeric_Value: 7/8}  (Short: \p{Nv=7/8}) (1)
T \p{Numeric_Value: 1}    (Short: \p{Nv=1}) (97)
```

```
T \p{Numeric_Value: 3/2}  (Short: \p{Nv=3/2}) (1)
T \p{Numeric_Value: 2}    (Short: \p{Nv=2}) (101)
T \p{Numeric_Value: 5/2}  (Short: \p{Nv=5/2}) (1)
T \p{Numeric_Value: 3}    (Short: \p{Nv=3}) (103)
T \p{Numeric_Value: 7/2}  (Short: \p{Nv=7/2}) (1)
T \p{Numeric_Value: 4}    (Short: \p{Nv=4}) (93)
T \p{Numeric_Value: 9/2}  (Short: \p{Nv=9/2}) (1)
T \p{Numeric_Value: 5}    (Short: \p{Nv=5}) (90)
T \p{Numeric_Value: 11/2} (Short: \p{Nv=11/2}) (1)
T \p{Numeric_Value: 6}    (Short: \p{Nv=6}) (82)
T \p{Numeric_Value: 13/2} (Short: \p{Nv=13/2}) (1)
T \p{Numeric_Value: 7}    (Short: \p{Nv=7}) (81)
T \p{Numeric_Value: 15/2} (Short: \p{Nv=15/2}) (1)
T \p{Numeric_Value: 8}    (Short: \p{Nv=8}) (77)
T \p{Numeric_Value: 17/2} (Short: \p{Nv=17/2}) (1)
T \p{Numeric_Value: 9}    (Short: \p{Nv=9}) (81)
T \p{Numeric_Value: 10}   (Short: \p{Nv=10}) (40)
T \p{Numeric_Value: 11}   (Short: \p{Nv=11}) (6)
T \p{Numeric_Value: 12}   (Short: \p{Nv=12}) (6)
T \p{Numeric_Value: 13}   (Short: \p{Nv=13}) (4)
T \p{Numeric_Value: 14}   (Short: \p{Nv=14}) (4)
T \p{Numeric_Value: 15}   (Short: \p{Nv=15}) (4)
T \p{Numeric_Value: 16}   (Short: \p{Nv=16}) (5)
T \p{Numeric_Value: 17}   (Short: \p{Nv=17}) (5)
T \p{Numeric_Value: 18}   (Short: \p{Nv=18}) (5)
T \p{Numeric_Value: 19}   (Short: \p{Nv=19}) (5)
T \p{Numeric_Value: 20}   (Short: \p{Nv=20}) (19)
T \p{Numeric_Value: 21}   (Short: \p{Nv=21}) (1)
T \p{Numeric_Value: 22}   (Short: \p{Nv=22}) (1)
T \p{Numeric_Value: 23}   (Short: \p{Nv=23}) (1)
T \p{Numeric_Value: 24}   (Short: \p{Nv=24}) (1)
T \p{Numeric_Value: 25}   (Short: \p{Nv=25}) (1)
T \p{Numeric_Value: 26}   (Short: \p{Nv=26}) (1)
T \p{Numeric_Value: 27}   (Short: \p{Nv=27}) (1)
T \p{Numeric_Value: 28}   (Short: \p{Nv=28}) (1)
T \p{Numeric_Value: 29}   (Short: \p{Nv=29}) (1)
T \p{Numeric_Value: 30}   (Short: \p{Nv=30}) (11)
T \p{Numeric_Value: 31}   (Short: \p{Nv=31}) (1)
T \p{Numeric_Value: 32}   (Short: \p{Nv=32}) (1)
T \p{Numeric_Value: 33}   (Short: \p{Nv=33}) (1)
T \p{Numeric_Value: 34}   (Short: \p{Nv=34}) (1)
T \p{Numeric_Value: 35}   (Short: \p{Nv=35}) (1)
T \p{Numeric_Value: 36}   (Short: \p{Nv=36}) (1)
T \p{Numeric_Value: 37}   (Short: \p{Nv=37}) (1)
T \p{Numeric_Value: 38}   (Short: \p{Nv=38}) (1)
T \p{Numeric_Value: 39}   (Short: \p{Nv=39}) (1)
T \p{Numeric_Value: 40}   (Short: \p{Nv=40}) (10)
T \p{Numeric_Value: 41}   (Short: \p{Nv=41}) (1)
T \p{Numeric_Value: 42}   (Short: \p{Nv=42}) (1)
T \p{Numeric_Value: 43}   (Short: \p{Nv=43}) (1)
T \p{Numeric_Value: 44}   (Short: \p{Nv=44}) (1)
T \p{Numeric_Value: 45}   (Short: \p{Nv=45}) (1)
T \p{Numeric_Value: 46}   (Short: \p{Nv=46}) (1)
T \p{Numeric_Value: 47}   (Short: \p{Nv=47}) (1)
T \p{Numeric_Value: 48}   (Short: \p{Nv=48}) (1)
T \p{Numeric_Value: 49}   (Short: \p{Nv=49}) (1)
```

```
T \p{Numeric_Value: 50}   (Short: \p{Nv=50}) (20)
T \p{Numeric_Value: 60}   (Short: \p{Nv=60}) (6)
T \p{Numeric_Value: 70}   (Short: \p{Nv=70}) (6)
T \p{Numeric_Value: 80}   (Short: \p{Nv=80}) (6)
T \p{Numeric_Value: 90}   (Short: \p{Nv=90}) (6)
T \p{Numeric_Value: 100}  (Short: \p{Nv=100}) (20)
T \p{Numeric_Value: 200}  (Short: \p{Nv=200}) (2)
T \p{Numeric_Value: 300}  (Short: \p{Nv=300}) (3)
T \p{Numeric_Value: 400}  (Short: \p{Nv=400}) (2)
T \p{Numeric_Value: 500}  (Short: \p{Nv=500}) (12)
T \p{Numeric_Value: 600}  (Short: \p{Nv=600}) (2)
T \p{Numeric_Value: 700}  (Short: \p{Nv=700}) (2)
T \p{Numeric_Value: 800}  (Short: \p{Nv=800}) (2)
T \p{Numeric_Value: 900}  (Short: \p{Nv=900}) (3)
T \p{Numeric_Value: 1000} (Short: \p{Nv=1000}) (17)
T \p{Numeric_Value: 2000} (Short: \p{Nv=2000}) (1)
T \p{Numeric_Value: 3000} (Short: \p{Nv=3000}) (1)
T \p{Numeric_Value: 4000} (Short: \p{Nv=4000}) (1)
T \p{Numeric_Value: 5000} (Short: \p{Nv=5000}) (5)
T \p{Numeric_Value: 6000} (Short: \p{Nv=6000}) (1)
T \p{Numeric_Value: 7000} (Short: \p{Nv=7000}) (1)
T \p{Numeric_Value: 8000} (Short: \p{Nv=8000}) (1)
T \p{Numeric_Value: 9000} (Short: \p{Nv=9000}) (1)
T \p{Numeric_Value: 10000} (= 1.0e+04) (Short: \p{Nv=10000}) (7)
T \p{Numeric_Value: 20000} (= 2.0e+04) (Short: \p{Nv=20000}) (1)
T \p{Numeric_Value: 30000} (= 3.0e+04) (Short: \p{Nv=30000}) (1)
T \p{Numeric_Value: 40000} (= 4.0e+04) (Short: \p{Nv=40000}) (1)
T \p{Numeric_Value: 50000} (= 5.0e+04) (Short: \p{Nv=50000}) (4)
T \p{Numeric_Value: 60000} (= 6.0e+04) (Short: \p{Nv=60000}) (1)
T \p{Numeric_Value: 70000} (= 7.0e+04) (Short: \p{Nv=70000}) (1)
T \p{Numeric_Value: 80000} (= 8.0e+04) (Short: \p{Nv=80000}) (1)
T \p{Numeric_Value: 90000} (= 9.0e+04) (Short: \p{Nv=90000}) (1)
T \p{Numeric_Value: 100000} (= 1.0e+05) (Short: \p{Nv=100000}) (1)
T \p{Numeric_Value: 216000} (= 2.2e+05) (Short: \p{Nv=216000}) (1)
T \p{Numeric_Value: 432000} (= 4.3e+05) (Short: \p{Nv=432000}) (1)
T \p{Numeric_Value: 100000000} (= 1.0e+08) (Short: \p{Nv=100000000})
                            (2)
T \p{Numeric_Value: 1000000000000} (= 1.0e+12) (Short: \p{Nv=
                            1000000000000}) (1)
  \p{Numeric_Value: NaN}  (Short: \p{Nv=NaN}) (1_112_883 plus all
                            above-Unicode code points)
  \p{Nv: *}               \p{Numeric_Value: *}
X \p{OCR}                 \p{Optical_Character_Recognition} (=
                            \p{Block=Optical_Character_Recognition})
                            (32)
  \p{Ogam}                \p{Ogham} (= \p{Script=Ogham}) (NOT
                            \p{Block=Ogham}) (29)
  \p{Ogham}               \p{Script=Ogham} (Short: \p{Ogam}; NOT
                            \p{Block=Ogham}) (29)
  \p{Ol_Chiki}            \p{Script=Ol_Chiki} (Short: \p{Olck}) (48)
  \p{Olck}                \p{Ol_Chiki} (= \p{Script=Ol_Chiki}) (48)
  \p{Old_Italic}          \p{Script=Old_Italic} (Short: \p{Ital};
                            NOT \p{Block=Old_Italic}) (35)
  \p{Old_Persian}         \p{Script=Old_Persian} (Short: \p{Xpeo};
                            NOT \p{Block=Old_Persian}) (50)
  \p{Old_South_Arabian}   \p{Script=Old_South_Arabian} (Short:
```

```
                                \p{Sarb}) (32)
    \p{Old_Turkic}          \p{Script=Old_Turkic} (Short: \p{Orkh};
                                NOT \p{Block=Old_Turkic}) (73)
    \p{Open_Punctuation}    \p{General_Category=Open_Punctuation}
                                (Short: \p{Ps}) (74)
  X \p{Optical_Character_Recognition} \p{Block=
                                Optical_Character_Recognition} (Short:
                                \p{InOCR}) (32)
    \p{Oriya}               \p{Script=Oriya} (Short: \p{Orya}; NOT
                                \p{Block=Oriya}) (90)
    \p{Orkh}                \p{Old_Turkic} (= \p{Script=Old_Turkic})
                                (NOT \p{Block=Old_Turkic}) (73)
    \p{Orya}                \p{Oriya} (= \p{Script=Oriya}) (NOT
                                \p{Block=Oriya}) (90)
    \p{Osma}                \p{Osmanya} (= \p{Script=Osmanya}) (NOT
                                \p{Block=Osmanya}) (40)
    \p{Osmanya}             \p{Script=Osmanya} (Short: \p{Osma}; NOT
                                \p{Block=Osmanya}) (40)
    \p{Other}               \p{General_Category=Other} (Short: \p{C})
                                (1_004_135 plus all above-Unicode code
                                points)
    \p{Other_Letter}        \p{General_Category=Other_Letter} (Short:
                                \p{Lo}) (97_553)
    \p{Other_Number}        \p{General_Category=Other_Number} (Short:
                                \p{No}) (464)
    \p{Other_Punctuation}   \p{General_Category=Other_Punctuation}
                                (Short: \p{Po}) (434)
    \p{Other_Symbol}        \p{General_Category=Other_Symbol} (Short:
                                \p{So}) (4404)
    \p{P} \pP               \p{Punct} (= \p{General_Category=
                                Punctuation}) (NOT
                                \p{General_Punctuation}) (636)
    \p{Paragraph_Separator} \p{General_Category=Paragraph_Separator}
                                (Short: \p{Zp}) (1)
    \p{Pat_Syn}             \p{Pattern_Syntax} (= \p{Pattern_Syntax=
                                Y}) (2760)
    \p{Pat_Syn: *}          \p{Pattern_Syntax: *}
    \p{Pat_WS}              \p{Pattern_White_Space} (=
                                \p{Pattern_White_Space=Y}) (11)
    \p{Pat_WS: *}           \p{Pattern_White_Space: *}
    \p{Pattern_Syntax}      \p{Pattern_Syntax=Y} (Short: \p{PatSyn})
                                (2760)
    \p{Pattern_Syntax: N*}  (Short: \p{PatSyn=N}, \P{PatSyn})
                                (1_111_352 plus all above-Unicode code
                                points)
    \p{Pattern_Syntax: Y*}  (Short: \p{PatSyn=Y}, \p{PatSyn}) (2760)
    \p{Pattern_White_Space} \p{Pattern_White_Space=Y} (Short:
                                \p{PatWS}) (11)
    \p{Pattern_White_Space: N*} (Short: \p{PatWS=N}, \P{PatWS})
                                (1_114_101 plus all above-Unicode code
                                points)
    \p{Pattern_White_Space: Y*} (Short: \p{PatWS=Y}, \p{PatWS}) (11)
    \p{Pc}                  \p{Connector_Punctuation} (=
                                \p{General_Category=
                                Connector_Punctuation}) (10)
    \p{Pd}                  \p{Dash_Punctuation} (=
```

```
                                \p{General_Category=Dash_Punctuation})
                                (23)
    \p{Pe}                      \p{Close_Punctuation} (=
                                \p{General_Category=Close_Punctuation})
                                (73)
    \p{PerlSpace}               \s, restricted to ASCII = [ \f\n\r\t] plus
                                vertical tab (6)
    \p{PerlWord}                \w, restricted to ASCII = [A-Za-z0-9_] (63)
    \p{Pf}                      \p{Final_Punctuation} (=
                                \p{General_Category=Final_Punctuation})
                                (10)
    \p{Phag}                    \p{Phags_Pa} (= \p{Script=Phags_Pa}) (NOT
                                \p{Block=Phags_Pa}) (56)
    \p{Phags_Pa}                \p{Script=Phags_Pa} (Short: \p{Phag}; NOT
                                \p{Block=Phags_Pa}) (56)
X   \p{Phaistos}                \p{Phaistos_Disc} (= \p{Block=
                                Phaistos_Disc}) (48)
X   \p{Phaistos_Disc}           \p{Block=Phaistos_Disc} (Short:
                                \p{InPhaistos}) (48)
    \p{Phli}                    \p{Inscriptional_Pahlavi} (= \p{Script=
                                Inscriptional_Pahlavi}) (NOT \p{Block=
                                Inscriptional_Pahlavi}) (27)
    \p{Phnx}                    \p{Phoenician} (= \p{Script=Phoenician})
                                (NOT \p{Block=Phoenician}) (29)
    \p{Phoenician}              \p{Script=Phoenician} (Short: \p{Phnx};
                                NOT \p{Block=Phoenician}) (29)
X   \p{Phonetic_Ext}            \p{Phonetic_Extensions} (= \p{Block=
                                Phonetic_Extensions}) (128)
X   \p{Phonetic_Ext_Sup}        \p{Phonetic_Extensions_Supplement} (=
                                \p{Block=
                                Phonetic_Extensions_Supplement}) (64)
X   \p{Phonetic_Extensions}     \p{Block=Phonetic_Extensions} (Short:
                                \p{InPhoneticExt}) (128)
X   \p{Phonetic_Extensions_Supplement} \p{Block=
                                Phonetic_Extensions_Supplement} (Short:
                                \p{InPhoneticExtSup}) (64)
    \p{Pi}                      \p{Initial_Punctuation} (=
                                \p{General_Category=
                                Initial_Punctuation}) (12)
X   \p{Playing_Cards}           \p{Block=Playing_Cards} (96)
    \p{Plrd}                    \p{Miao} (= \p{Script=Miao}) (NOT
                                \p{Block=Miao}) (133)
    \p{Po}                      \p{Other_Punctuation} (=
                                \p{General_Category=Other_Punctuation})
                                (434)
    \p{PosixAlnum}              [A-Za-z0-9] (62)
    \p{PosixAlpha}              [A-Za-z] (52)
    \p{PosixBlank}              \t and ' ' (2)
    \p{PosixCntrl}              ASCII control characters: NUL, SOH, STX,
                                ETX, EOT, ENQ, ACK, BEL, BS, HT, LF, VT,
                                FF, CR, SO, SI, DLE, DC1, DC2, DC3, DC4,
                                NAK, SYN, ETB, CAN, EOM, SUB, ESC, FS,
                                GS, RS, US, and DEL (33)
    \p{PosixDigit}              [0-9] (10)
    \p{PosixGraph}              [-!"#$%&'()*+,./:;<=>?@[\\]^_`{|}~0-9A-Za-
                                z] (94)
```

```
    \p{PosixLower}          [a-z] (/i= PosixAlpha) (26)
    \p{PosixPrint}          [- 0-9A-Za-z!"#$%&'()*+,./:;<=
                              >?@[\\]^_`{|}~] (95)
    \p{PosixPunct}          [-!"#$%&'()*+,./:;<=>?@[\\]^_`{|}~] (32)
    \p{PosixSpace}          \t, \n, \cK, \f, \r, and ' '.  (\cK is
                              vertical tab) (6)
    \p{PosixUpper}          [A-Z] (/i= PosixAlpha) (26)
    \p{PosixWord}           \p{PerlWord} (63)
    \p{PosixXDigit}         \p{ASCII_Hex_Digit=Y} [0-9A-Fa-f] (Short:
                              \p{AHex}) (22)
T \p{Present_In: 1.1}       \p{Age=V1_1} (Short: \p{In=1.1}) (Perl
                              extension) (33_979)
T \p{Present_In: 2.0}       Code point's usage introduced in version
                              2.0 or earlier (Short: \p{In=2.0}) (Perl
                              extension) (178_500)
T \p{Present_In: 2.1}       Code point's usage introduced in version
                              2.1 or earlier (Short: \p{In=2.1}) (Perl
                              extension) (178_502)
T \p{Present_In: 3.0}       Code point's usage introduced in version
                              3.0 or earlier (Short: \p{In=3.0}) (Perl
                              extension) (188_809)
T \p{Present_In: 3.1}       Code point's usage introduced in version
                              3.1 or earlier (Short: \p{In=3.1}) (Perl
                              extension) (233_787)
T \p{Present_In: 3.2}       Code point's usage introduced in version
                              3.2 or earlier (Short: \p{In=3.2}) (Perl
                              extension) (234_803)
T \p{Present_In: 4.0}       Code point's usage introduced in version
                              4.0 or earlier (Short: \p{In=4.0}) (Perl
                              extension) (236_029)
T \p{Present_In: 4.1}       Code point's usage introduced in version
                              4.1 or earlier (Short: \p{In=4.1}) (Perl
                              extension) (237_302)
T \p{Present_In: 5.0}       Code point's usage introduced in version
                              5.0 or earlier (Short: \p{In=5.0}) (Perl
                              extension) (238_671)
T \p{Present_In: 5.1}       Code point's usage introduced in version
                              5.1 or earlier (Short: \p{In=5.1}) (Perl
                              extension) (240_295)
T \p{Present_In: 5.2}       Code point's usage introduced in version
                              5.2 or earlier (Short: \p{In=5.2}) (Perl
                              extension) (246_943)
T \p{Present_In: 6.0}       Code point's usage introduced in version
                              6.0 or earlier (Short: \p{In=6.0}) (Perl
                              extension) (249_031)
T \p{Present_In: 6.1}       Code point's usage introduced in version
                              6.1 or earlier (Short: \p{In=6.1}) (Perl
                              extension) (249_763)
T \p{Present_In: 6.2}       Code point's usage introduced in version
                              6.2 or earlier (Short: \p{In=6.2}) (Perl
                              extension) (249_764)
T \p{Present_In: 6.3}       Code point's usage introduced in version
                              6.3 or earlier (Short: \p{In=6.3}) (Perl
                              extension) (249_769)
    \p{Present_In: Unassigned} \p{Age=Unassigned} (Short: \p{In=
                              Unassigned}) (Perl extension) (864_343
```

```
                                plus all above-Unicode code points)
    \p{Print}                 Characters that are graphical plus space
                                characters (but no controls) (247_588)
    \p{Private_Use}           \p{General_Category=Private_Use} (Short:
                                \p{Co}; NOT \p{Private_Use_Area})
                                (137_468)
  X \p{Private_Use_Area}      \p{Block=Private_Use_Area} (Short:
                                \p{InPUA}) (6400)
    \p{Prti}                  \p{Inscriptional_Parthian} (= \p{Script=
                                Inscriptional_Parthian}) (NOT \p{Block=
                                Inscriptional_Parthian}) (30)
    \p{Ps}                    \p{Open_Punctuation} (=
                                \p{General_Category=Open_Punctuation})
                                (74)
  X \p{PUA}                   \p{Private_Use_Area} (= \p{Block=
                                Private_Use_Area}) (6400)
    \p{Punct}                 \p{General_Category=Punctuation} (Short:
                                \p{P}; NOT \p{General_Punctuation}) (636)
    \p{Punctuation}           \p{Punct} (= \p{General_Category=
                                Punctuation}) (NOT
                                \p{General_Punctuation}) (636)
    \p{Qaac}                  \p{Coptic} (= \p{Script=Coptic}) (NOT
                                \p{Block=Coptic}) (137)
    \p{Qaai}                  \p{Inherited} (= \p{Script=Inherited})
                                (523)
    \p{QMark}                 \p{Quotation_Mark} (= \p{Quotation_Mark=
                                Y}) (29)
    \p{QMark: *}              \p{Quotation_Mark: *}
    \p{Quotation_Mark}        \p{Quotation_Mark=Y} (Short: \p{QMark})
                                (29)
    \p{Quotation_Mark: N*}    (Short: \p{QMark=N}, \P{QMark}) (1_114_083
                                plus all above-Unicode code points)
    \p{Quotation_Mark: Y*}    (Short: \p{QMark=Y}, \p{QMark}) (29)
    \p{Radical}               \p{Radical=Y} (329)
    \p{Radical: N*}           (Single: \P{Radical}) (1_113_783 plus all
                                above-Unicode code points)
    \p{Radical: Y*}           (Single: \p{Radical}) (329)
    \p{Rejang}                \p{Script=Rejang} (Short: \p{Rjng}; NOT
                                \p{Block=Rejang}) (37)
    \p{Rjng}                  \p{Rejang} (= \p{Script=Rejang}) (NOT
                                \p{Block=Rejang}) (37)
  X \p{Rumi}                  \p{Rumi_Numeral_Symbols} (= \p{Block=
                                Rumi_Numeral_Symbols}) (32)
  X \p{Rumi_Numeral_Symbols} \p{Block=Rumi_Numeral_Symbols} (Short:
                                \p{InRumi}) (32)
    \p{Runic}                 \p{Script=Runic} (Short: \p{Runr}; NOT
                                \p{Block=Runic}) (78)
    \p{Runr}                  \p{Runic} (= \p{Script=Runic}) (NOT
                                \p{Block=Runic}) (78)
    \p{S} \pS                 \p{Symbol} (= \p{General_Category=Symbol})
                                (5516)
    \p{Samaritan}             \p{Script=Samaritan} (Short: \p{Samr}; NOT
                                \p{Block=Samaritan}) (61)
    \p{Samr}                  \p{Samaritan} (= \p{Script=Samaritan})
                                (NOT \p{Block=Samaritan}) (61)
    \p{Sarb}                  \p{Old_South_Arabian} (= \p{Script=
```

```
                          Old_South_Arabian}) (32)
\p{Saur}                \p{Saurashtra} (= \p{Script=Saurashtra})
                          (NOT \p{Block=Saurashtra}) (81)
\p{Saurashtra}          \p{Script=Saurashtra} (Short: \p{Saur};
                          NOT \p{Block=Saurashtra}) (81)
\p{SB: *}               \p{Sentence_Break: *}
\p{Sc}                  \p{Currency_Symbol} (=
                          \p{General_Category=Currency_Symbol})
                          (49)
\p{Sc: *}               \p{Script: *}
\p{Script: Arab}        \p{Script=Arabic} (1236)
\p{Script: Arabic}      (Short: \p{Sc=Arab}, \p{Arab}) (1236)
\p{Script: Armenian}    (Short: \p{Sc=Armn}, \p{Armn}) (91)
\p{Script: Armi}        \p{Script=Imperial_Aramaic} (31)
\p{Script: Armn}        \p{Script=Armenian} (91)
\p{Script: Avestan}     (Short: \p{Sc=Avst}, \p{Avst}) (61)
\p{Script: Avst}        \p{Script=Avestan} (61)
\p{Script: Bali}        \p{Script=Balinese} (121)
\p{Script: Balinese}    (Short: \p{Sc=Bali}, \p{Bali}) (121)
\p{Script: Bamu}        \p{Script=Bamum} (657)
\p{Script: Bamum}       (Short: \p{Sc=Bamu}, \p{Bamu}) (657)
\p{Script: Batak}       (Short: \p{Sc=Batk}, \p{Batk}) (56)
\p{Script: Batk}        \p{Script=Batak} (56)
\p{Script: Beng}        \p{Script=Bengali} (92)
\p{Script: Bengali}     (Short: \p{Sc=Beng}, \p{Beng}) (92)
\p{Script: Bopo}        \p{Script=Bopomofo} (70)
\p{Script: Bopomofo}    (Short: \p{Sc=Bopo}, \p{Bopo}) (70)
\p{Script: Brah}        \p{Script=Brahmi} (108)
\p{Script: Brahmi}      (Short: \p{Sc=Brah}, \p{Brah}) (108)
\p{Script: Brai}        \p{Script=Braille} (256)
\p{Script: Braille}     (Short: \p{Sc=Brai}, \p{Brai}) (256)
\p{Script: Bugi}        \p{Script=Buginese} (30)
\p{Script: Buginese}    (Short: \p{Sc=Bugi}, \p{Bugi}) (30)
\p{Script: Buhd}        \p{Script=Buhid} (20)
\p{Script: Buhid}       (Short: \p{Sc=Buhd}, \p{Buhd}) (20)
\p{Script: Cakm}        \p{Script=Chakma} (67)
\p{Script: Canadian_Aboriginal} (Short: \p{Sc=Cans}, \p{Cans})
                          (710)
\p{Script: Cans}        \p{Script=Canadian_Aboriginal} (710)
\p{Script: Cari}        \p{Script=Carian} (49)
\p{Script: Carian}      (Short: \p{Sc=Cari}, \p{Cari}) (49)
\p{Script: Chakma}      (Short: \p{Sc=Cakm}, \p{Cakm}) (67)
\p{Script: Cham}        (Short: \p{Sc=Cham}, \p{Cham}) (83)
\p{Script: Cher}        \p{Script=Cherokee} (85)
\p{Script: Cherokee}    (Short: \p{Sc=Cher}, \p{Cher}) (85)
\p{Script: Common}      (Short: \p{Sc=Zyyy}, \p{Zyyy}) (6418)
\p{Script: Copt}        \p{Script=Coptic} (137)
\p{Script: Coptic}      (Short: \p{Sc=Copt}, \p{Copt}) (137)
\p{Script: Cprt}        \p{Script=Cypriot} (55)
\p{Script: Cuneiform}   (Short: \p{Sc=Xsux}, \p{Xsux}) (982)
\p{Script: Cypriot}     (Short: \p{Sc=Cprt}, \p{Cprt}) (55)
\p{Script: Cyrillic}    (Short: \p{Sc=Cyrl}, \p{Cyrl}) (417)
\p{Script: Cyrl}        \p{Script=Cyrillic} (417)
\p{Script: Deseret}     (Short: \p{Sc=Dsrt}, \p{Dsrt}) (80)
\p{Script: Deva}        \p{Script=Devanagari} (151)
\p{Script: Devanagari}  (Short: \p{Sc=Deva}, \p{Deva}) (151)
```

```
\p{Script: Dsrt}        \p{Script=Deseret} (80)
\p{Script: Egyp}        \p{Script=Egyptian_Hieroglyphs} (1071)
\p{Script: Egyptian_Hieroglyphs} (Short: \p{Sc=Egyp}, \p{Egyp})
                          (1071)
\p{Script: Ethi}        \p{Script=Ethiopic} (495)
\p{Script: Ethiopic}    (Short: \p{Sc=Ethi}, \p{Ethi}) (495)
\p{Script: Geor}        \p{Script=Georgian} (127)
\p{Script: Georgian}    (Short: \p{Sc=Geor}, \p{Geor}) (127)
\p{Script: Glag}        \p{Script=Glagolitic} (94)
\p{Script: Glagolitic}  (Short: \p{Sc=Glag}, \p{Glag}) (94)
\p{Script: Goth}        \p{Script=Gothic} (27)
\p{Script: Gothic}      (Short: \p{Sc=Goth}, \p{Goth}) (27)
\p{Script: Greek}       (Short: \p{Sc=Grek}, \p{Grek}) (511)
\p{Script: Grek}        \p{Script=Greek} (511)
\p{Script: Gujarati}    (Short: \p{Sc=Gujr}, \p{Gujr}) (84)
\p{Script: Gujr}        \p{Script=Gujarati} (84)
\p{Script: Gurmukhi}    (Short: \p{Sc=Guru}, \p{Guru}) (79)
\p{Script: Guru}        \p{Script=Gurmukhi} (79)
\p{Script: Han}         (Short: \p{Sc=Han}, \p{Han}) (75_963)
\p{Script: Hang}        \p{Script=Hangul} (11_739)
\p{Script: Hangul}      (Short: \p{Sc=Hang}, \p{Hang}) (11_739)
\p{Script: Hani}        \p{Script=Han} (75_963)
\p{Script: Hano}        \p{Script=Hanunoo} (21)
\p{Script: Hanunoo}     (Short: \p{Sc=Hano}, \p{Hano}) (21)
\p{Script: Hebr}        \p{Script=Hebrew} (133)
\p{Script: Hebrew}      (Short: \p{Sc=Hebr}, \p{Hebr}) (133)
\p{Script: Hira}        \p{Script=Hiragana} (91)
\p{Script: Hiragana}    (Short: \p{Sc=Hira}, \p{Hira}) (91)
\p{Script: Imperial_Aramaic} (Short: \p{Sc=Armi}, \p{Armi}) (31)
\p{Script: Inherited}   (Short: \p{Sc=Zinh}, \p{Zinh}) (523)
\p{Script: Inscriptional_Pahlavi} (Short: \p{Sc=Phli}, \p{Phli})
                          (27)
\p{Script: Inscriptional_Parthian} (Short: \p{Sc=Prti}, \p{Prti})
                          (30)
\p{Script: Ital}        \p{Script=Old_Italic} (35)
\p{Script: Java}        \p{Script=Javanese} (90)
\p{Script: Javanese}    (Short: \p{Sc=Java}, \p{Java}) (90)
\p{Script: Kaithi}      (Short: \p{Sc=Kthi}, \p{Kthi}) (66)
\p{Script: Kali}        \p{Script=Kayah_Li} (48)
\p{Script: Kana}        \p{Script=Katakana} (300)
\p{Script: Kannada}     (Short: \p{Sc=Knda}, \p{Knda}) (86)
\p{Script: Katakana}    (Short: \p{Sc=Kana}, \p{Kana}) (300)
\p{Script: Kayah_Li}    (Short: \p{Sc=Kali}, \p{Kali}) (48)
\p{Script: Khar}        \p{Script=Kharoshthi} (65)
\p{Script: Kharoshthi}  (Short: \p{Sc=Khar}, \p{Khar}) (65)
\p{Script: Khmer}       (Short: \p{Sc=Khmr}, \p{Khmr}) (146)
\p{Script: Khmr}        \p{Script=Khmer} (146)
\p{Script: Knda}        \p{Script=Kannada} (86)
\p{Script: Kthi}        \p{Script=Kaithi} (66)
\p{Script: Lana}        \p{Script=Tai_Tham} (127)
\p{Script: Lao}         (Short: \p{Sc=Lao}, \p{Lao}) (67)
\p{Script: Laoo}        \p{Script=Lao} (67)
\p{Script: Latin}       (Short: \p{Sc=Latn}, \p{Latn}) (1272)
\p{Script: Latn}        \p{Script=Latin} (1272)
\p{Script: Lepc}        \p{Script=Lepcha} (74)
\p{Script: Lepcha}      (Short: \p{Sc=Lepc}, \p{Lepc}) (74)
```

```
\p{Script: Limb}        \p{Script=Limbu} (66)
\p{Script: Limbu}       (Short: \p{Sc=Limb}, \p{Limb}) (66)
\p{Script: Linb}        \p{Script=Linear_B} (211)
\p{Script: Linear_B}    (Short: \p{Sc=Linb}, \p{Linb}) (211)
\p{Script: Lisu}        (Short: \p{Sc=Lisu}, \p{Lisu}) (48)
\p{Script: Lyci}        \p{Script=Lycian} (29)
\p{Script: Lycian}      (Short: \p{Sc=Lyci}, \p{Lyci}) (29)
\p{Script: Lydi}        \p{Script=Lydian} (27)
\p{Script: Lydian}      (Short: \p{Sc=Lydi}, \p{Lydi}) (27)
\p{Script: Malayalam}   (Short: \p{Sc=Mlym}, \p{Mlym}) (98)
\p{Script: Mand}        \p{Script=Mandaic} (29)
\p{Script: Mandaic}     (Short: \p{Sc=Mand}, \p{Mand}) (29)
\p{Script: Meetei_Mayek} (Short: \p{Sc=Mtei}, \p{Mtei}) (79)
\p{Script: Merc}        \p{Script=Meroitic_Cursive} (26)
\p{Script: Mero}        \p{Script=Meroitic_Hieroglyphs} (32)
\p{Script: Meroitic_Cursive} (Short: \p{Sc=Merc}, \p{Merc}) (26)
\p{Script: Meroitic_Hieroglyphs} (Short: \p{Sc=Mero}, \p{Mero})
                          (32)
\p{Script: Miao}        (Short: \p{Sc=Miao}, \p{Miao}) (133)
\p{Script: Mlym}        \p{Script=Malayalam} (98)
\p{Script: Mong}        \p{Script=Mongolian} (153)
\p{Script: Mongolian}   (Short: \p{Sc=Mong}, \p{Mong}) (153)
\p{Script: Mtei}        \p{Script=Meetei_Mayek} (79)
\p{Script: Myanmar}     (Short: \p{Sc=Mymr}, \p{Mymr}) (188)
\p{Script: Mymr}        \p{Script=Myanmar} (188)
\p{Script: New_Tai_Lue} (Short: \p{Sc=Talu}, \p{Talu}) (83)
\p{Script: Nko}         (Short: \p{Sc=Nko}, \p{Nko}) (59)
\p{Script: Nkoo}        \p{Script=Nko} (59)
\p{Script: Ogam}        \p{Script=Ogham} (29)
\p{Script: Ogham}       (Short: \p{Sc=Ogam}, \p{Ogam}) (29)
\p{Script: Ol_Chiki}    (Short: \p{Sc=Olck}, \p{Olck}) (48)
\p{Script: Olck}        \p{Script=Ol_Chiki} (48)
\p{Script: Old_Italic}  (Short: \p{Sc=Ital}, \p{Ital}) (35)
\p{Script: Old_Persian} (Short: \p{Sc=Xpeo}, \p{Xpeo}) (50)
\p{Script: Old_South_Arabian} (Short: \p{Sc=Sarb}, \p{Sarb}) (32)
\p{Script: Old_Turkic}  (Short: \p{Sc=Orkh}, \p{Orkh}) (73)
\p{Script: Oriya}       (Short: \p{Sc=Orya}, \p{Orya}) (90)
\p{Script: Orkh}        \p{Script=Old_Turkic} (73)
\p{Script: Orya}        \p{Script=Oriya} (90)
\p{Script: Osma}        \p{Script=Osmanya} (40)
\p{Script: Osmanya}     (Short: \p{Sc=Osma}, \p{Osma}) (40)
\p{Script: Phag}        \p{Script=Phags_Pa} (56)
\p{Script: Phags_Pa}    (Short: \p{Sc=Phag}, \p{Phag}) (56)
\p{Script: Phli}        \p{Script=Inscriptional_Pahlavi} (27)
\p{Script: Phnx}        \p{Script=Phoenician} (29)
\p{Script: Phoenician}  (Short: \p{Sc=Phnx}, \p{Phnx}) (29)
\p{Script: Plrd}        \p{Script=Miao} (133)
\p{Script: Prti}        \p{Script=Inscriptional_Parthian} (30)
\p{Script: Qaac}        \p{Script=Coptic} (137)
\p{Script: Qaai}        \p{Script=Inherited} (523)
\p{Script: Rejang}      (Short: \p{Sc=Rjng}, \p{Rjng}) (37)
\p{Script: Rjng}        \p{Script=Rejang} (37)
\p{Script: Runic}       (Short: \p{Sc=Runr}, \p{Runr}) (78)
\p{Script: Runr}        \p{Script=Runic} (78)
\p{Script: Samaritan}   (Short: \p{Sc=Samr}, \p{Samr}) (61)
\p{Script: Samr}        \p{Script=Samaritan} (61)
```

```
\p{Script: Sarb}        \p{Script=Old_South_Arabian} (32)
\p{Script: Saur}        \p{Script=Saurashtra} (81)
\p{Script: Saurashtra}  (Short: \p{Sc=Saur}, \p{Saur}) (81)
\p{Script: Sharada}     (Short: \p{Sc=Shrd}, \p{Shrd}) (83)
\p{Script: Shavian}     (Short: \p{Sc=Shaw}, \p{Shaw}) (48)
\p{Script: Shaw}        \p{Script=Shavian} (48)
\p{Script: Shrd}        \p{Script=Sharada} (83)
\p{Script: Sinh}        \p{Script=Sinhala} (80)
\p{Script: Sinhala}     (Short: \p{Sc=Sinh}, \p{Sinh}) (80)
\p{Script: Sora}        \p{Script=Sora_Sompeng} (35)
\p{Script: Sora_Sompeng} (Short: \p{Sc=Sora}, \p{Sora}) (35)
\p{Script: Sund}        \p{Script=Sundanese} (72)
\p{Script: Sundanese}   (Short: \p{Sc=Sund}, \p{Sund}) (72)
\p{Script: Sylo}        \p{Script=Syloti_Nagri} (44)
\p{Script: Syloti_Nagri} (Short: \p{Sc=Sylo}, \p{Sylo}) (44)
\p{Script: Syrc}        \p{Script=Syriac} (77)
\p{Script: Syriac}      (Short: \p{Sc=Syrc}, \p{Syrc}) (77)
\p{Script: Tagalog}     (Short: \p{Sc=Tglg}, \p{Tglg}) (20)
\p{Script: Tagb}        \p{Script=Tagbanwa} (18)
\p{Script: Tagbanwa}    (Short: \p{Sc=Tagb}, \p{Tagb}) (18)
\p{Script: Tai_Le}      (Short: \p{Sc=Tale}, \p{Tale}) (35)
\p{Script: Tai_Tham}    (Short: \p{Sc=Lana}, \p{Lana}) (127)
\p{Script: Tai_Viet}    (Short: \p{Sc=Tavt}, \p{Tavt}) (72)
\p{Script: Takr}        \p{Script=Takri} (66)
\p{Script: Takri}       (Short: \p{Sc=Takr}, \p{Takr}) (66)
\p{Script: Tale}        \p{Script=Tai_Le} (35)
\p{Script: Talu}        \p{Script=New_Tai_Lue} (83)
\p{Script: Tamil}       (Short: \p{Sc=Taml}, \p{Taml}) (72)
\p{Script: Taml}        \p{Script=Tamil} (72)
\p{Script: Tavt}        \p{Script=Tai_Viet} (72)
\p{Script: Telu}        \p{Script=Telugu} (93)
\p{Script: Telugu}      (Short: \p{Sc=Telu}, \p{Telu}) (93)
\p{Script: Tfng}        \p{Script=Tifinagh} (59)
\p{Script: Tglg}        \p{Script=Tagalog} (20)
\p{Script: Thaa}        \p{Script=Thaana} (50)
\p{Script: Thaana}      (Short: \p{Sc=Thaa}, \p{Thaa}) (50)
\p{Script: Thai}        (Short: \p{Sc=Thai}, \p{Thai}) (86)
\p{Script: Tibetan}     (Short: \p{Sc=Tibt}, \p{Tibt}) (207)
\p{Script: Tibt}        \p{Script=Tibetan} (207)
\p{Script: Tifinagh}    (Short: \p{Sc=Tfng}, \p{Tfng}) (59)
\p{Script: Ugar}        \p{Script=Ugaritic} (31)
\p{Script: Ugaritic}    (Short: \p{Sc=Ugar}, \p{Ugar}) (31)
\p{Script: Unknown}     (Short: \p{Sc=Zzzz}, \p{Zzzz}) (1_003_925
                          plus all above-Unicode code points)
\p{Script: Vai}         (Short: \p{Sc=Vai}, \p{Vai}) (300)
\p{Script: Vaii}        \p{Script=Vai} (300)
\p{Script: Xpeo}        \p{Script=Old_Persian} (50)
\p{Script: Xsux}        \p{Script=Cuneiform} (982)
\p{Script: Yi}          (Short: \p{Sc=Yi}, \p{Yi}) (1220)
\p{Script: Yiii}        \p{Script=Yi} (1220)
\p{Script: Zinh}        \p{Script=Inherited} (523)
\p{Script: Zyyy}        \p{Script=Common} (6418)
\p{Script: Zzzz}        \p{Script=Unknown} (1_003_925 plus all
                          above-Unicode code points)
\p{Script_Extensions: Arab} \p{Script_Extensions=Arabic} (1263)
\p{Script_Extensions: Arabic} (Short: \p{Scx=Arab}) (1263)
```

```
\p{Script_Extensions: Armenian} (Short: \p{Scx=Armn}) (92)
\p{Script_Extensions: Armi} \p{Script_Extensions=Imperial_Aramaic}
                        (31)
\p{Script_Extensions: Armn} \p{Script_Extensions=Armenian} (92)
\p{Script_Extensions: Avestan} (Short: \p{Scx=Avst}) (61)
\p{Script_Extensions: Avst} \p{Script_Extensions=Avestan} (61)
\p{Script_Extensions: Bali} \p{Script_Extensions=Balinese} (121)
\p{Script_Extensions: Balinese} (Short: \p{Scx=Bali}) (121)
\p{Script_Extensions: Bamu} \p{Script_Extensions=Bamum} (657)
\p{Script_Extensions: Bamum} (Short: \p{Scx=Bamu}) (657)
\p{Script_Extensions: Batak} (Short: \p{Scx=Batk}) (56)
\p{Script_Extensions: Batk} \p{Script_Extensions=Batak} (56)
\p{Script_Extensions: Beng} \p{Script_Extensions=Bengali} (94)
\p{Script_Extensions: Bengali} (Short: \p{Scx=Beng}) (94)
\p{Script_Extensions: Bopo} \p{Script_Extensions=Bopomofo} (306)
\p{Script_Extensions: Bopomofo} (Short: \p{Scx=Bopo}) (306)
\p{Script_Extensions: Brah} \p{Script_Extensions=Brahmi} (108)
\p{Script_Extensions: Brahmi} (Short: \p{Scx=Brah}) (108)
\p{Script_Extensions: Brai} \p{Script_Extensions=Braille} (256)
\p{Script_Extensions: Braille} (Short: \p{Scx=Brai}) (256)
\p{Script_Extensions: Bugi} \p{Script_Extensions=Buginese} (31)
\p{Script_Extensions: Buginese} (Short: \p{Scx=Bugi}) (31)
\p{Script_Extensions: Buhd} \p{Script_Extensions=Buhid} (22)
\p{Script_Extensions: Buhid} (Short: \p{Scx=Buhd}) (22)
\p{Script_Extensions: Cakm} \p{Script_Extensions=Chakma} (87)
\p{Script_Extensions: Canadian_Aboriginal} (Short: \p{Scx=Cans})
                        (710)
\p{Script_Extensions: Cans} \p{Script_Extensions=
                        Canadian_Aboriginal} (710)
\p{Script_Extensions: Cari} \p{Script_Extensions=Carian} (49)
\p{Script_Extensions: Carian} (Short: \p{Scx=Cari}) (49)
\p{Script_Extensions: Chakma} (Short: \p{Scx=Cakm}) (87)
\p{Script_Extensions: Cham} (Short: \p{Scx=Cham}) (83)
\p{Script_Extensions: Cher} \p{Script_Extensions=Cherokee} (85)
\p{Script_Extensions: Cherokee} (Short: \p{Scx=Cher}) (85)
\p{Script_Extensions: Common} (Short: \p{Scx=Zyyy}) (6061)
\p{Script_Extensions: Copt} \p{Script_Extensions=Coptic} (137)
\p{Script_Extensions: Coptic} (Short: \p{Scx=Copt}) (137)
\p{Script_Extensions: Cprt} \p{Script_Extensions=Cypriot} (112)
\p{Script_Extensions: Cuneiform} (Short: \p{Scx=Xsux}) (982)
\p{Script_Extensions: Cypriot} (Short: \p{Scx=Cprt}) (112)
\p{Script_Extensions: Cyrillic} (Short: \p{Scx=Cyrl}) (419)
\p{Script_Extensions: Cyrl} \p{Script_Extensions=Cyrillic} (419)
\p{Script_Extensions: Deseret} (Short: \p{Scx=Dsrt}) (80)
\p{Script_Extensions: Deva} \p{Script_Extensions=Devanagari} (193)
\p{Script_Extensions: Devanagari} (Short: \p{Scx=Deva}) (193)
\p{Script_Extensions: Dsrt} \p{Script_Extensions=Deseret} (80)
\p{Script_Extensions: Egyp} \p{Script_Extensions=
                        Egyptian_Hieroglyphs} (1071)
\p{Script_Extensions: Egyptian_Hieroglyphs} (Short: \p{Scx=Egyp})
                        (1071)
\p{Script_Extensions: Ethi} \p{Script_Extensions=Ethiopic} (495)
\p{Script_Extensions: Ethiopic} (Short: \p{Scx=Ethi}) (495)
\p{Script_Extensions: Geor} \p{Script_Extensions=Georgian} (128)
\p{Script_Extensions: Georgian} (Short: \p{Scx=Geor}) (128)
\p{Script_Extensions: Glag} \p{Script_Extensions=Glagolitic} (94)
```

```
\p{Script_Extensions: Glagolitic} (Short: \p{Scx=Glag}) (94)
\p{Script_Extensions: Goth} \p{Script_Extensions=Gothic} (27)
\p{Script_Extensions: Gothic} (Short: \p{Scx=Goth}) (27)
\p{Script_Extensions: Greek} (Short: \p{Scx=Grek}) (515)
\p{Script_Extensions: Grek} \p{Script_Extensions=Greek} (515)
\p{Script_Extensions: Gujarati} (Short: \p{Scx=Gujr}) (94)
\p{Script_Extensions: Gujr} \p{Script_Extensions=Gujarati} (94)
\p{Script_Extensions: Gurmukhi} (Short: \p{Scx=Guru}) (91)
\p{Script_Extensions: Guru} \p{Script_Extensions=Gurmukhi} (91)
\p{Script_Extensions: Han} (Short: \p{Scx=Han}) (76_218)
\p{Script_Extensions: Hang} \p{Script_Extensions=Hangul} (11_971)
\p{Script_Extensions: Hangul} (Short: \p{Scx=Hang}) (11_971)
\p{Script_Extensions: Hani} \p{Script_Extensions=Han} (76_218)
\p{Script_Extensions: Hano} \p{Script_Extensions=Hanunoo} (23)
\p{Script_Extensions: Hanunoo} (Short: \p{Scx=Hano}) (23)
\p{Script_Extensions: Hebr} \p{Script_Extensions=Hebrew} (133)
\p{Script_Extensions: Hebrew} (Short: \p{Scx=Hebr}) (133)
\p{Script_Extensions: Hira} \p{Script_Extensions=Hiragana} (356)
\p{Script_Extensions: Hiragana} (Short: \p{Scx=Hira}) (356)
\p{Script_Extensions: Imperial_Aramaic} (Short: \p{Scx=Armi}) (31)
\p{Script_Extensions: Inherited} (Short: \p{Scx=Zinh}) (459)
\p{Script_Extensions: Inscriptional_Pahlavi} (Short: \p{Scx=Phli})
                      (27)
\p{Script_Extensions: Inscriptional_Parthian} (Short: \p{Scx=
                      Prti}) (30)
\p{Script_Extensions: Ital} \p{Script_Extensions=Old_Italic} (35)
\p{Script_Extensions: Java} \p{Script_Extensions=Javanese} (91)
\p{Script_Extensions: Javanese} (Short: \p{Scx=Java}) (91)
\p{Script_Extensions: Kaithi} (Short: \p{Scx=Kthi}) (86)
\p{Script_Extensions: Kali} \p{Script_Extensions=Kayah_Li} (48)
\p{Script_Extensions: Kana} \p{Script_Extensions=Katakana} (565)
\p{Script_Extensions: Kannada} (Short: \p{Scx=Knda}) (86)
\p{Script_Extensions: Katakana} (Short: \p{Scx=Kana}) (565)
\p{Script_Extensions: Kayah_Li} (Short: \p{Scx=Kali}) (48)
\p{Script_Extensions: Khar} \p{Script_Extensions=Kharoshthi} (65)
\p{Script_Extensions: Kharoshthi} (Short: \p{Scx=Khar}) (65)
\p{Script_Extensions: Khmer} (Short: \p{Scx=Khmr}) (146)
\p{Script_Extensions: Khmr} \p{Script_Extensions=Khmer} (146)
\p{Script_Extensions: Knda} \p{Script_Extensions=Kannada} (86)
\p{Script_Extensions: Kthi} \p{Script_Extensions=Kaithi} (86)
\p{Script_Extensions: Lana} \p{Script_Extensions=Tai_Tham} (127)
\p{Script_Extensions: Lao} (Short: \p{Scx=Lao}) (67)
\p{Script_Extensions: Laoo} \p{Script_Extensions=Lao} (67)
\p{Script_Extensions: Latin} (Short: \p{Scx=Latn}) (1289)
\p{Script_Extensions: Latn} \p{Script_Extensions=Latin} (1289)
\p{Script_Extensions: Lepc} \p{Script_Extensions=Lepcha} (74)
\p{Script_Extensions: Lepcha} (Short: \p{Scx=Lepc}) (74)
\p{Script_Extensions: Limb} \p{Script_Extensions=Limbu} (66)
\p{Script_Extensions: Limbu} (Short: \p{Scx=Limb}) (66)
\p{Script_Extensions: Linb} \p{Script_Extensions=Linear_B} (268)
\p{Script_Extensions: Linear_B} (Short: \p{Scx=Linb}) (268)
\p{Script_Extensions: Lisu} (Short: \p{Scx=Lisu}) (48)
\p{Script_Extensions: Lyci} \p{Script_Extensions=Lycian} (29)
\p{Script_Extensions: Lycian} (Short: \p{Scx=Lyci}) (29)
\p{Script_Extensions: Lydi} \p{Script_Extensions=Lydian} (27)
\p{Script_Extensions: Lydian} (Short: \p{Scx=Lydi}) (27)
```

```
\p{Script_Extensions: Malayalam} (Short: \p{Scx=Mlym}) (98)
\p{Script_Extensions: Mand} \p{Script_Extensions=Mandaic} (30)
\p{Script_Extensions: Mandaic} (Short: \p{Scx=Mand}) (30)
\p{Script_Extensions: Meetei_Mayek} (Short: \p{Scx=Mtei}) (79)
\p{Script_Extensions: Merc} \p{Script_Extensions=Meroitic_Cursive}
                        (26)
\p{Script_Extensions: Mero} \p{Script_Extensions=
                        Meroitic_Hieroglyphs} (32)
\p{Script_Extensions: Meroitic_Cursive} (Short: \p{Scx=Merc}) (26)
\p{Script_Extensions: Meroitic_Hieroglyphs} (Short: \p{Scx=Mero})
                        (32)
\p{Script_Extensions: Miao} (Short: \p{Scx=Miao}) (133)
\p{Script_Extensions: Mlym} \p{Script_Extensions=Malayalam} (98)
\p{Script_Extensions: Mong} \p{Script_Extensions=Mongolian} (156)
\p{Script_Extensions: Mongolian} (Short: \p{Scx=Mong}) (156)
\p{Script_Extensions: Mtei} \p{Script_Extensions=Meetei_Mayek} (79)
\p{Script_Extensions: Myanmar} (Short: \p{Scx=Mymr}) (188)
\p{Script_Extensions: Mymr} \p{Script_Extensions=Myanmar} (188)
\p{Script_Extensions: New_Tai_Lue} (Short: \p{Scx=Talu}) (83)
\p{Script_Extensions: Nko} (Short: \p{Scx=Nko}) (59)
\p{Script_Extensions: Nkoo} \p{Script_Extensions=Nko} (59)
\p{Script_Extensions: Ogam} \p{Script_Extensions=Ogham} (29)
\p{Script_Extensions: Ogham} (Short: \p{Scx=Ogam}) (29)
\p{Script_Extensions: Ol_Chiki} (Short: \p{Scx=Olck}) (48)
\p{Script_Extensions: Olck} \p{Script_Extensions=Ol_Chiki} (48)
\p{Script_Extensions: Old_Italic} (Short: \p{Scx=Ital}) (35)
\p{Script_Extensions: Old_Persian} (Short: \p{Scx=Xpeo}) (50)
\p{Script_Extensions: Old_South_Arabian} (Short: \p{Scx=Sarb}) (32)
\p{Script_Extensions: Old_Turkic} (Short: \p{Scx=Orkh}) (73)
\p{Script_Extensions: Oriya} (Short: \p{Scx=Orya}) (92)
\p{Script_Extensions: Orkh} \p{Script_Extensions=Old_Turkic} (73)
\p{Script_Extensions: Orya} \p{Script_Extensions=Oriya} (92)
\p{Script_Extensions: Osma} \p{Script_Extensions=Osmanya} (40)
\p{Script_Extensions: Osmanya} (Short: \p{Scx=Osma}) (40)
\p{Script_Extensions: Phag} \p{Script_Extensions=Phags_Pa} (59)
\p{Script_Extensions: Phags_Pa} (Short: \p{Scx=Phag}) (59)
\p{Script_Extensions: Phli} \p{Script_Extensions=
                        Inscriptional_Pahlavi} (27)
\p{Script_Extensions: Phnx} \p{Script_Extensions=Phoenician} (29)
\p{Script_Extensions: Phoenician} (Short: \p{Scx=Phnx}) (29)
\p{Script_Extensions: Plrd} \p{Script_Extensions=Miao} (133)
\p{Script_Extensions: Prti} \p{Script_Extensions=
                        Inscriptional_Parthian} (30)
\p{Script_Extensions: Qaac} \p{Script_Extensions=Coptic} (137)
\p{Script_Extensions: Qaai} \p{Script_Extensions=Inherited} (459)
\p{Script_Extensions: Rejang} (Short: \p{Scx=Rjng}) (37)
\p{Script_Extensions: Rjng} \p{Script_Extensions=Rejang} (37)
\p{Script_Extensions: Runic} (Short: \p{Scx=Runr}) (78)
\p{Script_Extensions: Runr} \p{Script_Extensions=Runic} (78)
\p{Script_Extensions: Samaritan} (Short: \p{Scx=Samr}) (61)
\p{Script_Extensions: Samr} \p{Script_Extensions=Samaritan} (61)
\p{Script_Extensions: Sarb} \p{Script_Extensions=
                        Old_South_Arabian} (32)
\p{Script_Extensions: Saur} \p{Script_Extensions=Saurashtra} (81)
\p{Script_Extensions: Saurashtra} (Short: \p{Scx=Saur}) (81)
\p{Script_Extensions: Sharada} (Short: \p{Scx=Shrd}) (83)
```

```
\p{Script_Extensions: Shavian} (Short: \p{Scx=Shaw}) (48)
\p{Script_Extensions: Shaw} \p{Script_Extensions=Shavian} (48)
\p{Script_Extensions: Shrd} \p{Script_Extensions=Sharada} (83)
\p{Script_Extensions: Sinh} \p{Script_Extensions=Sinhala} (80)
\p{Script_Extensions: Sinhala} (Short: \p{Scx=Sinh}) (80)
\p{Script_Extensions: Sora} \p{Script_Extensions=Sora_Sompeng} (35)
\p{Script_Extensions: Sora_Sompeng} (Short: \p{Scx=Sora}) (35)
\p{Script_Extensions: Sund} \p{Script_Extensions=Sundanese} (72)
\p{Script_Extensions: Sundanese} (Short: \p{Scx=Sund}) (72)
\p{Script_Extensions: Sylo} \p{Script_Extensions=Syloti_Nagri} (54)
\p{Script_Extensions: Syloti_Nagri} (Short: \p{Scx=Sylo}) (54)
\p{Script_Extensions: Syrc} \p{Script_Extensions=Syriac} (94)
\p{Script_Extensions: Syriac} (Short: \p{Scx=Syrc}) (94)
\p{Script_Extensions: Tagalog} (Short: \p{Scx=Tglg}) (22)
\p{Script_Extensions: Tagb} \p{Script_Extensions=Tagbanwa} (20)
\p{Script_Extensions: Tagbanwa} (Short: \p{Scx=Tagb}) (20)
\p{Script_Extensions: Tai_Le} (Short: \p{Scx=Tale}) (45)
\p{Script_Extensions: Tai_Tham} (Short: \p{Scx=Lana}) (127)
\p{Script_Extensions: Tai_Viet} (Short: \p{Scx=Tavt}) (72)
\p{Script_Extensions: Takr} \p{Script_Extensions=Takri} (78)
\p{Script_Extensions: Takri} (Short: \p{Scx=Takr}) (78)
\p{Script_Extensions: Tale} \p{Script_Extensions=Tai_Le} (45)
\p{Script_Extensions: Talu} \p{Script_Extensions=New_Tai_Lue} (83)
\p{Script_Extensions: Tamil} (Short: \p{Scx=Taml}) (72)
\p{Script_Extensions: Taml} \p{Script_Extensions=Tamil} (72)
\p{Script_Extensions: Tavt} \p{Script_Extensions=Tai_Viet} (72)
\p{Script_Extensions: Telu} \p{Script_Extensions=Telugu} (93)
\p{Script_Extensions: Telugu} (Short: \p{Scx=Telu}) (93)
\p{Script_Extensions: Tfng} \p{Script_Extensions=Tifinagh} (59)
\p{Script_Extensions: Tglg} \p{Script_Extensions=Tagalog} (22)
\p{Script_Extensions: Thaa} \p{Script_Extensions=Thaana} (66)
\p{Script_Extensions: Thaana} (Short: \p{Scx=Thaa}) (66)
\p{Script_Extensions: Thai} (Short: \p{Scx=Thai}) (86)
\p{Script_Extensions: Tibetan} (Short: \p{Scx=Tibt}) (207)
\p{Script_Extensions: Tibt} \p{Script_Extensions=Tibetan} (207)
\p{Script_Extensions: Tifinagh} (Short: \p{Scx=Tfng}) (59)
\p{Script_Extensions: Ugar} \p{Script_Extensions=Ugaritic} (31)
\p{Script_Extensions: Ugaritic} (Short: \p{Scx=Ugar}) (31)
\p{Script_Extensions: Unknown} (Short: \p{Scx=Zzzz}) (1_003_925
                        plus all above-Unicode code points)
\p{Script_Extensions: Vai} (Short: \p{Scx=Vai}) (300)
\p{Script_Extensions: Vaii} \p{Script_Extensions=Vai} (300)
\p{Script_Extensions: Xpeo} \p{Script_Extensions=Old_Persian} (50)
\p{Script_Extensions: Xsux} \p{Script_Extensions=Cuneiform} (982)
\p{Script_Extensions: Yi} (Short: \p{Scx=Yi}) (1246)
\p{Script_Extensions: Yiii} \p{Script_Extensions=Yi} (1246)
\p{Script_Extensions: Zinh} \p{Script_Extensions=Inherited} (459)
\p{Script_Extensions: Zyyy} \p{Script_Extensions=Common} (6061)
\p{Script_Extensions: Zzzz} \p{Script_Extensions=Unknown}
                        (1_003_925 plus all above-Unicode code
                        points)
\p{Scx: *}              \p{Script_Extensions: *}
\p{SD}                  \p{Soft_Dotted} (= \p{Soft_Dotted=Y}) (46)
\p{SD: *}               \p{Soft_Dotted: *}
\p{Sentence_Break: AT}  \p{Sentence_Break=ATerm} (4)
\p{Sentence_Break: ATerm} (Short: \p{SB=AT}) (4)
```

```
    \p{Sentence_Break: CL}  \p{Sentence_Break=Close} (181)
    \p{Sentence_Break: Close} (Short: \p{SB=CL}) (181)
    \p{Sentence_Break: CR}  (Short: \p{SB=CR}) (1)
    \p{Sentence_Break: EX}  \p{Sentence_Break=Extend} (1649)
    \p{Sentence_Break: Extend} (Short: \p{SB=EX}) (1649)
    \p{Sentence_Break: FO}  \p{Sentence_Break=Format} (143)
    \p{Sentence_Break: Format} (Short: \p{SB=FO}) (143)
    \p{Sentence_Break: LE}  \p{Sentence_Break=OLetter} (97_841)
    \p{Sentence_Break: LF}  (Short: \p{SB=LF}) (1)
    \p{Sentence_Break: LO}  \p{Sentence_Break=Lower} (1933)
    \p{Sentence_Break: Lower} (Short: \p{SB=LO}) (1933)
    \p{Sentence_Break: NU}  \p{Sentence_Break=Numeric} (452)
    \p{Sentence_Break: Numeric} (Short: \p{SB=NU}) (452)
    \p{Sentence_Break: OLetter} (Short: \p{SB=LE}) (97_841)
    \p{Sentence_Break: Other} (Short: \p{SB=XX}) (1_010_264 plus all
                              above-Unicode code points)
    \p{Sentence_Break: SC}  \p{Sentence_Break=SContinue} (26)
    \p{Sentence_Break: SContinue} (Short: \p{SB=SC}) (26)
    \p{Sentence_Break: SE}  \p{Sentence_Break=Sep} (3)
    \p{Sentence_Break: Sep} (Short: \p{SB=SE}) (3)
    \p{Sentence_Break: Sp}  (Short: \p{SB=Sp}) (20)
    \p{Sentence_Break: ST}  \p{Sentence_Break=STerm} (80)
    \p{Sentence_Break: STerm} (Short: \p{SB=ST}) (80)
    \p{Sentence_Break: UP}  \p{Sentence_Break=Upper} (1514)
    \p{Sentence_Break: Upper} (Short: \p{SB=UP}) (1514)
    \p{Sentence_Break: XX}  \p{Sentence_Break=Other} (1_010_264 plus
                              all above-Unicode code points)
    \p{Separator}           \p{General_Category=Separator} (Short:
                              \p{Z}) (19)
    \p{Sharada}             \p{Script=Sharada} (Short: \p{Shrd}; NOT
                              \p{Block=Sharada}) (83)
    \p{Shavian}             \p{Script=Shavian} (Short: \p{Shaw}) (48)
    \p{Shaw}                \p{Shavian} (= \p{Script=Shavian}) (48)
    \p{Shrd}                \p{Sharada} (= \p{Script=Sharada}) (NOT
                              \p{Block=Sharada}) (83)
    \p{Sinh}                \p{Sinhala} (= \p{Script=Sinhala}) (NOT
                              \p{Block=Sinhala}) (80)
    \p{Sinhala}             \p{Script=Sinhala} (Short: \p{Sinh}; NOT
                              \p{Block=Sinhala}) (80)
    \p{Sk}                  \p{Modifier_Symbol} (=
                              \p{General_Category=Modifier_Symbol})
                              (115)
    \p{Sm}                  \p{Math_Symbol} (= \p{General_Category=
                              Math_Symbol}) (948)
X   \p{Small_Form_Variants} \p{Block=Small_Form_Variants} (Short:
                              \p{InSmallForms}) (32)
X   \p{Small_Forms}         \p{Small_Form_Variants} (= \p{Block=
                              Small_Form_Variants}) (32)
    \p{So}                  \p{Other_Symbol} (= \p{General_Category=
                              Other_Symbol}) (4404)
    \p{Soft_Dotted}         \p{Soft_Dotted=Y} (Short: \p{SD}) (46)
    \p{Soft_Dotted: N*}     (Short: \p{SD=N}, \P{SD}) (1_114_066 plus
                              all above-Unicode code points)
    \p{Soft_Dotted: Y*}     (Short: \p{SD=Y}, \p{SD}) (46)
    \p{Sora}                \p{Sora_Sompeng} (= \p{Script=
                              Sora_Sompeng}) (NOT \p{Block=
```

```
                              Sora_Sompeng}) (35)
    \p{Sora_Sompeng}        \p{Script=Sora_Sompeng} (Short: \p{Sora};
                              NOT \p{Block=Sora_Sompeng}) (35)
    \p{Space}               \p{White_Space=Y} \s including beyond
                              ASCII and vertical tab (25)
    \p{Space: *}            \p{White_Space: *}
    \p{Space_Separator}     \p{General_Category=Space_Separator}
                              (Short: \p{Zs}) (17)
    \p{SpacePerl}           \p{XPerlSpace} (25)
    \p{Spacing_Mark}        \p{General_Category=Spacing_Mark} (Short:
                              \p{Mc}) (352)
  X \p{Spacing_Modifier_Letters} \p{Block=Spacing_Modifier_Letters}
                              (Short: \p{InModifierLetters}) (80)
  X \p{Specials}            \p{Block=Specials} (16)
    \p{STerm}               \p{STerm=Y} (83)
    \p{STerm: N*}           (Single: \P{STerm}) (1_114_029 plus all
                              above-Unicode code points)
    \p{STerm: Y*}           (Single: \p{STerm}) (83)
    \p{Sund}                \p{Sundanese} (= \p{Script=Sundanese})
                              (NOT \p{Block=Sundanese}) (72)
    \p{Sundanese}           \p{Script=Sundanese} (Short: \p{Sund}; NOT
                              \p{Block=Sundanese}) (72)
  X \p{Sundanese_Sup}       \p{Sundanese_Supplement} (= \p{Block=
                              Sundanese_Supplement}) (16)
  X \p{Sundanese_Supplement} \p{Block=Sundanese_Supplement} (Short:
                              \p{InSundaneseSup}) (16)
  X \p{Sup_Arrows_A}        \p{Supplemental_Arrows_A} (= \p{Block=
                              Supplemental_Arrows_A}) (16)
  X \p{Sup_Arrows_B}        \p{Supplemental_Arrows_B} (= \p{Block=
                              Supplemental_Arrows_B}) (128)
  X \p{Sup_Math_Operators}  \p{Supplemental_Mathematical_Operators} (=
                              \p{Block=
                              Supplemental_Mathematical_Operators})
                              (256)
  X \p{Sup_PUA_A}           \p{Supplementary_Private_Use_Area_A} (=
                              \p{Block=
                              Supplementary_Private_Use_Area_A})
                              (65_536)
  X \p{Sup_PUA_B}           \p{Supplementary_Private_Use_Area_B} (=
                              \p{Block=
                              Supplementary_Private_Use_Area_B})
                              (65_536)
  X \p{Sup_Punctuation}     \p{Supplemental_Punctuation} (= \p{Block=
                              Supplemental_Punctuation}) (128)
  X \p{Super_And_Sub}       \p{Superscripts_And_Subscripts} (=
                              \p{Block=Superscripts_And_Subscripts})
                              (48)
  X \p{Superscripts_And_Subscripts} \p{Block=
                              Superscripts_And_Subscripts} (Short:
                              \p{InSuperAndSub}) (48)
  X \p{Supplemental_Arrows_A} \p{Block=Supplemental_Arrows_A} (Short:
                              \p{InSupArrowsA}) (16)
  X \p{Supplemental_Arrows_B} \p{Block=Supplemental_Arrows_B} (Short:
                              \p{InSupArrowsB}) (128)
  X \p{Supplemental_Mathematical_Operators} \p{Block=
                              Supplemental_Mathematical_Operators}
```

```
                              (Short: \p{InSupMathOperators}) (256)
X \p{Supplemental_Punctuation} \p{Block=Supplemental_Punctuation}
                              (Short: \p{InSupPunctuation}) (128)
X \p{Supplementary_Private_Use_Area_A} \p{Block=
                              Supplementary_Private_Use_Area_A}
                              (Short: \p{InSupPUAA}) (65_536)
X \p{Supplementary_Private_Use_Area_B} \p{Block=
                              Supplementary_Private_Use_Area_B}
                              (Short: \p{InSupPUAB}) (65_536)
  \p{Surrogate}           \p{General_Category=Surrogate} (Short:
                              \p{Cs}) (2048)
  \p{Sylo}                \p{Syloti_Nagri} (= \p{Script=
                              Syloti_Nagri}) (NOT \p{Block=
                              Syloti_Nagri}) (44)
  \p{Syloti_Nagri}        \p{Script=Syloti_Nagri} (Short: \p{Sylo};
                              NOT \p{Block=Syloti_Nagri}) (44)
  \p{Symbol}              \p{General_Category=Symbol} (Short: \p{S})
                              (5516)
  \p{Syrc}                \p{Syriac} (= \p{Script=Syriac}) (NOT
                              \p{Block=Syriac}) (77)
  \p{Syriac}              \p{Script=Syriac} (Short: \p{Syrc}; NOT
                              \p{Block=Syriac}) (77)
  \p{Tagalog}             \p{Script=Tagalog} (Short: \p{Tglg}; NOT
                              \p{Block=Tagalog}) (20)
  \p{Tagb}                \p{Tagbanwa} (= \p{Script=Tagbanwa}) (NOT
                              \p{Block=Tagbanwa}) (18)
  \p{Tagbanwa}            \p{Script=Tagbanwa} (Short: \p{Tagb}; NOT
                              \p{Block=Tagbanwa}) (18)
X \p{Tags}                \p{Block=Tags} (128)
  \p{Tai_Le}              \p{Script=Tai_Le} (Short: \p{Tale}; NOT
                              \p{Block=Tai_Le}) (35)
  \p{Tai_Tham}            \p{Script=Tai_Tham} (Short: \p{Lana}; NOT
                              \p{Block=Tai_Tham}) (127)
  \p{Tai_Viet}            \p{Script=Tai_Viet} (Short: \p{Tavt}; NOT
                              \p{Block=Tai_Viet}) (72)
X \p{Tai_Xuan_Jing}       \p{Tai_Xuan_Jing_Symbols} (= \p{Block=
                              Tai_Xuan_Jing_Symbols}) (96)
X \p{Tai_Xuan_Jing_Symbols} \p{Block=Tai_Xuan_Jing_Symbols} (Short:
                              \p{InTaiXuanJing}) (96)
  \p{Takr}                \p{Takri} (= \p{Script=Takri}) (NOT
                              \p{Block=Takri}) (66)
  \p{Takri}               \p{Script=Takri} (Short: \p{Takr}; NOT
                              \p{Block=Takri}) (66)
  \p{Tale}                \p{Tai_Le} (= \p{Script=Tai_Le}) (NOT
                              \p{Block=Tai_Le}) (35)
  \p{Talu}                \p{New_Tai_Lue} (= \p{Script=New_Tai_Lue})
                              (NOT \p{Block=New_Tai_Lue}) (83)
  \p{Tamil}               \p{Script=Tamil} (Short: \p{Taml}; NOT
                              \p{Block=Tamil}) (72)
  \p{Taml}                \p{Tamil} (= \p{Script=Tamil}) (NOT
                              \p{Block=Tamil}) (72)
  \p{Tavt}                \p{Tai_Viet} (= \p{Script=Tai_Viet}) (NOT
                              \p{Block=Tai_Viet}) (72)
  \p{Telu}                \p{Telugu} (= \p{Script=Telugu}) (NOT
                              \p{Block=Telugu}) (93)
  \p{Telugu}              \p{Script=Telugu} (Short: \p{Telu}; NOT
```

```
                              \p{Block=Telugu}) (93)
    \p{Term}                 \p{Terminal_Punctuation} (=
                              \p{Terminal_Punctuation=Y}) (176)
    \p{Term: *}              \p{Terminal_Punctuation: *}
    \p{Terminal_Punctuation} \p{Terminal_Punctuation=Y} (Short:
                              \p{Term}) (176)
    \p{Terminal_Punctuation: N*} (Short: \p{Term=N}, \P{Term})
                              (1_113_936 plus all above-Unicode code
                              points)
    \p{Terminal_Punctuation: Y*} (Short: \p{Term=Y}, \p{Term}) (176)
    \p{Tfng}                 \p{Tifinagh} (= \p{Script=Tifinagh}) (NOT
                              \p{Block=Tifinagh}) (59)
    \p{Tglg}                 \p{Tagalog} (= \p{Script=Tagalog}) (NOT
                              \p{Block=Tagalog}) (20)
    \p{Thaa}                 \p{Thaana} (= \p{Script=Thaana}) (NOT
                              \p{Block=Thaana}) (50)
    \p{Thaana}               \p{Script=Thaana} (Short: \p{Thaa}; NOT
                              \p{Block=Thaana}) (50)
    \p{Thai}                 \p{Script=Thai} (NOT \p{Block=Thai}) (86)
    \p{Tibetan}              \p{Script=Tibetan} (Short: \p{Tibt}; NOT
                              \p{Block=Tibetan}) (207)
    \p{Tibt}                 \p{Tibetan} (= \p{Script=Tibetan}) (NOT
                              \p{Block=Tibetan}) (207)
    \p{Tifinagh}             \p{Script=Tifinagh} (Short: \p{Tfng}; NOT
                              \p{Block=Tifinagh}) (59)
    \p{Title}                \p{Titlecase} (/i= Cased=Yes) (31)
    \p{Titlecase}            (= \p{Gc=Lt}) (Short: \p{Title}; /i=
                              Cased=Yes) (31)
    \p{Titlecase_Letter}     \p{General_Category=Titlecase_Letter}
                              (Short: \p{Lt}; /i= General_Category=
                              Cased_Letter) (31)
X   \p{Transport_And_Map}    \p{Transport_And_Map_Symbols} (= \p{Block=
                              Transport_And_Map_Symbols}) (128)
X   \p{Transport_And_Map_Symbols} \p{Block=Transport_And_Map_Symbols}
                              (Short: \p{InTransportAndMap}) (128)
X   \p{UCAS}                 \p{Unified_Canadian_Aboriginal_Syllabics}
                              (= \p{Block=
                              Unified_Canadian_Aboriginal_Syllabics})
                              (640)
X   \p{UCAS_Ext}             \p{Unified_Canadian_Aboriginal_Syllabics_-
                              Extended} (= \p{Block=
                              Unified_Canadian_Aboriginal_Syllabics_-
                              Extended}) (80)
    \p{Ugar}                 \p{Ugaritic} (= \p{Script=Ugaritic}) (NOT
                              \p{Block=Ugaritic}) (31)
    \p{Ugaritic}             \p{Script=Ugaritic} (Short: \p{Ugar}; NOT
                              \p{Block=Ugaritic}) (31)
    \p{UIdeo}                \p{Unified_Ideograph} (=
                              \p{Unified_Ideograph=Y}) (74_617)
    \p{UIdeo: *}             \p{Unified_Ideograph: *}
    \p{Unassigned}           \p{General_Category=Unassigned} (Short:
                              \p{Cn}) (864_409 plus all above-Unicode
                              code points)
    \p{Unicode}              \p{Any} (1_114_112)
X   \p{Unified_Canadian_Aboriginal_Syllabics} \p{Block=
                              Unified_Canadian_Aboriginal_Syllabics}
```

```
                              (Short: \p{InUCAS}) (640)
X \p{Unified_Canadian_Aboriginal_Syllabics_Extended} \p{Block=
                              Unified_Canadian_Aboriginal_Syllabics_-
                              Extended} (Short: \p{InUCASExt}) (80)
  \p{Unified_Ideograph}     \p{Unified_Ideograph=Y} (Short: \p{UIdeo})
                              (74_617)
  \p{Unified_Ideograph: N*} (Short: \p{UIdeo=N}, \P{UIdeo})
                              (1_039_495 plus all above-Unicode code
                              points)
  \p{Unified_Ideograph: Y*} (Short: \p{UIdeo=Y}, \p{UIdeo}) (74_617)
  \p{Unknown}               \p{Script=Unknown} (Short: \p{Zzzz})
                              (1_003_925 plus all above-Unicode code
                              points)
  \p{Upper}                 \p{Uppercase=Y} (/i= Cased=Yes) (1483)
  \p{Upper: *}              \p{Uppercase: *}
  \p{Uppercase}             \p{Upper} (= \p{Uppercase=Y}) (/i= Cased=
                              Yes) (1483)
  \p{Uppercase: N*}         (Short: \p{Upper=N}, \P{Upper}; /i= Cased=
                              No) (1_112_629 plus all above-Unicode
                              code points)
  \p{Uppercase: Y*}         (Short: \p{Upper=Y}, \p{Upper}; /i= Cased=
                              Yes) (1483)
  \p{Uppercase_Letter}      \p{General_Category=Uppercase_Letter}
                              (Short: \p{Lu}; /i= General_Category=
                              Cased_Letter) (1441)
  \p{Vai}                   \p{Script=Vai} (NOT \p{Block=Vai}) (300)
  \p{Vaii}                  \p{Vai} (= \p{Script=Vai}) (NOT \p{Block=
                              Vai}) (300)
  \p{Variation_Selector}    \p{Variation_Selector=Y} (Short: \p{VS};
                              NOT \p{Variation_Selectors}) (259)
  \p{Variation_Selector: N*} (Short: \p{VS=N}, \P{VS}) (1_113_853
                              plus all above-Unicode code points)
  \p{Variation_Selector: Y*} (Short: \p{VS=Y}, \p{VS}) (259)
X \p{Variation_Selectors}   \p{Block=Variation_Selectors} (Short:
                              \p{InVS}) (16)
X \p{Variation_Selectors_Supplement} \p{Block=
                              Variation_Selectors_Supplement} (Short:
                              \p{InVSSup}) (240)
X \p{Vedic_Ext}             \p{Vedic_Extensions} (= \p{Block=
                              Vedic_Extensions}) (48)
X \p{Vedic_Extensions}      \p{Block=Vedic_Extensions} (Short:
                              \p{InVedicExt}) (48)
X \p{Vertical_Forms}        \p{Block=Vertical_Forms} (16)
  \p{VertSpace}             \v (7)
  \p{VS}                    \p{Variation_Selector} (=
                              \p{Variation_Selector=Y}) (NOT
                              \p{Variation_Selectors}) (259)
  \p{VS: *}                 \p{Variation_Selector: *}
X \p{VS_Sup}                \p{Variation_Selectors_Supplement} (=
                              \p{Block=
                              Variation_Selectors_Supplement}) (240)
  \p{WB: *}                 \p{Word_Break: *}
  \p{White_Space}           \p{White_Space=Y} (Short: \p{WSpace}) (25)
  \p{White_Space: N*}       (Short: \p{Space=N}, \P{WSpace})
                              (1_114_087 plus all above-Unicode code
                              points)
```

```
\p{White_Space: Y*}     (Short: \p{Space=Y}, \p{WSpace}) (25)
\p{Word}                \w, including beyond ASCII; = \p{Alnum} +
                          \pM + \p{Pc} (103_406)
\p{Word_Break: ALetter} (Short: \p{WB=LE}) (24_867)
\p{Word_Break: CR}      (Short: \p{WB=CR}) (1)
\p{Word_Break: Double_Quote} (Short: \p{WB=DQ}) (1)
\p{Word_Break: DQ}      \p{Word_Break=Double_Quote} (1)
\p{Word_Break: EX}      \p{Word_Break=ExtendNumLet} (10)
\p{Word_Break: Extend}  (Short: \p{WB=Extend}) (1649)
\p{Word_Break: ExtendNumLet} (Short: \p{WB=EX}) (10)
\p{Word_Break: FO}      \p{Word_Break=Format} (142)
\p{Word_Break: Format}  (Short: \p{WB=FO}) (142)
\p{Word_Break: Hebrew_Letter} (Short: \p{WB=HL}) (74)
\p{Word_Break: HL}      \p{Word_Break=Hebrew_Letter} (74)
\p{Word_Break: KA}      \p{Word_Break=Katakana} (310)
\p{Word_Break: Katakana} (Short: \p{WB=KA}) (310)
\p{Word_Break: LE}      \p{Word_Break=ALetter} (24_867)
\p{Word_Break: LF}      (Short: \p{WB=LF}) (1)
\p{Word_Break: MB}      \p{Word_Break=MidNumLet} (7)
\p{Word_Break: MidLetter} (Short: \p{WB=ML}) (9)
\p{Word_Break: MidNum}  (Short: \p{WB=MN}) (15)
\p{Word_Break: MidNumLet} (Short: \p{WB=MB}) (7)
\p{Word_Break: ML}      \p{Word_Break=MidLetter} (9)
\p{Word_Break: MN}      \p{Word_Break=MidNum} (15)
\p{Word_Break: Newline} (Short: \p{WB=NL}) (5)
\p{Word_Break: NL}      \p{Word_Break=Newline} (5)
\p{Word_Break: NU}      \p{Word_Break=Numeric} (451)
\p{Word_Break: Numeric} (Short: \p{WB=NU}) (451)
\p{Word_Break: Other}   (Short: \p{WB=XX}) (1_086_543 plus all
                          above-Unicode code points)
\p{Word_Break: Regional_Indicator} (Short: \p{WB=RI}) (26)
\p{Word_Break: RI}      \p{Word_Break=Regional_Indicator} (26)
\p{Word_Break: Single_Quote} (Short: \p{WB=SQ}) (1)
\p{Word_Break: SQ}      \p{Word_Break=Single_Quote} (1)
\p{Word_Break: XX}      \p{Word_Break=Other} (1_086_543 plus all
                          above-Unicode code points)
\p{WSpace}              \p{White_Space} (= \p{White_Space=Y}) (25)
\p{WSpace: *}           \p{White_Space: *}
\p{XDigit}              \p{Hex_Digit=Y} (Short: \p{Hex}) (44)
\p{XID_Continue}        \p{XID_Continue=Y} (Short: \p{XIDC})
                          (103_336)
\p{XID_Continue: N*}    (Short: \p{XIDC=N}, \P{XIDC}) (1_010_776
                          plus all above-Unicode code points)
\p{XID_Continue: Y*}    (Short: \p{XIDC=Y}, \p{XIDC}) (103_336)
\p{XID_Start}           \p{XID_Start=Y} (Short: \p{XIDS}) (101_217)
\p{XID_Start: N*}       (Short: \p{XIDS=N}, \P{XIDS}) (1_012_895
                          plus all above-Unicode code points)
\p{XID_Start: Y*}       (Short: \p{XIDS=Y}, \p{XIDS}) (101_217)
\p{XIDC}                \p{XID_Continue} (= \p{XID_Continue=Y})
                          (103_336)
\p{XIDC: *}             \p{XID_Continue: *}
\p{XIDS}                \p{XID_Start} (= \p{XID_Start=Y}) (101_217)
\p{XIDS: *}             \p{XID_Start: *}
\p{Xpeo}                \p{Old_Persian} (= \p{Script=Old_Persian})
                          (NOT \p{Block=Old_Persian}) (50)
\p{XPerlSpace}          \s, including beyond ASCII (Short:
```

```
                                \p{SpacePerl}) (25)
  \p{XPosixAlnum}           \p{Alnum} (102_619)
  \p{XPosixAlpha}           \p{Alpha} (= \p{Alphabetic=Y}) (102_159)
  \p{XPosixBlank}           \p{Blank} (18)
  \p{XPosixCntrl}           \p{Cntrl} (= \p{General_Category=Control})
                              (65)
  \p{XPosixDigit}           \p{Digit} (= \p{General_Category=
                              Decimal_Number}) (460)
  \p{XPosixGraph}           \p{Graph} (247_571)
  \p{XPosixLower}           \p{Lower} (= \p{Lowercase=Y}) (/i= Cased=
                              Yes) (1934)
  \p{XPosixPrint}           \p{Print} (247_588)
  \p{XPosixPunct}           \p{Punct} + ASCII-range \p{Symbol} (645)
  \p{XPosixSpace}           \p{Space} (= \p{White_Space=Y}) (25)
  \p{XPosixUpper}           \p{Upper} (= \p{Uppercase=Y}) (/i= Cased=
                              Yes) (1483)
  \p{XPosixWord}            \p{Word} (103_406)
  \p{XPosixXDigit}          \p{XDigit} (= \p{Hex_Digit=Y}) (44)
  \p{Xsux}                  \p{Cuneiform} (= \p{Script=Cuneiform})
                              (NOT \p{Block=Cuneiform}) (982)
  \p{Yi}                    \p{Script=Yi} (1220)
X \p{Yi_Radicals}           \p{Block=Yi_Radicals} (64)
X \p{Yi_Syllables}          \p{Block=Yi_Syllables} (1168)
  \p{Yiii}                  \p{Yi} (= \p{Script=Yi}) (1220)
X \p{Yijing}                \p{Yijing_Hexagram_Symbols} (= \p{Block=
                              Yijing_Hexagram_Symbols}) (64)
X \p{Yijing_Hexagram_Symbols} \p{Block=Yijing_Hexagram_Symbols}
                              (Short: \p{InYijing}) (64)
  \p{Z} \pZ                 \p{Separator} (= \p{General_Category=
                              Separator}) (19)
  \p{Zinh}                  \p{Inherited} (= \p{Script=Inherited})
                              (523)
  \p{Zl}                    \p{Line_Separator} (= \p{General_Category=
                              Line_Separator}) (1)
  \p{Zp}                    \p{Paragraph_Separator} (=
                              \p{General_Category=
                              Paragraph_Separator}) (1)
  \p{Zs}                    \p{Space_Separator} (=
                              \p{General_Category=Space_Separator})
                              (17)
  \p{Zyyy}                  \p{Common} (= \p{Script=Common}) (6418)
  \p{Zzzz}                  \p{Unknown} (= \p{Script=Unknown})
                              (1_003_925 plus all above-Unicode code
                              points)
 TX\p{_CanonDCIJ}           (For internal use by Perl, not necessarily
                              stable) (= \p{Soft_Dotted=Y}) (46)
 TX\p{_Case_Ignorable}      (For internal use by Perl, not necessarily
                              stable) (= \p{Case_Ignorable=Y}) (1806)
 TX\p{_CombAbove}           (For internal use by Perl, not necessarily
                              stable) (= \p{Canonical_Combining_Class=
                              Above}) (349)
```

Legal \p{} and \P{} constructs that match no characters

Unicode has some property-value pairs that currently don't match anything. This happens generally either because they are obsolete, or they exist for symmetry with other forms, but no language has

yet been encoded that uses them. In this version of Unicode, the following match zero code points:

\p{Canonical_Combining_Class=Attached_Below_Left}

\p{Canonical_Combining_Class=CCC133}

\p{Grapheme_Cluster_Break=Prepend}

Properties accessible through Unicode::UCD

All the Unicode character properties mentioned above (except for those marked as for internal use by Perl) are also accessible by *"prop_invlist()" in Unicode::UCD*.

Due to their nature, not all Unicode character properties are suitable for regular expression matches, nor `prop_invlist()`. The remaining non-provisional, non-internal ones are accessible via *"prop_invmap()" in Unicode::UCD* (except for those that this Perl installation hasn't included; see *below for which those are*).

For compatibility with other parts of Perl, all the single forms given in the table in the *section above* are recognized. BUT, there are some ambiguities between some Perl extensions and the Unicode properties, all of which are silently resolved in favor of the official Unicode property. To avoid surprises, you should only use `prop_invmap()` for forms listed in the table below, which omits the non-recommended ones. The affected forms are the Perl single form equivalents of Unicode properties, such as `\p{sc}` being a single-form equivalent of `\p{gc=sc}`, which is treated by `prop_invmap()` as the `Script` property, whose short name is `sc`. The table indicates the current ambiguities in the INFO column, beginning with the word `"NOT"`.

The standard Unicode properties listed below are documented in *http://www.unicode.org/reports/tr44/*; Perl_Decimal_Digit is documented in *"prop_invmap()" in Unicode::UCD*. The other Perl extensions are in *"Other Properties" in perlunicode*;

The first column in the table is a name for the property; the second column is an alternative name, if any, plus possibly some annotations. The alternative name is the property's full name, unless that would simply repeat the first column, in which case the second column indicates the property's short name (if different). The annotations are given only in the entry for the full name. If a property is obsolete, etc, the entry will be flagged with the same characters used in the table in the *section above* , like **D** or **S**.

```
NAME                    INFO

Age
AHex                  ASCII_Hex_Digit
All                   (Perl extension).  All code points,
                      including those above Unicode.  Same as
                      qr/./s
Alnum                 (Perl extension).  Alphabetic and
                      (decimal) Numeric
Alpha                 Alphabetic
Alphabetic            (Short: Alpha)
Any                   (Perl extension).  All Unicode code
                      points: [\x{0000}-\x{10FFFF}]
ASCII                 Block=ASCII.  (Perl extension).
                      [[:ASCII:]]
ASCII_Hex_Digit       (Short: AHex)
Assigned              (Perl extension).  All assigned code points
Bc                    Bidi_Class
Bidi_C                Bidi_Control
Bidi_Class            (Short: bc)
Bidi_Control          (Short: Bidi_C)
Bidi_M                Bidi_Mirrored
```

```
Bidi_Mirrored           (Short: Bidi_M)
Bidi_Mirroring_Glyph    (Short: bmg)
Bidi_Paired_Bracket     (Short: bpb)
Bidi_Paired_Bracket_Type (Short: bpt)
Blank                   (Perl extension).  \h, Horizontal white
                        space
Blk                     Block
Block                   (Short: blk)
Bmg                     Bidi_Mirroring_Glyph
Bpb                     Bidi_Paired_Bracket
Bpt                     Bidi_Paired_Bracket_Type
Canonical_Combining_Class (Short: ccc)
Case_Folding            (Short: cf)
Case_Ignorable          (Short: CI)
Cased
Category                General_Category
Ccc                     Canonical_Combining_Class
CE                      Composition_Exclusion
Cf                      Case_Folding; NOT 'cf' meaning
                        'General_Category=Format'
Changes_When_Casefolded (Short: CWCF)
Changes_When_Casemapped (Short: CWCM)
Changes_When_Lowercased (Short: CWL)
Changes_When_NFKC_Casefolded (Short: CWKCF)
Changes_When_Titlecased (Short: CWT)
Changes_When_Uppercased (Short: CWU)
CI                      Case_Ignorable
Cntrl                   General_Category=Cntrl.  (Perl extension).
                        Control characters
Comp_Ex                 Full_Composition_Exclusion
Composition_Exclusion   (Short: CE)
CWCF                    Changes_When_Casefolded
CWCM                    Changes_When_Casemapped
CWKCF                   Changes_When_NFKC_Casefolded
CWL                     Changes_When_Lowercased
CWT                     Changes_When_Titlecased
CWU                     Changes_When_Uppercased
Dash
Decomposition_Mapping   (Short: dm)
Decomposition_Type      (Short: dt)
Default_Ignorable_Code_Point (Short: DI)
Dep                     Deprecated
Deprecated              (Short: Dep)
DI                      Default_Ignorable_Code_Point
Dia                     Diacritic
Diacritic               (Short: Dia)
Digit                   General_Category=Digit.  (Perl extension).
                        [0-9] + all other decimal digits
Dm                      Decomposition_Mapping
Dt                      Decomposition_Type
Ea                      East_Asian_Width
East_Asian_Width        (Short: ea)
Ext                     Extender
Extender                (Short: Ext)
Full_Composition_Exclusion (Short: Comp_Ex)
Gc                      General_Category
```

```
    GCB                     Grapheme_Cluster_Break
    General_Category        (Short: gc)
    Gr_Base                 Grapheme_Base
    Gr_Ext                  Grapheme_Extend
    Graph                   (Perl extension).  Characters that are
                            graphical
    Grapheme_Base           (Short: Gr_Base)
    Grapheme_Cluster_Break  (Short: GCB)
    Grapheme_Extend         (Short: Gr_Ext)
    Hangul_Syllable_Type    (Short: hst)
    Hex                     Hex_Digit
    Hex_Digit               (Short: Hex)
    HorizSpace              Blank.  (Perl extension)
    Hst                     Hangul_Syllable_Type
  D Hyphen                  Supplanted by Line_Break property values;
                            see www.unicode.org/reports/tr14
    ID_Continue             (Short: IDC)
    ID_Start                (Short: IDS)
    IDC                     ID_Continue
    Ideo                    Ideographic
    Ideographic             (Short: Ideo)
    IDS                     ID_Start
    IDS_Binary_Operator     (Short: IDSB)
    IDS_Trinary_Operator    (Short: IDST)
    IDSB                    IDS_Binary_Operator
    IDST                    IDS_Trinary_Operator
    In                      Present_In.  (Perl extension)
    Isc                     ISO_Comment; NOT 'isc' meaning
                            'General_Category=Other'
    ISO_Comment             (Short: isc)
    Jg                      Joining_Group
    Join_C                  Join_Control
    Join_Control            (Short: Join_C)
    Joining_Group           (Short: jg)
    Joining_Type            (Short: jt)
    Jt                      Joining_Type
    Lb                      Line_Break
    Lc                      Lowercase_Mapping; NOT 'lc' meaning
                            'General_Category=Cased_Letter'
    Line_Break              (Short: lb)
    LOE                     Logical_Order_Exception
    Logical_Order_Exception (Short: LOE)
    Lower                   Lowercase
    Lowercase               (Short: Lower)
    Lowercase_Mapping       (Short: lc)
    Math
    Na                      Name
    Na1                     Unicode_1_Name
    Name                    (Short: na)
    Name_Alias
    NChar                   Noncharacter_Code_Point
    NFC_QC                  NFC_Quick_Check
    NFC_Quick_Check         (Short: NFC_QC)
    NFD_QC                  NFD_Quick_Check
    NFD_Quick_Check         (Short: NFD_QC)
    NFKC_Casefold           (Short: NFKC_CF)
```

```
NFKC_CF                  NFKC_Casefold
NFKC_QC                  NFKC_Quick_Check
NFKC_Quick_Check         (Short: NFKC_QC)
NFKD_QC                  NFKD_Quick_Check
NFKD_Quick_Check         (Short: NFKD_QC)
Noncharacter_Code_Point (Short: NChar)
Nt                       Numeric_Type
Numeric_Type             (Short: nt)
Numeric_Value            (Short: nv)
Nv                       Numeric_Value
Pat_Syn                  Pattern_Syntax
Pat_WS                   Pattern_White_Space
Pattern_Syntax           (Short: Pat_Syn)
Pattern_White_Space      (Short: Pat_WS)
Perl_Decimal_Digit       (Perl extension)
PerlSpace                (Perl extension).  \s, restricted to ASCII
                         = [ \f\n\r\t] plus vertical tab
PerlWord                 (Perl extension).  \w, restricted to ASCII
                         = [A-Za-z0-9_]
PosixAlnum               (Perl extension).  [A-Za-z0-9]
PosixAlpha               (Perl extension).  [A-Za-z]
PosixBlank               (Perl extension).  \t and ' '
PosixCntrl               (Perl extension).  ASCII control
                         characters: NUL, SOH, STX, ETX, EOT, ENQ,
                         ACK, BEL, BS, HT, LF, VT, FF, CR, SO, SI,
                         DLE, DC1, DC2, DC3, DC4, NAK, SYN, ETB,
                         CAN, EOM, SUB, ESC, FS, GS, RS, US, and DEL
PosixDigit               (Perl extension).  [0-9]
PosixGraph               (Perl extension).  [-!"#$%&'()*+,./:;<=
                         >?@[\\]^_`{|}~0-9A-Za-z]
PosixLower               (Perl extension).  [a-z]
PosixPrint               (Perl extension).  [- 0-9A-Za-
                         z!"#$%&'()*+,./:;<=>?@[\\]^_`{|}~]
PosixPunct               (Perl extension).  [-!"#$%&'()*+,./:;<=
                         >?@[\\]^_`{|}~]
PosixSpace               (Perl extension).  \t, \n, \cK, \f, \r,
                         and ' '.  (\cK is vertical tab)
PosixUpper               (Perl extension).  [A-Z]
PosixWord                PerlWord.  (Perl extension)
PosixXDigit              (Perl extension).  [0-9A-Fa-f]
Present_In               (Short: In).  (Perl extension)
Print                    (Perl extension).  Characters that are
                         graphical plus space characters (but no
                         controls)
Punct                    General_Category=Punct.  (Perl extension)
QMark                    Quotation_Mark
Quotation_Mark           (Short: QMark)
Radical
SB                       Sentence_Break
Sc                       Script; NOT 'sc' meaning
                         'General_Category=Currency_Symbol'
Scf                      Simple_Case_Folding
Script                   (Short: sc)
Script_Extensions        (Short: scx)
Scx                      Script_Extensions
SD                       Soft_Dotted
```

```
Sentence_Break          (Short: SB)
Sfc                     Simple_Case_Folding
Simple_Case_Folding     (Short: scf)
Simple_Lowercase_Mapping (Short: slc)
Simple_Titlecase_Mapping (Short: stc)
Simple_Uppercase_Mapping (Short: suc)
Slc                     Simple_Lowercase_Mapping
Soft_Dotted             (Short: SD)
Space                   White_Space
SpacePerl               XPerlSpace.  (Perl extension)
Stc                     Simple_Titlecase_Mapping
STerm
Suc                     Simple_Uppercase_Mapping
Tc                      Titlecase_Mapping
Term                    Terminal_Punctuation
Terminal_Punctuation    (Short: Term)
Title                   Titlecase.  (Perl extension)
Titlecase               (Short: Title).  (Perl extension).  (=
                        \p{Gc=Lt})
Titlecase_Mapping       (Short: tc)
Uc                      Uppercase_Mapping
UIdeo                   Unified_Ideograph
Unicode                 Any.  (Perl extension)
Unicode_1_Name          (Short: na1)
Unified_Ideograph       (Short: UIdeo)
Upper                   Uppercase
Uppercase               (Short: Upper)
Uppercase_Mapping       (Short: uc)
Variation_Selector      (Short: VS)
VertSpace               (Perl extension).  \v
VS                      Variation_Selector
WB                      Word_Break
White_Space             (Short: WSpace)
Word                    (Perl extension).  \w, including beyond
                        ASCII; = \p{Alnum} + \pM + \p{Pc}
Word_Break              (Short: WB)
WSpace                  White_Space
XDigit                  (Perl extension)
XID_Continue            (Short: XIDC)
XID_Start               (Short: XIDS)
XIDC                    XID_Continue
XIDS                    XID_Start
XPerlSpace              (Perl extension).  \s, including beyond
                        ASCII
XPosixAlnum             Alnum.  (Perl extension)
XPosixAlpha             Alpha.  (Perl extension)
XPosixBlank             Blank.  (Perl extension)
XPosixCntrl             General_Category=Cntrl.  (Perl extension)
XPosixDigit             General_Category=Digit.  (Perl extension)
XPosixGraph             Graph.  (Perl extension)
XPosixLower             Lower.  (Perl extension)
XPosixPrint             Print.  (Perl extension)
XPosixPunct             (Perl extension).  \p{Punct} + ASCII-range
                        \p{Symbol}
XPosixSpace             Space.  (Perl extension)
XPosixUpper             Upper.  (Perl extension)
```

```
XPosixWord              Word.  (Perl extension)
XPosixXDigit            XDigit.  (Perl extension)
```

Properties accessible through other means

Certain properties are accessible also via core function calls. These are:

```
Lowercase_Mapping          lc() and lcfirst()
Titlecase_Mapping          ucfirst()
Uppercase_Mapping          uc()
```

Also, Case_Folding is accessible through the `/i` modifier in regular expressions, the `\F` transliteration escape, and the *`fc`* operator.

And, the Name and Name_Aliases properties are accessible through the `\N{}` interpolation in double-quoted strings and regular expressions; and functions `charnames::viacode()`, `charnames::vianame()`, and `charnames::string_vianame()` (which require a `use charnames ();` to be specified.

Finally, most properties related to decomposition are accessible via *Unicode::Normalize*.

Unicode character properties that are NOT accepted by Perl

Perl will generate an error for a few character properties in Unicode when used in a regular expression. The non-Unihan ones are listed below, with the reasons they are not accepted, perhaps with work-arounds. The short names for the properties are listed enclosed in (parentheses). As described after the list, an installation can change the defaults and choose to accept any of these. The list is machine generated based on the choices made for the installation that generated this document.

Expands_On_NFC (XO_NFC)

Expands_On_NFD (XO_NFD)

Expands_On_NFKC (XO_NFKC)

Expands_On_NFKD (XO_NFKD)

> Deprecated by Unicode. These are characters that expand to more than one character in the specified normalization form, but whether they actually take up more bytes or not depends on the encoding being used. For example, a UTF-8 encoded character may expand to a different number of bytes than a UTF-32 encoded character.

Grapheme_Link (Gr_Link)

> Deprecated by Unicode: Duplicates ccc=vr (Canonical_Combining_Class=Virama)

Indic_Matra_Category (InMC)

Indic_Syllabic_Category (InSC)

> Provisional

Jamo_Short_Name (JSN)

Other_Alphabetic (OAlpha)

Other_Default_Ignorable_Code_Point (ODI)

Other_Grapheme_Extend (OGr_Ext)

Other_ID_Continue (OIDC)

Other_ID_Start (OIDS)

Other_Lowercase (OLower)

Other_Math (OMath)

Other_Uppercase (OUpper)

Used by Unicode internally for generating other properties and not intended to be used stand-alone

Script=Katakana_Or_Hiragana (sc=Hrkt)

Obsolete. All code points previously matched by this have been moved to "Script=Common". Consider instead using "Script_Extensions=Katakana" or "Script_Extensions=Hiragana" (or both)

Script_Extensions=Katakana_Or_Hiragana (scx=Hrkt)

All code points that would be matched by this are matched by either "Script_Extensions=Katakana" or "Script_Extensions=Hiragana"

An installation can choose to allow any of these to be matched by downloading the Unicode database from *http://www.unicode.org/Public/* to `$Config{privlib}`*/unicore/* in the Perl source tree, changing the controlling lists contained in the program `$Config{privlib}`*/unicore/mktables* and then re-compiling and installing. (`%Config` is available from the Config module).

Other information in the Unicode data base

The Unicode data base is delivered in two different formats. The XML version is valid for more modern Unicode releases. The other version is a collection of files. The two are intended to give equivalent information. Perl uses the older form; this allows you to recompile Perl to use early Unicode releases.

The only non-character property that Perl currently supports is Named Sequences, in which a sequence of code points is given a name and generally treated as a single entity. (Perl supports these via the `\N{...}` double-quotish construct, *"charnames::string_vianame(name)" in charnames*, and *"namedseq()" in Unicode::UCD*.

Below is a list of the files in the Unicode data base that Perl doesn't currently use, along with very brief descriptions of their purposes. Some of the names of the files have been shortened from those that Unicode uses, in order to allow them to be distinguishable from similarly named files on file systems for which only the first 8 characters of a name are significant.

auxiliary/GraphemeBreakTest.html

auxiliary/LineBreakTest.html

auxiliary/SentenceBreakTest.html

auxiliary/WordBreakTest.html

Documentation of validation tests

auxiliary/LBTest.txt

auxiliary/SBTest.txt

auxiliary/WBTest.txt

BidiCharacterTest.txt

BidiTest.txt

NormTest.txt

Validation Tests

CJKRadicals.txt

Maps the kRSUnicode property values to corresponding code points

EmojiSources.txt

Maps certain Unicode code points to their legacy Japanese cell-phone values

Index.txt

Alphabetical index of Unicode characters

IndicMatraCategory.txt

IndicSyllabicCategory.txt

Provisional; for the analysis and processing of Indic scripts

NamedSqProv.txt

Named sequences proposed for inclusion in a later version of the Unicode Standard; if you need them now, you can append this file to *NamedSequences.txt* and recompile perl

NamesList.html

Describes the format and contents of *NamesList.txt*

NamesList.txt

Annotated list of characters

NormalizationCorrections.txt

Documentation of corrections already incorporated into the Unicode data base

Props.txt

Only in very early releases; is a subset of *PropList.txt* (which is used instead)

ReadMe.txt

Documentation

StandardizedVariants.html

Provides a visual display of the standard variant sequences derived from *StandardizedVariants.txt*.

StandardizedVariants.txt

Certain glyph variations for character display are standardized. This lists the non-Unihan ones; the Unihan ones are also not used by Perl, and are in a separate Unicode data base *http://www.unicode.org/ivd*

USourceData.txt

Documentation of status and cross reference of proposals for encoding by Unicode of Unihan characters

USourceGlyphs.pdf

Pictures of the characters in *USourceData.txt*

SEE ALSO

http://www.unicode.org/reports/tr44/

perlrecharclass

perlunicode