The following collections are from the Basic Multilingual Plane.

NOTE - Use of implementation level 1 restricts the repertoire of some character collections (see clause 23.6). Collections which include combining characters are 7, 8, 9, 11 to 24, 31, 45, 46 and 51.

<table>
<thead>
<tr>
<th>Collection number and name</th>
<th>Positions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 ISO-646 IRV</td>
<td>0020 - 007E</td>
</tr>
<tr>
<td>2 LATIN-1 SUPPLEMENT</td>
<td>00A0 - 00FF</td>
</tr>
<tr>
<td>3 EXTENDED LATIN-A</td>
<td>0100 - 017F</td>
</tr>
<tr>
<td>4 EXTENDED LATIN-B</td>
<td>0180 - 024F</td>
</tr>
<tr>
<td>5 IPA EXTENSIONS</td>
<td>0250 - 02AF</td>
</tr>
<tr>
<td>6 SPACING MODIFIER LETTERS</td>
<td>02B0 - 02FF</td>
</tr>
<tr>
<td>7 COMBINING DIACRITICAL MARKS</td>
<td>0300 - 036F</td>
</tr>
<tr>
<td>8 GREEK</td>
<td>0370 - 03FF</td>
</tr>
<tr>
<td>9 CYRILLIC</td>
<td>0400 - 04FF</td>
</tr>
<tr>
<td>10 ARMENIAN</td>
<td>0530 - 05FF</td>
</tr>
<tr>
<td>11 HEBREW</td>
<td>0590 - 05FF</td>
</tr>
<tr>
<td>12 ARABIC</td>
<td>0600 - 06FF</td>
</tr>
<tr>
<td>13 DEVANAGARI</td>
<td>0900 - 097F</td>
</tr>
<tr>
<td>14 BENGALI</td>
<td>0980 - 09FF</td>
</tr>
<tr>
<td>15 GURMUKHI</td>
<td>0A00 - 0A7F</td>
</tr>
<tr>
<td>16 GUJARATI</td>
<td>0A80 - 0AFF</td>
</tr>
<tr>
<td>17 ORIYA</td>
<td>0B00 - 0B7F</td>
</tr>
<tr>
<td>18 TAMIL</td>
<td>0B80 - 0BFF</td>
</tr>
<tr>
<td>19 TELUGU</td>
<td>0C00 - 0C7F</td>
</tr>
<tr>
<td>20 KANNADA</td>
<td>0C80 - 0CFF</td>
</tr>
<tr>
<td>21 MALAYALAM</td>
<td>0D00 - 0D7F</td>
</tr>
<tr>
<td>22 THAI</td>
<td>0E00 - 0E7F</td>
</tr>
<tr>
<td>23 LAO</td>
<td>0E80 - 0EFF</td>
</tr>
<tr>
<td>24 TIBETAN</td>
<td>1000 - 105F</td>
</tr>
<tr>
<td>25 GEORGIAN</td>
<td>10A0 - 10FF</td>
</tr>
<tr>
<td>26 ADDITIONAL EXTENDED LATIN</td>
<td>1E00 - 1EFF</td>
</tr>
<tr>
<td>27 GREEK EXTENSIONS</td>
<td>1F00 - 1FFF</td>
</tr>
<tr>
<td>28 GENERAL PUNCTUATION</td>
<td>2000 - 206F</td>
</tr>
<tr>
<td>29 SUPERSCRIPTS AND SUBSCRIPTS</td>
<td>2070 - 209F</td>
</tr>
<tr>
<td>30 CURRENCY SYMBOLS</td>
<td>20A0 - 20CF</td>
</tr>
<tr>
<td>31 COMBINING DIACRITICAL MARKS FOR SYMBOLS</td>
<td>20DD - 20FF</td>
</tr>
<tr>
<td>32 LETTERLIKE SYMBOLS</td>
<td>2100 - 214F</td>
</tr>
<tr>
<td>33 NUMBER FORMS</td>
<td>2150 - 218F</td>
</tr>
<tr>
<td>34 ARROWS</td>
<td>2190 - 21FF</td>
</tr>
<tr>
<td>35 MATHEMATICAL OPERATORS</td>
<td>2200 - 22FF</td>
</tr>
<tr>
<td>36 MISCELLANEOUS TECHNICAL</td>
<td>2300 - 23FF</td>
</tr>
<tr>
<td>37 CONTROL PICTURES</td>
<td>2400 - 243F</td>
</tr>
<tr>
<td>38 OPTICAL CHARACTER RECOGNITION</td>
<td>2440 - 245F</td>
</tr>
<tr>
<td>39 ENCLOSED ALPHANUMERIC</td>
<td>2460 - 24FF</td>
</tr>
<tr>
<td>40 BOX DRAWING</td>
<td>2500 - 257F</td>
</tr>
<tr>
<td>41 BLOCK ELEMENTS</td>
<td>2580 - 259F</td>
</tr>
<tr>
<td>42 GEOMETRIC SHAPES</td>
<td>25A0 - 25FF</td>
</tr>
<tr>
<td>43 MISCELLANEOUS DINGBATS</td>
<td>2600 - 26FF</td>
</tr>
<tr>
<td>44 DINGBATS</td>
<td>2700 - 27BF</td>
</tr>
<tr>
<td>45 CJK SYMBOLS AND PUNCTUATION</td>
<td>3000 - 303F</td>
</tr>
<tr>
<td>46 HIRAGANA</td>
<td>3040 - 309F</td>
</tr>
</tbody>
</table>
The following specify collections which are the union of particular collections defined above.

90 GENERAL FORMAT CHARACTERS
   Collections 80 - 83

91 SCRIPT-SPECIFIC FORMAT CHARACTERS
   Collections 84 - 86

The following specify other collections.

92 COMBINING CHARACTERS
   characters specified in ANNEX B

100 BMP
   0000 - FFFD

300 PRIVATE USE PLANES
   G=00, P=E0 - FF

400 PRIVATE USE GROUPS
   G=60 - 7F

The following collections specify characters used for alternate formats and script-specific formats. See annex D for more information.

80 ZERO-WIDTH BOUNDARY INDICATORS
   200B - 200D
   FEFF

81 FORMAT SEPARATORS 2028 - 2029
Annex B (normative)

List of combining characters

The characters in the blocks COMBINING DIA-CRITICAL MARKS (0300 to 036F), COMBINING DIACRITICAL MARKS FOR SYMBOLS (20D0 to 20FF) and COMBINING HANGUL JAMO (31A0 to 31FF) are combining characters. In addition, the following characters are combining characters.

0370 GREEK COMBINING IOTA BELOW
0371 GREEK COMBINING DASIA PNEUMATA
0372 GREEK COMBINING PSIILI PNEUMATA
0384 GREEK COMBINING TONOS
0385 GREEK COMBINING DIAERESIS TONOS
0483 CYRILLIC COMBINING TILIO
0484 CYRILLIC COMBINING PALATALIZATION
0485 CYRILLIC COMBINING DASIA PNEUMATA
0486 CYRILLIC COMBINING PSIILI PNEUMATA
05B0 HEBREW POINT SHEVA
05B1 HEBREW POINT HATAL SEGOL
05B2 HEBREW POINT HATAL PATAH
05B3 HEBREW POINT HATAL QAMATS
05B4 HEBREW POINT HIRIQ
05B5 HEBREW POINT TSERE
05B6 HEBREW POINT SEGOL
05B7 HEBREW POINT PATAH
05B8 HEBREW POINT QAMATS
05B9 HEBREW POINT HOLAM
05BB HEBREW POINT QUBUTS
05BC HEBREW POINT DAGESH
05BD HEBREW POINT METEG
05BF HEBREW POINT RAFE
05C1 HEBREW POINT SHIN DOT
05C2 HEBREW POINT SIN DOT
05F5 HEBREW POINT VAVICA
0648 ARABIC FATHATAN
064C ARABIC DAMMATAN
064D ARABIC KASRATAN
064E ARABIC FATHAH
064F ARABIC Dammah
0650 ARABIC KASRAH
0651 ARABIC SHADDAH
0652 ARABIC SUKUN
0670 ARABIC ALEF ABOVE
06D6 SMALL HIGH LIGATURE SAD WITH LAM WITH ALEF MAKSURA
06D7 SMALL HIGH LIGATURE QAF WITH LAM WITH ALEF MAKSURA
06D8 SMALL HIGH MEEM INITIAL FORM
06D9 SMALL HIGH LAM ALEF
06DA SMALL HIGH JEEM
06DB SMALL HIGH THREE DOTS
06DC SMALL HIGH SEEN
06DD END OF AYAH
06DE START OF RUB EL HIZB
06DF SMALL HIGH ROUNDED ZERO
06E0 SMALL HIGH UPRIGHT RECTANGULAR ZERO
06E1 SMALL HIGH DOTLESS HEAD OF KHAH
06E2 SMALL HIGH MEEM ISOLATED FORM
06E3 SMALL LOW SEEN
06E4 SMALL HIGH MADDAR
06E7 SMALL HIGH YEH
06E8 SMALL HIGH NOON
06EA EMPTY CENTER LOW STOP
06EB EMPTY CENTER HIGH STOP
06EC ROUNDED HIGH STOP WITH FILLED CENTER
06ED SMALL LOW MEEM
06F1 DEVANAGARI SIGN CANDRABINDU
06F2 DEVANAGARI SIGN ANUSVARA
06F3 DEVANAGARI SIGN NUKTA
06F4 DEVANAGARI VOWEL SIGN I
06F5 DEVANAGARI VOWEL SIGN U
06F6 DEVANAGARI VOWEL SIGN UI
06F7 DEVANAGARI VOWEL SIGN VOCALIC R
06F8 DEVANAGARI VOWEL SIGN VOCALIC RR
06F9 DEVANAGARI VOWEL SIGN VOCALIC RA
06FA DEVANAGARI GRAVE ACCENT
06FB DEVANAGARI ACUTE ACCENT
06FC DEVANAGARI VOWEL SIGN VOCALIC L
06FD DEVANAGARI VOWEL SIGN VOCALIC LL
06FE BENGALI SIGN CANDRABINDU
06FF BENGALI SIGN NUKTA
06C1 BENGALI SIGN VAV
06C8 BENGALI VOWEL SIGN I
06C9 BENGALI VOWEL SIGN U
06CC BENGALI VOWEL SIGN O
06CD BENGALI VOWEL SIGN AU
06D0 BENGALI VOWEL SIGN SHORT E
06D1 BENGALI VOWEL SIGN AI
06D2 BENGALI VOWEL SIGN I
06D3 BENGALI VOWEL SIGN U
06D4 BENGALI VOWEL SIGN VOCALIC R
06D5 BENGALI VOWEL SIGN VOCALIC RR
06D6 BENGALI VOWEL SIGN E
06D7 BENGALI VOWEL SIGN A
06D8 BENGALI VOWEL SIGN O
06D9 BENGALI VOWEL SIGN AU
06DA BENGALI VOWEL SIGN SHORT E
06DB BENGALI VOWEL SIGN I
06DC BENGALI VOWEL SIGN U
06DD BENGALI VOWEL SIGN VOCALIC R
06DE BENGALI VOWEL SIGN VOCALIC RR
06DF BENGALI VOWEL SIGN E
06E0 BENGALI VOWEL SIGN AI
06E1 BENGALI VOWEL SIGN I
06E2 BENGALI VOWEL SIGN U
06E3 BENGALI VOWEL SIGN VOCALIC R
06E4 BENGALI VOWEL SIGN VOCALIC RR
06E5 BENGALI VOWEL SIGN E
06E6 BENGALI VOWEL SIGN AI
06E7 BENGALI VOWEL SIGN I
06E8 BENGALI VOWEL SIGN U
06E9 BENGALI VOWEL SIGN VOCALIC R
06EA BENGALI VOWEL SIGN VOCALIC RR
06EB BENGALI VOWEL SIGN E
06EC BENGALI VOWEL SIGN AI
06ED BENGALI VOWEL SIGN I
06EE BENGALI VOWEL SIGN U
06EF BENGALI VOWEL SIGN VOCALIC R
06F0 BENGALI VOWEL SIGN VOCALIC RR
06F1 BENGALI VOWEL SIGN E
06F2 BENGALI VOWEL SIGN AI
06F3 BENGALI VOWEL SIGN I
06F4 BENGALI VOWEL SIGN U
06F5 BENGALI VOWEL SIGN VOCALIC R
06F6 BENGALI VOWEL SIGN VOCALIC RR
06F7 BENGALI VOWEL SIGN E
06F8 BENGALI VOWEL SIGN AI
06F9 BENGALI VOWEL SIGN I
06FA BENGALI VOWEL SIGN U
06FB BENGALI VOWEL SIGN VOCALIC R
06FC BENGALI VOWEL SIGN VOCALIC RR
06FD BENGALI VOWEL SIGN E
06FE BENGALI VOWEL SIGN AI
06FF BENGALI VOWEL SIGN I

0901 BENGALI SIGN CANDRABINDU
0902 BENGALI SIGN ANUSVARA
0903 BENGALI SIGN NUKTA
0904 BENGALI VOWEL SIGN I
0905 BENGALI VOWEL SIGN U
0906 BENGALI VOWEL SIGN UI
0907 BENGALI VOWEL SIGN VOCALIC R
0908 BENGALI VOWEL SIGN VOCALIC RR
0909 BENGALI VOWEL SIGN VOCALIC RA
090A BENGALI VOWEL SIGN E
090B BENGALI VOWEL SIGN AI
090C BENGALI VOWEL SIGN I
090D BENGALI VOWEL SIGN U
090E BENGALI VOWEL SIGN VOCALIC R
090F BENGALI VOWEL SIGN VOCALIC RR
0910 BENGALI VOWEL SIGN E
0911 BENGALI VOWEL SIGN AI
0912 BENGALI VOWEL SIGN I
0913 BENGALI VOWEL SIGN U
0914 BENGALI VOWEL SIGN VOCALIC R
0915 BENGALI VOWEL SIGN VOCALIC RR
0916 BENGALI VOWEL SIGN E
0917 BENGALI VOWEL SIGN AI
0918 BENGALI VOWEL SIGN I
0919 BENGALI VOWEL SIGN U
091A BENGALI VOWEL SIGN VOCALIC R
091B BENGALI VOWEL SIGN VOCALIC RR
091C BENGALI VOWEL SIGN E
091D BENGALI VOWEL SIGN AI
091E BENGALI VOWEL SIGN I
091F BENGALI VOWEL SIGN U
0920 BENGALI VOWEL SIGN VOCALIC R
0921 BENGALI VOWEL SIGN VOCALIC RR
0922 BENGALI VOWEL SIGN E
0923 BENGALI VOWEL SIGN AI
0924 BENGALI VOWEL SIGN I
0925 BENGALI VOWEL SIGN U
0926 BENGALI VOWEL SIGN VOCALIC R
0927 BENGALI VOWEL SIGN VOCALIC RR
0928 BENGALI VOWEL SIGN E
0929 BENGALI VOWEL SIGN AI
092A BENGALI VOWEL SIGN I
092B BENGALI VOWEL SIGN U
092C BENGALI VOWEL SIGN VOCALIC R
092D BENGALI VOWEL SIGN VOCALIC RR
092E BENGALI VOWEL SIGN E
092F BENGALI VOWEL SIGN AI
0930 BENGALI VOWEL SIGN I
0931 BENGALI VOWEL SIGN U
0932 BENGALI VOWEL SIGN VOCALIC R
0933 BENGALI VOWEL SIGN VOCALIC RR
0934 BENGALI VOWEL SIGN E
0935 BENGALI VOWEL SIGN AI
0936 BENGALI VOWEL SIGN I
0937 BENGALI VOWEL SIGN U
0938 BENGALI VOWEL SIGN VOCALIC R
0939 BENGALI VOWEL SIGN VOCALIC RR
093A BENGALI VOWEL SIGN E
093B BENGALI VOWEL SIGN AI
093C BENGALI VOWEL SIGN I
093D BENGALI VOWEL SIGN U
093E BENGALI VOWEL SIGN VOCALIC R
093F BENGALI VOWEL SIGN VOCALIC RR

302B IDEOGRAPHIC RISING TONE MARK
302C IDEOGRAPHIC DEPARTING TONE MARK
302D IDEOGRAPHIC ENTERING TONE MARK
302E HANGUL SINGLE DOT TONE MARK
302F HANGUL DOUBLE DOT TONE MARK
3099 COMBINING KATAKANA-HIRAGANA VOICED SOUND MARK
309A COMBINING KATAKANA-HIRAGANA SEMI-VOICED SOUND MARK
Annex C
(informative)

Mirrored characters in bi-directional context

In the context of right-to-left (bi-directional) text, the following characters have semantic meaning. To preserve the meaning in right-to-left text, the character symbol should be rendered as the mirror image of the associated graphical symbol from the left-to-right context. These characters include mathematical symbols and paired characters such as the SQUARE BRACKETS. For example, in a right-to-left text segment, the GREATER-T  

LESS-THERE

SIGMA

SIGMA

TILDE OPERATOR

REVERSED TILDE

INVERTED LAZY S

SINE WAVE

WREATH PRODUCT

NOT TILDE

MINUS TILDE

ASYMPTOTICALLY EQUAL TO

NOT ASYMPTOTICALLY EQUAL TO

APPROXIMATELY EQUAL TO

APPROXIMATELY BUT NOT ACTUALLY EQUAL TO

NEITHER APPROXIMATELY NOR ACTUALLY EQUAL TO

ALMOST EQUAL TO

NOT ALMOST EQUAL TO

ALMOST OR EQUAL TO

TRIPLE TILDE

ALL EQUAL TO

APPROXIMATELY EQUAL TO OR THE IMAGE OF

IMAGE OF OR APPROXIMATELY EQUAL TO

COLON EQUAL

EQUAL COLON

QUESTIONED EQUAL TO

NOT EQUAL TO

NOT IDENTICAL TO

LESS-THERE OR EQUAL TO

GREATER-THERE OR EQUAL TO

LESS-THERE OVER EQUAL TO

GREATER-THERE OVER EQUAL TO

LESS-THERE BUT NOT EQUAL TO

GREATER-THERE BUT NOT EQUAL TO

MUCH LESS-THERE

MUCH GREATER-THERE

NOT LESS-THERE

NOT GREATER-THERE

NEITHER LESS-THERE NOR EQUAL TO

NEITHER GREATER-THERE NOR EQUAL TO

LESS-THERE OR EQUIVALENT TO

GREATER-THERE OR EQUIVALENT TO

NEITHER LESS-THERE NOR EQUIVALENT TO

NEITHER GREATER-THERE NOR EQUIVALENT TO

LESS-THERE OR GREATER-THERE

GREATER-THERE OR LESS-THERE

NEITHER LESS-THERE NOR GREATER-THERE

NEITHER GREATER-THERE NOR LESS-THERE

PRECEDES

SUCCEDES

PRECEDES OR EQUAL TO

SUCCEDES OR EQUAL TO

PRECEDES OR EQUIVALENT TO

SUCCEDES OR EQUIVALENT TO

DOES NOT PRECEDE
Annex D
(informative)

Alternate Format Characters

There is a special class of characters called Alternate Format Characters which are included for compatibility with some industry practices. These characters do not have printable graphic symbols, and are thus represented in the character code tables by dotted boxes.

The function of most of these characters is to indicate the correct presentation of a sequence of characters. For any text processing other than presentation (such as sorting and searching), the alternate format characters can be ignored by filtering them out. The alternate format characters are not intended to be used in conjunction with bi-directional control functions from ISO/IEC 6429.

There are collections of graphic characters for selected subsets which consists of Alternate Format Characters (see Annex A).

D.1 General format characters

D.1.1 Zero-width boundary indicators

The following characters are used to indicate whether or not the adjacent characters should be separated by a word boundary.

ZERO WIDTH SPACE (200B): This character behaves like a SPACE in that it indicates a word boundary, except that it has no width. For example, this character could be used to indicate word boundaries in Thai, which does not use visible spaces to separate words.

ZERO WIDTH NO-BREAK SPACE (FEFF): This character behaves like a NO-BREAK SPACE in that it indicates the absence of word boundaries, except that it has no width. For example, this character could be inserted after the fourth character in the text "base+delta" to indicate that there is to be no word break between the "e" and the "+".

NOTE - For additional usages of this character for "signature", see annex E.

The following characters are used to indicate whether or not the adjacent characters should be joined together in rendering (cursive joiners).

ZERO WIDTH NON-JOINER (200C): This character indicates that the adjacent characters should not be joined together in a ligature or cursive connection, even when normal rendering would join the characters. For examples:

1. ZERO WIDTH NON-JOINER between LATIN SMALL LETTER F and LATIN SMALL LETTER I indicates that they should not join to form a ligature "fi" in rendering.

2. ZERO WIDTH NON-JOINER between ARABIC LETTER NOON and ARABIC LETTER MEEM indicates that the characters should not be the normal cursive connection form being rendered.

3. In the sequence DEVANAGARI KA followed by DEVANAGARI SIGN VIRAMA followed by DEVANAGARI KA, ZERO WIDTH NON-JOINER between the second and third characters indicates that the characters should not form the conjunct "ka".

ZERO WIDTH JOINER (200D): This character indicates that the adjacent characters should be represented with joining forms, even if the normal rendering would not join the characters. However, the joiner should not be used to cause a spacing character to be rendered as a combining mark. For examples:

1. In the sequence SPACE followed by ARABIC LETTER BAA, ZERO WIDTH JOINER can be inserted between the two characters to display the initial form of the ARABIC LETTER BAA.

2. ZERO WIDTH JOINER inserted between LATIN SMALL LETTER S and LATIN SMALL LETTER T can be used to indicate that the "st" ligature be rendered if possible, even if
the ligature would not normally be rendered.

D.1.2 Format separators

The following characters are used to indicate formatting boundaries between lines or paragraphs.

**LINE SEPARATOR** (2028): This character indicates where a new line should start; although the text should continue to the next line, it does not start a new paragraph; e.g. no inter-paragraph indentation might be applied.

**PARAGRAPH SEPARATOR** (2029): This character indicates where a new paragraph should start; e.g. the text should continue on the next line and inter-paragraph line spacing or paragraph indentation might be applied.

D.1.3 Bi-directional text formatting

The following characters are used in formatting bi-directional text. If the specification of a subset includes these characters, then text containing right-to-left characters are to be rendered with an implicit bi-directional algorithm.

An implicit algorithm uses the directional character properties to determine the correct display order of characters on a horizontal line of text.

The following characters are format characters that act exactly like right-to-left or left-to-right characters in terms of affecting ordering (Bi-directional format marks). They have no visible graphic symbols, and they do not have any other semantic effect.

Their use can be more convenient than the explicit embeddings or overrides, since their scope is much more local.

**LEFT-TO-RIGHT MARK** (200E): In bi-directional formatting, this character acts like a left-to-right character (such as LATIN SMALL LETTER A).

**RIGHT-TO-LEFT MARK** (200F): In bi-directional formatting, this character acts like a right-to-left character (such as ARABIC LETTER NOON).

English quotation in the middle of an Arabic sentence can be marked as being an embedded left-to-right string. These format characters nest in blocks, with the embedding and override characters initiating (pushing) a block, and the pop character terminating (popping) a block.

The function of the embedding and override characters are very similar; the main difference is that the embedding characters specify the implicit direction of the text, while the override characters specify the explicit direction of the text. When text has an explicit direction, the normal directional character properties are ignored, and all of the text is assumed to have the ordering direction determined by the override character.

**LEFT-TO-RIGHT EMBEDDING** (202A): This character is used to indicate the start of a left-to-right implicit embedding.

**RIGHT-TO-LEFT EMBEDDING** (202B): This character is used to indicate the start of a right-to-left implicit embedding.

**LEFT-TO-RIGHT OVERRIDE** (202D): This character is used to indicate the start of a left-to-right explicit embedding.

**RIGHT-TO-LEFT OVERRIDE** (202E): This character is used to indicate the start of a right-to-left explicit embedding.

**POP DIRECTIONAL FORMATTING** (202C): This character is used to indicate the termination of an implicit or explicit directional embedding initiated by the above characters.

D.2 Script-specific format characters

D.2.1 Hangul fill characters

The following format characters have a special usage for Hangul characters.

**HANGUL FILL** (3164): This character represents the fill value used with the standard spacing Jamos.

**HALFWIDTH HANGUL FILL** (FFA): As with the other halfwidth characters, this character is included for compatibility with certain systems that provide halfwidth forms of characters.

D.2.2 Character shaping selectors

The following characters are used in conjunction
with Arabic presentation forms. During the presentation process, certain characters may be joined together in cursive connection or ligatures. The following characters indicate that the character shape determination process (CSD) used to achieve this presentation effect is to be either activated or inhibited. Unlike the directional embedding characters, the following characters do not nest.

**INHIBIT ARABIC FORM SHAPING** (206C): Between this character and the following ACTIVATE ARABIC FORM SHAPING format character (if any) the CSD process is to be inhibited. The stored Arabic presentation forms will be presented without shape modification. This is the default state.

**ACTIVATE ARABIC FORM SHAPING** (206D): Between this character and the following INHIBIT ARABIC FORM SHAPING format character (if any), the stored Arabic presentation forms should be presented with shape modification by means of the CSD process.

*NOTE:* These characters have no effect on characters that are not presentation forms; in particular, Arabic nominal characters as from 0600 to 06FF are always subject to character shaping, and are unaffected by these formatting characters.

### D.2.3 Numeric shape selectors

The following characters allow the selection of the shapes in which the digits from 0030 to 0039 are to be rendered. Unlike the directional embedding characters, these do not nest.

**NATIONAL DIGIT SHAPES** (206E): Between this character and the following NOMINAL DIGIT SHAPES format character (if any), digits from 0030 to 0039 are rendered with the appropriate national digit shapes as specified by means of appropriate agreements. For example, they could be displayed with shapes such as the ARABIC-INDIC digits from 0660 to 0669.

**NOMINAL DIGIT SHAPES** (206F): Between this character and the following NATIONAL DIGIT SHAPES format character (if any), the digits from 0030 to 0039 will be rendered with the shapes as those shown in the code tables for those digits. This is the default state.
Annex E  
(informative)

The use of "signatures" to identify UCS

This annex describes a convention for the identification of features of the UCS, by the use of "signatures" within data streams of coded characters. The convention makes use of the character ZERO WIDTH NO-BREAK SPACE, and is applied by a certain class of applications.

When this convention is used, a signature at the beginning of a stream of coded characters indicates that the characters following are encoded in the UCS-2 or UCS-4 coded representation, and indicates the ordering of the octets within the coded representation of each character (see 6.3). It is typical of the class of applications mentioned above, that some make use of the signatures when receiving data, while others do not. The signatures are therefore designed in a way that makes it easy to ignore them.

In this convention, the ZERO WIDTH NO-BREAK SPACE character has the following significance when it is present at the beginning of a stream of coded characters:

UCS-2 signature: FEFF  
UCS-4 signature: 0000 FEFF

An application receiving data may either use these signatures to identify the coded representation form, or may ignore them and treat FEFF as the ZERO WIDTH NO-BREAK SPACE character.

If an application which uses one of these signatures recognises its coded representation in reverse sequence (e.g. hexadecimal FFFE), the application can identify that the coded representations of the following characters use the opposite octet sequence to the sequence expected, and may take the necessary action to recognise the characters correctly.

NOTE - The hexadecimal value FFFE does not correspond to any coded character within this International Standard.
The following method transforms the coded representation of graphic characters in this coded character set into a form that does not use octet values specified in ISO 2022 as coded representations of C0, SPACE, DEL, or C1 characters, and can thus be used for transmitting text data through communication systems that are sensitive to these octet values.

F.1 Outline of the algorithm

The algorithm can be summarized as follows:

1. A UCS character from 0000 0000 to 0000 009F is mapped to the corresponding octet from 00 to 9F.

2. A UCS character from 0000 00A0 to 0000 00FF is mapped to a sequence of two octets, with the first octet being A0, and the second octet in the range A0 to FF.

3. A UCS character from 0000 0100 to 0000 401F is mapped to a sequence of two octets, with the first octet in the range from A1 to F5, and the second octet having 190 values in the range 21 to 7E or the range A0 to FF.

4. A UCS character from 0000 4016 to 0003 BE2E is mapped to a sequence of three octets, with the first octet in the range from F6 to FB, and the other octets in the range 21 to 7E or the range A0 to FF.

5. A UCS character at 0003 BE2E or larger is mapped to a sequence of five octets, with the first octet in the range from FC to FF, and the other octets in the range 21 to 7E or the range A0 to FF.

Notice that four-octet sequences are not used, since this maximizes the number of characters that can use the three-octet form.

F.2 Notation

1) All numbers are in hexadecimal notation.

2) Octet boundaries in the transformed text are indicated with semicolons.

3) The symbol "%" indicates the modulo operation, e.g.:
   \[ x \% y = x \text{ modulo } y \]

The symbol "\(^\)" indicates the integer division operation, e.g.:
   \[ 7 \div 3 = 2 \]

Superscripting indicates the power-of-operation, e.g.:
   \[ 2^3 = 8 \]

Precedence is \(^3\) > \(^\) > ", e.g.:
   \[ x \div y^2 \% w = (\frac{x}{y^2}) \% w \]

4) \( T(z) \) is defined for \( z = 00..FF \) such that:

   \[
   \begin{align*}
   z &= 00 .. 0D: & T(z) &= z + 21 \\
   z &= 0E .. BD: & T(z) &= z + 42 \\
   z &= BE .. FF: & T(z) &= z - BE \\
   z &= DF .. FF: & T(z) &= z - 60 \\
   \end{align*}
   \]

   e.g. \( T(00) = 21, T(0D) = 7E, T(5E) = A0, T(BD) = FF, T(BE) = 00, T(DE) = 20, T(DF) = 7F, T(FF) = 9F \)

5) \( U(z) \) is the inverse of \( T(z) \): that is, \( U(T(z)) = z \), and \( T(U(z)) = z \):

   \[
   \begin{align*}
   z &= 00 .. 20: & U(z) &= z + BE \\
   z &= 21 .. 7E: & U(z) &= z - 21 \\
   z &= 7F .. 9F: & U(z) &= z + 60 \\
   z &= A0 .. FF: & U(z) &= z - 42 \\
   \end{align*}
   \]

   e.g. \( U(00) = BE, U(20) = DE, U(21) = 00, U(7E) = 5D, U(7F) = DF, U(9F) = FF, U(A0) = 5E, U(FF) = BD \)

6) The algorithm in this annex has been presented in a descriptive format. The implementation may differ for efficiency. For example, the \( T \) and \( U \) functions can be implemented with a small table look-up.
### F.3 From UCS to UTF format

<table>
<thead>
<tr>
<th>Condition/UCS</th>
<th>UTF octets</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x = 0000 0000 .. 0000 009F:$</td>
<td>$x;$</td>
</tr>
<tr>
<td>$x = 0000 00A0 .. 0000 00FF:$</td>
<td>$A0;$</td>
</tr>
<tr>
<td>$x = 0000 0100 .. 0000 4015:$</td>
<td>$A1 + y/BE; T(y%BE);$</td>
</tr>
<tr>
<td>$(y = x - 0000 0100)$</td>
<td></td>
</tr>
<tr>
<td>$x = 0000 4016 .. 0003 8E2D:$</td>
<td>$F6 + y/BE^2; T(y/BE%BE); T(y%BE);$</td>
</tr>
<tr>
<td>$(y = x - 0000 4016)$</td>
<td></td>
</tr>
<tr>
<td>$x = 0003 8E2E .. 7FFF FFFF:$</td>
<td>$FC + y/BE^4; T(y/BE^3%BE); T(y/BE^2%BE); T(y/BE%BE); T(y%BE);$</td>
</tr>
<tr>
<td>$(y = x - 0003 8E2E)$</td>
<td></td>
</tr>
</tbody>
</table>

### F.4 From UTF to UCS format

<table>
<thead>
<tr>
<th>Condition/UTF</th>
<th>UCS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x = 00 .. 9F;$</td>
<td>$x$</td>
</tr>
<tr>
<td>$x = A0;$</td>
<td>$y$</td>
</tr>
<tr>
<td>$y;$</td>
<td>$$(x-A1) \times BE + U(y) + 100$$</td>
</tr>
<tr>
<td>$x = A1 .. F5;$</td>
<td>$$(x-F6) \times BE^2 + U(y) \times BE + U(z) + 4016$$</td>
</tr>
<tr>
<td>$y;$</td>
<td></td>
</tr>
<tr>
<td>$z;$</td>
<td></td>
</tr>
<tr>
<td>$w;$</td>
<td>$$+(x-FC) \times BE^4 + U(y) \times BE^3 + U(z) \times BE^2 + U(w) + 38E2E$$</td>
</tr>
</tbody>
</table>
Annex G
(informative)

Recommendation for combined receiving/originating devices with internal storage

This annex is applicable to a widely-used class of devices that can store received CC-data elements for subsequent retransmission.

This recommendation is intended to ensure that loss of information is minimised between the receipt of a CC-data-element and its retransmission.

A device of this class includes a receiving device component and an originating device component as in clause 2.3, and can also store received CC-data-elements for retransmission, with or without modification by the actions of the user on the corresponding characters represented within it.

It is therefore recommended that the originating device ensure that any such characters that may be outside the identified subset of the receiving device component be capable of being retransmitted without change to their coded representations, unless modified by the user. Such a device is a "receiving device with full retransmission capability".

If the originating device component can transmit only the coded representations of the characters of the subset adopted by the receiving device component, such a type of device is a "receiving device with subset retransmission capability".
Annex H
(informative)

Notations of octet value representations

Representation of octet values in this International Standard is different from other character coding standards such as ISO 2022, ISO/IEC 6429 and ISO 8859. This section clarifies the relationship between the two notations.

- In this International Standard, the notation used to express an octet value is \( z \), where \( z \) is a hexadecimal number in the range 00 to FF.

  For example, the character ESCAPE (ESC) of ISO 2022 is represented by 1B.

- In other character coding standards, the notation used to express an octet value is \( x/y \), where \( x \) and \( y \) are two numbers in the range 00 to 15. The correspondence between the notations of the form \( x/y \) and the octet value is as follows.

  \( x \) is the number represented by bit 8, bit 7, bit 6 and bit 5 where these bits are given the weight 8, 4, 2 and 1 respectively;

  \( y \) is the number represented by bit 4, bit 3, bit 2 and bit 1 where these bits are given the weight 8, 4, 2 and 1 respectively.

  For example, the character ESC of ISO 2022 is represented by 01/11.

Thus ISO 2022 (and other character coding standards) octet value notation can be converted to ISO/IEC 10646 octet value notation by converting the value of \( x \) and \( y \) to hexadecimal notation. For example, 04/15 is equivalent with 4F.
Annex J  
(informative)

Character naming guidelines

Guidelines for generating and presenting unique names of characters in ISO/IEC JTC1/SC2 standards is listed in this annex for reference. These guidelines are used in information technology coded character set standards such as ISO 646, ISO 6937, ISO 8859, ISO/IEC 10367 as well as in this International Standard.

These Guidelines specify rules for generating and presenting unique names of characters. Rules 1 to 3 are implemented without exceptions. However, it must be accepted that in some cases (e.g., historical or traditional usage, unforeseen special cases, difficulties inherent to the nature of the character considered), exceptions to some of the other rules will have to be tolerated. Nonetheless, these rules are applied wherever possible.

Rule 1

By convention, only Latin capital letters A to Z, space, and hyphen shall be used for writing the names of characters.

NOTE: Names of ideographic characters may also include digits 0 to 9 provided that a digit is not the first character in a word.

Rule 2

The names of control functions shall be coupled with an acronym consisting of Latin capital letters A to Z and, where required, digits. Once the name has been specified for the first time, the acronym may be used in the remainder of the text where required for simplification and clarity of the text.

Exceptionally, acronyms may be used for graphic characters where usage already exists and clarity requires it, in particular in code tables.

Examples:

Name: LOCKING-SHIFT TWO RIGHT
Acronym: LS2R

Name: SOFT-HYPHEN

Acronym: SHY

NOTE: In ISO/IEC 6429, also the names of the modes have been presented in the same way as control functions.

Rule 3

In some cases, the names of a character can be followed by an additional explanatory statement not part of the name. These statements shall be in parentheses and not in capital Latin letters except the initials of the word where required. See examples in rule 12.

Rule 4

The names of a character shall wherever possible denote its customary meaning, for examples PLUS SIGN. Where this is not possible, names should describe shapes, not usage; for example: UPWARDS ARROW.

Rule 5

Only one name will be given to each character.

Rule 6

The names shall be constructed from an appropriate set of the applicable terms of the following grid and ordered in the sequence of this grid. Exceptions are specified in Rule 11.

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Script</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Case</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Type</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Language</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Attribute</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Designation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Mark(s)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Qualifier</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Examples of such terms:

- Script: Latin, Cyrillic, Arabic
- Case: capital, small
- Type: letter, ligature, digit
- Language: Ukrainian
- Attribute: final, sharp, subscript, vulgar
- Designation: customary name, name of letter
Mark: acute, ogonek, ring above, diaeresis
Qualifier: sign, symbol

Examples of names:
LATIN CAPITAL LETTER A WITH ACUTE
1 2 3 6 7
DIGIT FIVE
3 6
LEFT CURLY BRACKET
5 5 6

NOTE - A ligature is a graphic symbol in which two or more other graphic symbols are imaged as single graphic symbol.

Rule 7
The letters of the Latin script shall be represented within their name by their basic graphic symbols (A, B, C, ...). The letters of all other scripts shall be represented by their transcription in the language of the first published International Standard.

Examples:
K LATIN CAPITAL LETTER K
Ю CYRILLIC CAPITAL LETTER YU

Rule 8
In principle when a character of a given script is used in more than one language, no language name is specified. Exceptions are tolerated where an ambiguity would otherwise result.

Examples:
И CYRILLIC CAPITAL LETTER I
І CYRILLIC CAPITAL LETTER BYELORUSSIANUKRAINIAN I

Rule 9
Letters that are elements of more than one script are considered different even if their shape is the same, they shall have different names.

Examples:
A LATIN CAPITAL LETTER A

Rule 10
A character of one script used in isolation in another script, for example as a graphic symbol in relation with physical units of dimension, is considered as a character different from the character of its native script.

Example:
µ MICRO SIGN

Rule 11
A number of characters have a traditional name consisting of one or two words. It is not intended to change this usage.

Examples:
' APOSTROPHE
: COLON
@ COMMERCIAL AT
= LOW LINE
~ TILDE

Rule 12
In some cases, characters of a given script, often punctuation marks, are used in another script for a different usage. In these cases the customary name reflecting the most general use will be given to the character. The customary name may be followed in the list of characters of a particular standard by the name in parentheses which this character has in the script specified by this particular standard.

Examples:
HORIZONTAL BAR (Parenthetiki pavla)
DIAERESIS (Dialetika)

Rule 13
The above rules shall not apply to ideographic characters. These characters will be identified by alpha-numeric identifiers specified for each ideographic character (see clause 25).
Annex K  
(informative)

Sources of characters

Several sources and contributions were used for constructing this coded character set. In particular, all characters of the following national and international standards are included in this International Standard.


ISO 2047:1975 Information processing — Graphical representations for the control characters of the 7-bit coded character set.


ISO 2033:1983 Information processing — Coding of machine-readable characters (OCR and MICR).


ISO 5428:1984 Greek alphabet coded character set for bibliographic information interchange.


ISO DIS 6861.2:1990 Information and documentation — Glagolitic character set for bibliographic information interchange.


ISO DIS 6937:1991 Information processing — Coded character sets for text communication.

ISO 8859 Information processing — 8-bit single-byte coded graphic character sets. Contents:


ISO 9036:1983 Information processing — Arabic 7-bit coded character set for information interchange.


ISO International register of character sets to be used with escape sequences. (registration procedure ISO 2375:1985).


ISO/IEC DIS 10646-1.2 : 1992 (E)


ASMO 449-1982 Arab Organization for Standardization and Methodology. Data processing — 7-bit coded character set for information interchange.


LTD 37(1610)-1988 Indian standard code for information interchange.


KS C 5601-1987 Korea Industrial Standards Association. Jeongho gyohwanyong buho (Hangul mit Hanja) (Code for Information Interchange (Hangul and Hanja)).


International Phonetic Association. The IPA 1989
Annex L
(informative)

Scripts under consideration for future editions of this International Standard

In order to make sure that this International Standard is useful for people using their native scripts, characters included in this International Standard were selected with input and feedback from national standards organisations and/or qualified experts.

Some scripts and symbols were not included in this edition because sufficient input and feedback have not been provided during the preparation and review stages.

It is intended that character code positions for these scripts and symbols will be allocated when sufficient input and review is provided. Such scripts and symbols includes:

- Burmese
- Cree and Inuktitut
- Ethiopian
- Hieroglyphics
- Khmer
- Maldive
- Mongolian
- Runic
- Sinhalese
- Syriac
- Yi

This list is not exhaustive. Other scripts and symbols as well as additional characters for the included scripts are expected to be included in future editions of this International Standard.