INTERNATIONAL STANDARD

ISO/IEC 10646

Sixth edition
2020-12

Information technology — Universal coded character set (UCS)

Technologies de l'information — Jeu universel de caractères codés (JUC)

Reference number
ISO/IEC 10646:2020(E)

Documents
Document Sharing Hub
© ISO/IEC 2020

Full document details: ISO IEC 10646-2020



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2020

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO IEC 10646-2020

CONTENTS

1 Scope 1
2 Normative references 1
3 Terms and definitions 2
4 Conformance 8
4.1 General 8
4.2 Conformance of information interchange 8
4.3 Conformance of devices 8
5 Electronic data attachments 9
6 General structure of the UCS 10
7 Basic structure and nomenclature 11
7.1 Structure 11
7.2 Coding of characters 12
7.3 Types of code points 12
7.4 Naming of characters 13
7.5 Short identifiers for code points (UIDs) 14
7.6 UCS Sequence Identifiers 14
7.7 Octet sequence identifiers 15
8 Revision and updating of the UCS 15
9 Subsets 15
9.1 General 15
9.2 Limited subset 15
9.3 Selected subset 15
10 UCS encoding forms 15
10.1 General 15
10.2 UTF-8 15
10.3 UTF-16 16
10.4 UTF-32 17
11 UCS encoding schemes 17
11.1 General 17
11.2 UTF-8 17
11.3 UTF-16BE 17
11.4 UTF-16LE 18
11.5 UTF-16 18
11.6 UTF-32BE 18
11.7 UTF-32LE 18
11.8 UTF-32 19
12 Use of control functions with the UCS 19
13 Declaration of identification of features 20
13.1 Purpose and context of identification 20
13.2 Identification of a UCS encoding scheme 20
13.3 Identification of subsets of graphic characters 21

© ISO/IEC 2020 – All rights reserved

AVA Documents
Document Sharing Hub

Full document details: ISO IEC 10646-2020

13.4 Identification of control function set ... 21
13.5 Identification of the coding system of ISO/IEC 2022 ... 21
14 Structure of the code charts and lists ... 22
15 Block and collection names ... 22
15.1 Block names ... 22
15.2 Collection names ... 23
16 Mirrored characters in bidirectional context ... 23
16.1 Mirrored characters ... 23
16.2 Directionality of bidirectional text ... 23
17 Special characters ... 23
17.1 General ... 23
17.2 Space characters ... 23
17.3 Currency symbols ... 24
17.4 Format characters ... 24
17.5 Ideographic description characters ... 24
17.6 Variation selectors and variation sequences ... 25
18 Presentation forms of characters ... 27
19 Compatibility characters ... 27
20 Order of characters ... 27
21 Combining characters ... 28
21.1 Order of combining characters ... 28
21.2 Combining class and canonical ordering ... 28
21.3 Appearance in code charts ... 28
21.4 Alternate coded representations ... 28
21.5 Multiple combining characters ... 28
21.6 Collections containing combining characters ... 29
21.7 Combining Grapheme Joiner ... 29
22 Normalization forms ... 29
23 Special features of individual scripts and symbol repertoires ... 30
23.1 Hangul syllable composition method ... 30
23.2 Features of scripts used in India and some other South Asian countries ... 30
23.3 Byzantine musical symbols ... 31
23.4 Source references for pictographic symbols ... 31
24 Source references for CJK ideographs ... 32
24.1 List of source references ... 32
24.2 Source references file for CJK ideographs ... 35
24.3 Source reference presentation for CJK Unified ideographs ... 37
24.4 Source references presentation for CJK Compatibility ideographs ... 40
25 Source references for Tangut ideographs ... 40
25.1 List of source references ... 40
25.2 Source reference file for Tangut ideographs ... 41
25.3 Source reference presentation for Tanguts ideographs ... 42
26 Source references for Nüshu characters ... 42

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 10646-2020

26.1 List of source references ... 42
26.2 Source reference file for Nüshu characters ... 42
27 Character names and annotations ... 43
27.1 Entity names ... 43
27.2 Name formation ... 43
27.3 Single name ... 44
27.4 Name immutability ... 44
27.5 Name uniqueness ... 44
27.6 Character names for CJK ideographs ... 45
27.7 Character names for Tangut ideographs ... 45
27.8 Character names for Nüshu characters ... 45
27.9 Character names for Khitan Small Script characters ... 46
27.10 Character names for Hangul syllables ... 46
28 Named UCS Sequence Identifiers ... 47
29 Structure of the Basic Multilingual Plane ... 49
30 Structure of the Supplementary Multilingual Plane for scripts and symbols (SMP) ... 51
31 Structure of the Supplementary Ideographic Plane (SIP) ... 54
32 Structure of the Tertiary Ideographic Plane (TIP) ... 55
33 Structure of the Supplementary Special-purpose Plane (SSP) ... 56
34 Code charts and lists of character names ... 57
34.1 General ... 57
34.2 Code chart ... 57
34.3 Character names list ... 57
34.4 Summary of standardized variation sequences ... 58
34.5 Code charts and lists of character names ... 58
Annex A (normative) Collections of graphic characters for subsets ... 2743
A.1 Collections of coded graphic characters ... 2743
A.2 Blocks lists ... 2750
A.3 Fixed collections of the whole UCS (except Unicode collections) ... 2753
A.4 CJK collections ... 2756
A.5 Other collections ... 2757
A.6 Unicode collections ... 2761
Annex B (normative) List of combining characters ... 2763
Annex C (normative) Transformation format for planes 01 to 10 of the UCS (UTF-16) ... 2764
Annex D (normative) UCS Transformation Format 8 (UTF-8) ... 2765
Annex E (normative) Mirrored characters in bidirectional context ... 2766
Annex F (informative) Format characters ... 2767
F.1 General format characters ... 2767
F.2 Script-specific format characters ... 2769
F.3 Interlinear annotation characters ... 2770
F.4 Subtending format characters ... 2770
F.5 Shorthand format characters ... 2771
F.6 Invisible mathematical operators ... 2771

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 10646-2020

F.7 Western musical symbols ... 2771
F.8 Language tagging using Tag characters ... 2772
Annex G (informative) Alphabetically sorted list of character names ... 2774
Annex H (informative) The use of "signatures" to identify UCS ... 2775
Annex I (informative) Ideographic description characters ... 2776
I.1 General ... 2776
I.2 Syntax of an ideographic description sequence ... 2776
I.3 Individual definitions of the ideographic description characters ... 2777
Annex J (informative) Recommendation for combined receiving/originating devices with internal storage ... 2779
Annex K (informative) Notations of octet value representations ... 2780
Annex L (informative) Character naming guidelines ... 2781
Annex M (informative) Sources of characters ... 2784
Annex N (informative) External references to character repertoires ... 2785
N.1 Methods of reference to character repertoires and their coding ... 2785
N.2 Identification of ASN.1 character abstract syntaxes ... 2785
N.3 Identification of ASN.1 character transfer syntaxes ... 2786
Annex P (informative) Additional information on CJK Unified ideographs ... 2787
Annex Q (informative) Code mapping table for Hangul syllables ... 2790
Annex R (informative) Names of Hangul syllables ... 2791
Annex S (informative) Procedure for the unification and arrangement of CJK ideographs ... 2792
S.1 Unification procedure ... 2792
S.2 Arrangement procedure ... 2796
S.3 Source separation examples ... 2796
S.4 Non-unification examples ... 2801
Annex T (informative) Language tagging using Tag Characters ... 2803
Annex U (informative) Characters in identifiers ... 2804

© ISO/IEC 2020 – All rights reserved

AVA Documents
Document Sharing Hub

Full document details: ISO IEC 10646-2020

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see http://patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organiza-tion (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 2, *Coded character sets.*

This sixth edition of ISO/IEC 10646 cancels and replaces the fifth edition (ISO/IEC 10646:2017), which has been technically revised. It also incorporates ISO/IEC 10646:2017/Amd 1:2019 and ISO/IEC 10646:2017/Amd 2:2019.

This edition includes the following significant changes with respect to the previous edition:

- New scripts covered: Chorasmian, Dives Akuru, Dogra, Elymaic, Gunjala Gondi, Hanifi Rohingya, Khitan Small Script, Makasar, Medefaidrin, Nandinagari, Nyiakeng Puachue Hmong, Old Sogdian, Sogdian, Yezidi, Wancho;
- Existing scripts significantly extended: Georgian, CJK Unified Ideographs (Extension G);
- New symbol sets: Chess Symbols, Symbols for Legacy Computing;
- New set of Emoji symbols.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO IEC 10646-2020

Introduction

This document specifies the Universal Coded Character Set (UCS). It is applicable to the representation, transmission, interchange, processing, storage, input and presentation of the written form of the languages of the world as well as additional symbols.

By defining a consistent way of encoding multilingual text it enables the exchange of data internationally. The information technology industry gains data stability, greater global interoperability and data interchange. This International Standard has been widely adopted in new Internet protocols and implemented in modern operating systems and computer languages. This edition covers over 130 000 characters from the world's scripts.

The UCS is an encoding system different from that specified in ISO/IEC 2022. The method to designate UCS from ISO/IEC 2022 is specified in 13.2.

A graphic character will be assigned only one code point in the standard, located either in the BMP or in one of the supplementary planes.

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO IEC 10646-2020

Information technology — Universal Coded Character Set (UCS)

1 Scope

This document

- specifies the architecture of the UCS;
- defines terms used for the UCS;
- describes the general structure of the UCS codespace;
- specifies the assigned planes of the UCS: the Basic Multilingual Plane (BMP) of the UCS, the Supplementary Multilingual Plane (SMP), the Supplementary Ideographic Plane (SIP), the Tertiary Ideographic Plane (TIP), and the Supplementary Special-purpose Plane (SSP);
- defines a set of graphic characters used in scripts and the written form of languages on a world-wide scale;
- specifies the names for the graphic characters and format characters of the BMP, SMP, SIP, TIP, SSP and their coded representations within the UCS codespace;
- specifies the coded representations for control characters and private use characters;
- specifies three encoding forms of the UCS: UTF-8, UTF-16, and UTF-32;
- specifies seven encoding schemes of the UCS: UTF-8, UTF-16, UTF-16BE, UTF-16LE, UTF-32, UTF-32BE, and UTF-32LE;
- specifies the management of future additions to this coded character set.

NOTE – The determination of suitability of these characters for use as identifiers in programming languages is not specified by this document but can be found in an external reference. See Annex U.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 2022 *Information technology — Character code structure and extension techniques.*

ISO/IEC 6429 *Information technology — Control functions for coded character sets.*

Unicode Standard Annex, UAX #9, *The Unicode Bidirectional Algorithm:*
http://www.unicode.org/reports/tr9/tr9-42.html

Unicode Standard Annex, UAX #15, *Unicode Normalization Forms:*
http://www.unicode.org/reports/tr15/tr15-50.html

Unicode Technical Standard, *UTS #37, Ideographic Variation Database:*
http://www.unicode.org/reports/tr37/tr37-12.html

Unicode Standard Version 13.0, *Chapter 4, Character Properties*
http://www.unicode.org/versions/Unicode13.0.0/ch04.pdf
Section 4.3, Combining Classes – Normative
Section 4.5, General Category – Normative
Section 4.7, Bidi Mirrored – Normative

Unicode Standard Version 12.1, *Age Property:*
https://www.unicode.org/Public/13.0.0/ucd/DerivedAge.txt

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO IEC 10646-2020

Note – Parts of this document which use machine-readable format are available as electronic data attachments. See Clause 5.

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— IEC Electropedia: available at http://www.electropedia.org/

— ISO Online browsing platform: available at http://www.iso.org/obp

3.1
base character
graphic character which is not a combining character

Note 1 to entry – Most graphic characters are base characters. This sense of graphic combination does not preclude the presentation of base characters from adopting different contextual forms or from participating in ligatures.

Note 2 to entry – A base character typically does not graphically combine with preceding characters. There are exceptions for some complex writing systems.

3.2
Basic Multilingual Plane
BMP
plane 00 of the UCS codespace

3.3
block
contiguous range of code points to which a set of characters that share common characteristics, such as a script, are allocated; a block does not overlap another block; one or more of the code points within a block may have no character allocated to them

3.4
canonical form
form with which characters of this coded character set are specified using a single code point within the UCS codespace

Note 1 to entry – The canonical form is not to be confused with an encoding form which describes the relationship between UCS code points and one or several code units (see 3.23).

3.5
character
member of a set of elements used for the organization, control, or representation of textual data

Note 1 to entry – A graphic symbol can be represented by a sequence of one or several coded characters.

3.6
character boundary
(code unit sequence) demarcation between the last code unit of a coded character and the first code unit of the next coded character

3.7
code chart
code table
rectangular array showing the representation of coded characters allocated within a range of the UCS codespace

3.8
coded character
association between a character and a code point

 © ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO IEC 10646-2020

3.9
code point
code position
value in the UCS codespace

Note 1 to entry – Code points in the UCS codespace are integer values. Throughout this document, UCS code points are cited in hexadecimal. UCS code points range from 0 to 10FFFF.

3.10
code unit
minimal bit combination that can represent a unit of encoded text for processing or interchange

Note 1 to entry – Examples of code units are octets (8-bit code units) used in the UTF-8 encoding form, 16-bit code units in the UTF-16 encoding form, and 32-bit code units in the UTF-32 encoding form.

3.11
code unit sequence
CC-data-element
coded-character-data-element
element of interchanged information that is specified to consist of a sequence of code units, in accordance with one or more identified standards for coded character sets

Note 1 to entry – Such sequence can contain code units associated with any type of code point (see 7.3).

Note 2 to entry – Since its second edition: ISO/IEC 10646:2011, this document does not use implementation levels. Its definition of code unit sequence corresponds to the former unrestricted implementation level 3. Other definitions of code unit sequence, previously known as level 1 and 2, are deprecated. To maintain compatibility with these previous editions, in the context of identification of coded representation in International Standards such as ISO/IEC 8824 and ISO/IEC 8825, the concept of implementation level can still be referenced as 'Implementation level 3'. See Annex N.

3.12
collection
numbered and named set of entities made of code points or sequences of code points, the sequences conforming to Normalization C; code points lie within one or more identified ranges

Note 1 to entry – Non extended collections do not contain sequences of code points (see 3.25 for extended collection).

Note 2 to entry – If any of the identified ranges include code points to which no character is allocated, the repertoire of the collection will change if an additional character is assigned to any of those code points at a future amendment of this document. However, it is intended that the collection number and name will remain unchanged in future editions of this document.

3.13
combining character
character which has General Category values of Spacing Combining Mark (Mc), Non Spacing Mark (Mn), and Enclosing Mark (Me)

Note 1 to entry – These characters are intended for combination with the preceding base character, or with a sequence of combining characters preceded by a base character (see also 3.17).

3.14
combining class
value associated with each combining character determining its typographical interaction and its canonical ordering within a sequence of combining character

Note 1 to entry – See 21.2 for details on canonical ordering.

3.15
compatibility character
graphic character included as a coded character of this document primarily for compatibility with existing coded character sets

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO IEC 10646-2020

3.16
composite sequence
combining character sequence
sequence of graphic characters consisting of a base character followed by one or more combining characters, ZERO WIDTH JOINER, or ZERO WIDTH NON-JOINER

Note 1 to entry – See also 3.14.

Note 2 to entry – A graphic symbol for a composite sequence generally consists of the combination of the graphic symbols of each character in the sequence.

Note 3 to entry – A composite sequence can be used to represent characters not encoded in the repertoire of this document.

3.17
control character
control function the coded representation of which consists of a single code point

Note 1 to entry – Although control characters are often 'named' using terms such as DELETE, FORM FEED, ESC, these qualifiers do not correspond to formal character names. See Clause 12 for a list of the long names used by ISO/IEC 6429 in association with the control characters.

3.18
control function
action that affects the recording, processing, transmission, or interpretation of data, and that is represented by a code unit sequence

3.19
decomposition mapping
mapping from a character to a sequence of one or more characters

Note 1 to entry – Decomposition mappings are of two types: canonical decompositions, and compatibility decompositions. These are used in the derivation of various normalization forms (see Clause 22). The code charts for various blocks include decomposition mappings and distinguish between the two types of mapping (see 34.3).

3.20
default state
state that is assumed when no state has been explicitly specified

Note 1 to entry – See F.2.1, F.2.2, and F.2.3.

3.21
device
component of information processing equipment which can transmit and/or receive coded information within code unit sequences

Note 1 to entry – It may be an input/output device in the conventional sense, or a process such as an application program or gateway function.

3.22
encoding form
form that determines how each UCS code point for a UCS character is to be expressed as one or more code units used by the encoding form

Note 1 to entry – This document specifies UTF-8, UTF-16, and UTF-32.

3.23
encoding scheme
scheme that specifies the serialization of the code units from the encoding form into octets

Note 1 to entry – Some of the UCS encoding schemes have the same labels as UCS encoding form. However, references to encoding schemes and encoding forms generally occur in different contexts. UCS encoding forms refer to in-memory and application interface representation of textual data. UCS encoding schemes refer to octet-serialized textual data.

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO IEC 10646-2020

3.24
extended collection
collection for which the entities can also consist of sequences of code points that are in Normalization Form C (NFC)

Note 1 to entry – Some collections such as 3 LATIN EXTENDED-A, 4 LATIN EXTENDED-B, 15 ARABIC EXTENDED, and many more, have the term 'extended' in their name. This does not make them extended collections.

Note 2 to entry – See Clause 22 for discussion of Normalization Form C.

Note 3 to entry – The sequences of code points are typically referenced by Named UCS Sequence Identifiers (NUSI) (see Clause 28).

3.25
fixed collection
collection in which every code point within the identified range(s) has a character allocated to it, and which is intended to remain unchanged in future editions of this document

3.26
format character
character whose primary function is to affect the layout or processing of characters around it, or that is presented in a complex, graphic interaction with neighbouring characters

Note 1 to entry – A format character generally does not have a visible representation of its own.

3.27
General Category
GC
value assigned to each UCS code point which determines its major class, such as letter, punctuation, and symbol

Note 1 to entry – Possible values are two-letter abbreviations for the General Category in the Unicode Standard (see reference to the current Unicode Standard General Category in Clause 2).

Note 2 to entry – When referred as a group containing all GC values sharing the same first letter, the group may be described using the first letter only. For example, 'L' stands for all letters: 'Lu', 'Ll', 'Lt', 'Lm', and 'Lo'.

3.28
graphic character
character, other than a control function or a format character, that has a visual representation normally handwritten, printed, or displayed

3.29
graphic symbol
visual representation of a graphic character or of a composite sequence

3.30
high-surrogate code point
code point in the range D800 to DBFF

Note 1 to entry – Reserved for use in UTF-16 (see 10.3).

3.31
high-surrogate code unit
16-bit code unit in the range D800 to DBFF and used in UTF-16

Note 1 to entry – A high-surrogate code unit is used as the leading code unit of a surrogate pair (see also 3.39, 3.54, and 10.3).

3.32
ill-formed code unit sequence
UCS code unit sequence that purports to be in a UCS encoding form that does not conform to the specification of that encoding form

EXAMPLE – An unpaired surrogate code unit is an ill-formed code unit sequence.

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 10646-2020

3.33
ill-formed code unit subsequence
non-empty subsequence of a code unit sequence X that does not contain any code units that also belong to a minimal well-formed code unit subsequence of X

Note 1 to entry – An ill-formed code unit subsequence cannot overlap with a minimal well-formed code unit subsequence.

3.34
interchange
transfer of character coded data from one user to another, using telecommunication means or interchangeable media

Note 1 to entry – Interchange implies data serialization and the use of a UCS encoding scheme.

3.35
ISO/IEC 10646-1
former subdivision of ISO/IEC 10646 containing the specification of the overall UCS architecture and the Basic Multilingual Plane (BMP)

Note 1 to entry – It is also referred to as Part 1 of ISO/IEC 10646.

Note 2 to entry – There are a first and a second Edition of ISO/IEC 10646-1.

3.36
ISO/IEC 10646-2
former subdivision of ISO/IEC 10646 containing the specification of the Supplementary Multilingual Plane (SMP), the Supplementary Ideographic Plane (SIP) and the Supplementary Special-purpose Plane (SSP)

Note 1 to entry – It is also referred to as Part 2 of ISO/IEC 10646.

Note 2 to entry – There is only a first edition of ISO/IEC 10646-2.

3.37
low-surrogate code point
code point in the range DC00 to DFFF

Note 1 to entry – Reserved for the use of UTF-16 (see 10.3).

3.38
low-surrogate code unit
16-bit code unit in the range DC00 to DFFF and used in UTF-16

Note 1 to entry – A low-surrogate code unit is used as the trailing code unit of a surrogate pair (see also 3.32, 3.54, and 10.3).

3.39
minimal well-formed code unit sequence
well-formed code unit sequence that maps to a single UCS scalar value

3.40
mirrored character
character whose image is mirrored horizontally in text that is laid out from right to left

3.41
octet
8-bit code unit

Note 1 to entry – The value is expressed in hexadecimal notation from 00 to FF in this document (see Annex K).

3.42
plane
subdivision of the UCS codespace consisting of 65 536 contiguous code points beginning at a multiple of 65 536

Note 1 to entry – The 17 UCS planes can be identified by a hexadecimal number from 00 to 10.

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO IEC 10646-2020

3.43
presentation
process of writing, printing, or displaying a graphic symbol

3.44
presentation form
(in the presentation of some scripts) form of a graphic symbol representing a character that depends on the position of the character relative to other characters

3.45
private use plane
plane within this coded character set, the content of which is not specified in this document

Note 1 to entry – Planes 0F and 10 are private use planes.

3.46
repertoire
specified set of characters that are represented in a coded character set

3.47
row
subdivision of a plane consisting of 256 contiguous code points beginning at a multiple of 256

Note 1 to entry – Within the context of a given plane, rows can be identified by a hexadecimal number from 00 to FF.

3.48
script
set of graphic characters used for the written form of one or more languages

3.49
supplementary plane
plane other than Plane 00 of the UCS codespace

Note 1 to entry – A supplementary plane accommodates characters which have not been allocated to the Basic Multilingual Plane.

3.50
Supplementary Multilingual Plane
SMP
plane 01 of the UCS codespace

3.51
Supplementary Ideographic Plane
SIP
plane 02 of the UCS codespace

3.52
Supplementary Special-purpose Plane
SSP
plane 0E of the UCS codespace

3.53
surrogate pair
UTF-16 encoded representation for a single supplementary-plane character that consists of a sequence of two 16-bit code units, where the first value of the pair is a high-surrogate code unit and the second value is a low-surrogate code unit

3.54
Tertiary Ideographic Plane
TIP
plane 03 of the UCS codespace

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO IEC 10646-2020

3.55
UCS codespace
codespace consisting of the integers from 0 to 10FFFF (hexadecimal) available for assigning the repertoire of the UCS characters

3.56
UCS scalar value
any UCS code point except high-surrogate and low-surrogate code points

3.57
unpaired surrogate code unit
code unit in a a UTF-16 code unit sequence that is either a high-surrogate code unit that is not immediately followed by a low-surrogate unit, or a low-surrogate code unit that is not immediately preceded by a high-surrogate code unit

Note 1 to entry – Any unpaired surrogate code unit constitutes an ill-formed code unit sequence.

3.58
user
person or other entity that invokes the service provided by a device

Note 1 to entry – This entity can be a process such as an application program if the "device" is a code converter or a gateway function.

3.59
well-formed code unit sequence
UCS code unit sequence that purports to be in a UCS encoding form that conforms to the specification of that encoding form and contains no ill-formed code unit subsequence

4 Conformance

4.1 General

Whenever private use characters are used as specified in this document, the characters themselves shall not be covered by these conformance requirements.

4.2 Conformance of information interchange

A code unit sequence (CC-data-element) within coded information for interchange is in conformance with this document if

a) all the coded representations of graphic characters within that code unit sequence conform to Clause 7, to an identified encoding form chosen from Clause 10, and to an identified encoding scheme chosen from Clause 11;

b) all the graphic characters represented within that code unit sequence are taken from those within an identified subset (see Clause 9);

c) all the coded representations of control functions within that code unit sequence conform to Clause 12.

A claim of conformance shall identify the adopted encoding form, the adopted encoding scheme, and the adopted subset by means of a list of collections and/or characters.

4.3 Conformance of devices

A device is in conformance with this document if it conforms to the requirements of item a) below, and either or both of items b) and c).

A claim of conformance shall identify the document that contains the description specified in a) below, and shall identify the adopted encoding form(s), the adopted encoding scheme(s), and the adopted subset (by means of a list of collections and/or characters), and the selection of control functions adopted in accordance with Clause 12.

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO IEC 10646-2020

a) **Device description**: A device that conforms to this document shall be the subject of a description that identifies how the user may supply characters to the device and/or may recognize them when they are made available to the user, as specified respectively, in items b) and c) below.

b) **Originating device**: An originating device shall allow its user to supply any characters from an adopted subset, and to be capable of transmitting their coded representations within a code unit sequence in accordance with the adopted encoding form and adopted encoding scheme. As such, the originating device shall not emit ill-formed code unit sequences.

c) **Receiving device**: A receiving device shall be capable of receiving and interpreting any coded representation of characters that are within a code unit sequence in accordance with the adopted encoding form and the adopted encoding scheme, and shall make any corresponding characters from the adopted subset available to the user in such a way that the user can identify them. The receiving device shall treat ill-formed code unit sequences as an error condition and shall not interpret such data as character sequences.

Any corresponding characters that are not within the adopted subset shall be indicated to the user. The way used for indicating them need not distinguish them from each other.

NOTE 1 – The manner in which a user is notified of either an error condition or characters not within the adopted subset is not specified by this document.

NOTE 2 – See also Annex J for receiving devices with retransmission capability.

5 Electronic data attachments

The following electronic data attachments constitute an integral part of this document. They use machine-readable formats that make them more convenient as separate data attachments.

Variation selectors and variation sequences:
http://standards.iso.org/iso-iec/10646/ed-6/en/UCSVariants.txt

Emoji variation sequences:
http://standards.iso.org/iso-iec/10646/ed-6/en/emoji-variation-sequences.txt

Source references for pictographic symbols
http://standards.iso.org/iso-iec/10646/ed-6/en/EmojiSrc.txt

Source references for CJK ideographs:
http://standards.iso.org/iso-iec/10646/ed-6/en/CJKSrc.txt

Source references for Nüshu characters:
http://standards.iso.org/iso-iec/10646/ed-6/en/NushuSrc.txt

Source references for Tangut ideographs:
http://standards.iso.org/iso-iec/10646/ed-6/en/TangutSrc.txt

Named UCS Sequence Identifiers:
http://standards.iso.org/iso-iec/10646/ed-6/en/NUSI.txt

Collection MOJI-JOHO-KIBAN IDEOGRAPHS-2016:
http://standards.iso.org/iso-iec/10646/ed-6/en/JMJKI-2016.txt

Collection MOJI-JOHO-KIBAN IDEOGRAPHS-2018:
http://standards.iso.org/iso-iec/10646/ed-6/en/JMJKI-2018.txt

Collection JAPANESE JIS X 0213:2004 IDEOGRAPHS FROM PREVIOUS JIS STANDARDS:
http://standards.iso.org/iso-iec/10646/ed-6/en/JIS-X-0213-FromPrevious.txt

Collection JAPANESE CORE KANJI:
http://standards.iso.org/iso-iec/10646/ed-6/en/JapaneseCoreKanji.txt

Alphabetically sorted list of character names:
http://standards.iso.org/iso-iec/10646/ed-6/en/Allnames.txt

Names of Hangul syllables:
http://standards.iso.org/iso-iec/10646/ed-6/en/HangulSy.txt

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO IEC 10646-2020

6 General structure of the UCS

The general structure of the Universal Coded Character Set (referred to hereafter as “this coded character set”) is described in this explanatory clause and is illustrated in figure 1. The normative specification of the structure is given in the following clauses.

The canonical form of this coded character set – the way in which it is to be conceived – uses the UCS codespace which consists of the integers from 0 to 10FFFF.

This document defines coded characters for the following planes:

- Basic Multilingual Plane (BMP, Plane 00);
- Supplementary Multilingual Plane (SMP, Plane 01);
- Supplementary Ideographic Plane (SIP, Plane 02);
- Tertiary Ideographic Plane (TIP, Plane 03);
- Supplementary Special-purpose Plane (SSP, Plane 0E).

Planes from 04 to 0D are reserved for future standardization.

Planes 0F and 10 are reserved for private use.

Subsets of the codespace may be used to give a sub-repertoire of graphic characters.

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO IEC 10646-2020

7 Basic structure and nomenclature

7.1 Structure

The Universal Coded Character Set as specified in this document shall be regarded as a single entity made of 17 planes.

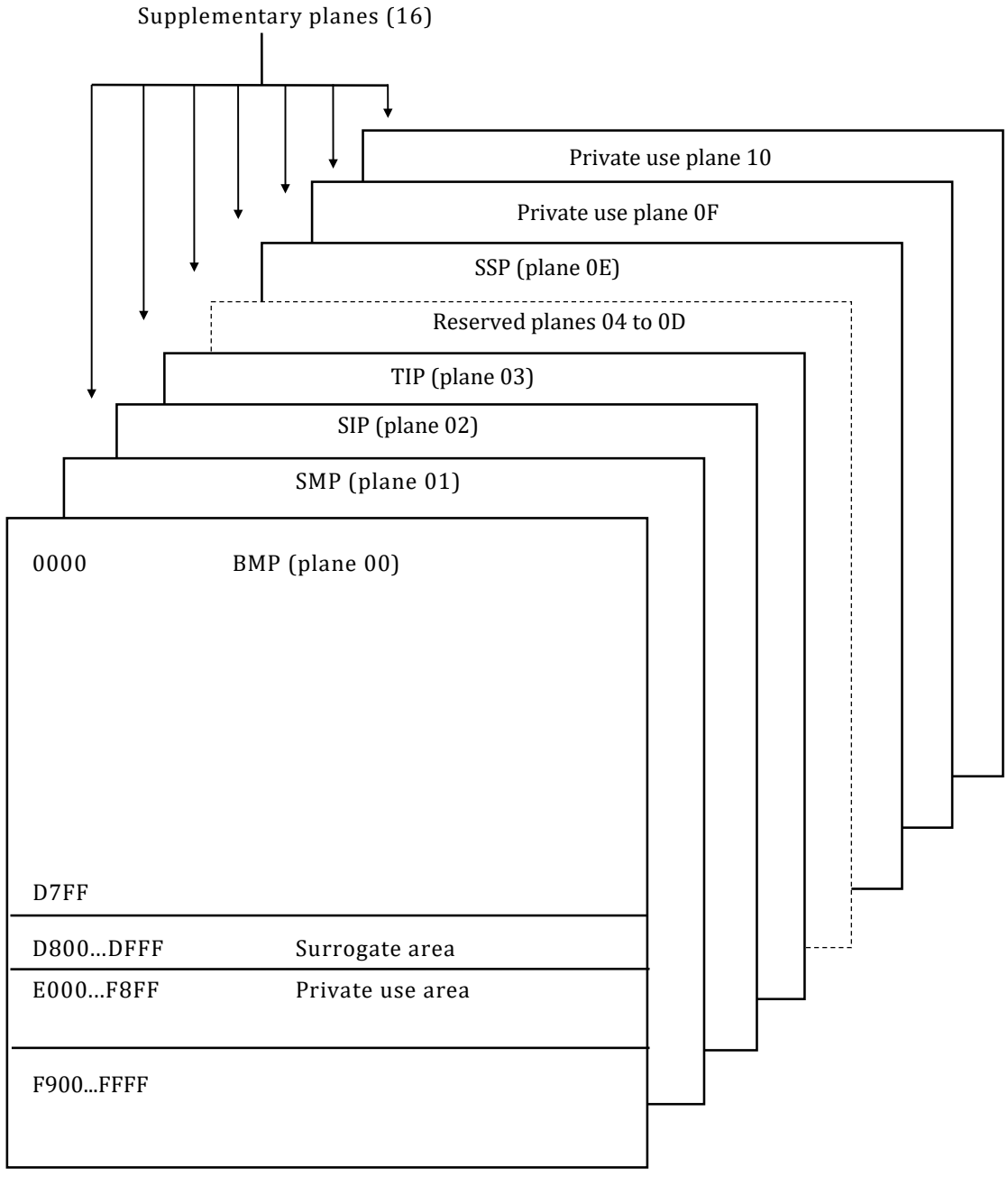


Figure 1 - Planes of the Universal Coded Character Set

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 10646-2020

7.2 Coding of characters

Each encoded character within the UCS codespace is represented by an integer between 0 and 10FFFF identified as a code point.

When a single character is to be identified in terms of its code point, it is represented by a six-digit form of the integer such as:

000030 for DIGIT ZERO
000041 for LATIN CAPITAL LETTER A
010000 for LINEAR B SYLLABLE B008 A

When referring to characters within plane 00, the leading two digits may be omitted; for characters within planes 01 to 0F, the leading digit may be omitted. For example:

0030 for DIGIT ZERO
0041 for LATIN CAPITAL LETTER A
10000 for LINEAR B SYLLABLE B008 A

7.3 Types of code points

7.3.1 Classification

UCS code points are categorized into basic types, according to their General Category value. These values shall be determined according to the Unicode Standard General Category property (see Clause 2). Table 1 summarizes the types:

Table 1: Type of code points

Basic Type	Brief Description	General Category	Character status	Code point status
Graphic	Letter, mark, number, punctuation, symbols, and spaces	L, M, N, P, S, Zs	Assigned to character	Assigned code point
Format	Affects layout or processing of neighbouring characters, or has a complex graphic interaction with neighbouring characters	Cf, Zl, Zp		
Control	Control functions defined in ISO/IEC 6429	Cc		
Private use	Usage defined by private agreement outside this standard	Co		
Surrogate	Permanently reserved for UTF-16	Cs	Not assigned to character	
Noncharacter	Permanently reserved for internal usage	Cn		
Reserved	Reserved for future assignment			Unassigned code point

Surrogate, noncharacter, and reserved code points are not assigned to characters and are subject to restriction in interchange. For example, surrogate code points do not have well-formed representations in any UCS encoding form.

7.3.2 Graphic characters

The same graphic character shall not be allocated to more than one code point. There are graphic characters with similar shapes in the coded character set; they are used for different purposes and have different character names.

7.3.3 Format characters

Format characters form a class of characters that affect or strongly interact with neighbouring characters.

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 10646-2020

7.3.4 Control characters

Code points 0000 to 001F, 007F to 009F in the BMP are reserved for control characters (see Clause 12).

7.3.5 Private use characters

Code points from E000 to F8FF in the BMP are reserved for private use. All code points of Plane 0F and Plane 10 – except for noncharacter code points FFFFE, FFFFF, 10FFFE, and 10FFFF – are reserved for private use.

Private use characters are not constrained in any way by this document. Private use characters can be used to provide user-defined characters. For example, this is a common requirement for users of ideographic scripts.

NOTE – For meaningful interchange of private use characters, an agreement, independent of this document, is necessary between sender and recipient.

7.3.6 Surrogate code points

Code points D800 to DFFF are reserved for the use of the UTF-16 encoding form (see 10.3). The first half (D800 to DBFF) contains the high-surrogate code points and the second half (DC00 to DFFF) contains the low-surrogate code points.

7.3.7 Noncharacter code points

The status of noncharacter code points cannot be changed by future amendments. Noncharacters consist of FDD0-FDEF and any code point ending in the value FFFE or FFFF.

NOTE – Code point FFFE is reserved for use as a "signature" for detecting correct byte order in UTF-16 or UTF-32 text data streams. Because the UTF-16 and UTF-32 encoding forms use 16-bit and 32-bit code units, but many processes handle data streams as byte sequences, the ordering of bytes within UTF-16 or UTF-32 code units strongly affects the interpretation of text data streams encoded in these encoding forms. The character FEFF ZERO WIDTH NO-BREAK SPACE, which has a minimal effect on the meaning or processing of text, can be included in a text data stream. If the bytes for this character were interpreted using the wrong byte order, then the bytes would be interpreted as the noncharacter code point FFFE. Since inclusion of this noncharacter code point is not expected in valid text content, the process would be able recognize the correct byte order. See also Clause 11.

NOTE 2 – Code points FDD0 to FDEF, and FFFF can be used for internal processing uses requiring numeric values which are guaranteed not to be coded characters, such as in terminating tables, or signaling end-of-text. Furthermore, since FFFF is the largest BMP value, it may also be used as the final value in binary or sequential searching index within the context of UTF-16.

7.3.8 Reserved code points

Reserved code points are reserved for future standardization and shall not be used for any other purpose. Future editions of this document may allocate characters to some of these reserved code points.

7.4 Naming of characters

This document assigns a unique name to each graphic and format character. The name of a character either

a) denotes the customary meaning of the character, or

b) describes the shape of the corresponding graphic symbol, or

c) follows the rule given in 27.6 for Chinese /Japanese/Korean (CJK) ideographs, or

d) follows the rule given in 27.7 for Tangut ideographs, or

e) follows the rules given in 27.8 for Nüshu characters, or

f) follows the rules given in 27.9 for Khitan Small Script characters, or

g) follows the rule given in 27.10 for Hangul syllables.

Some characters may have one or more alternate names, called character name aliases, which are correction of the original names.

NOTE — Character name aliases, which are normative, should not be confused with informative aliases, which are other names for characters that may be used outside this document but that are not normative

Additional rules to be used for constructing the names of characters are given in Clause 27.

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO IEC 10646-2020

The list of character names, except for CJK ideographs, Tangut ideographs and Hangul syllables, is provided in Clause 34.

NOTE – The list of character names is also part of the Unicode character Database in: http://www.unicode.org/Public/UNIDATA/NamesList.txt with the syntax described in: http://www.unicode.org/Public/UNIDATA/NamesList.html.

7.5 Short identifiers for code points (UIDs)

This document defines short identifiers for each code point, including code points that are reserved (unassigned). A short identifier for any code point is distinct from a short identifier for any other code point. If a character is allocated at a code point, a short identifier for that code point can be used to refer to the character allocated at that code point.

NOTE 1 – For instance, U+DC00 identifies a surrogate code point, and U+FFFF identifies a noncharacter code point. U+0025 identifies a graphic code point to which a graphic character is allocated; U+0025 also identifies that character (named PERCENT SIGN).

NOTE 2 – These short identifiers are independent of the language in which this standard is written and are thus retained in all translations of the text.

The following alternative forms of notation of a short identifier are defined here.

a) The six-digit form of short identifier consists of the sequence of six hexadecimal digits that represents the code point of the character (see 7.2).

b) The four-to-five-digit form of short identifier shall consist of the last four to five digits of the six-digit form, with all non-zero digits kept but any leading zeroes beyond four digits suppressed.

c) The character “+” (PLUS SIGN) may, as an option, precede the digit form of short identifier.

d) The prefix letter “U” (LATIN CAPITAL LETTER U) may, as an option, precede any of the three forms of short identifier defined in a) to c) above.

The capital letters A to F, and U that appear within short identifiers may be replaced by the corresponding small letters.

The full syntax of the notation of a short identifier, in Backus-Naur form, is

{ U | u } {+}(xxxx | xxxxx | xxxxxx)

where “x” represents one hexadecimal digit (0 to 9, A to F, or a to f).

EXAMPLE

The short identifier for LATIN SMALL LETTER LONG S may be notated in any of the following forms:

017F +017F U017F U+017F

Any of the capital letters may be replaced by the corresponding small letter.

7.6 UCS Sequence Identifiers

This document defines an identifier for any sequence of code points taken from the standard. Such an identifier is known as a UCS Sequence Identifier (USI). For a sequence of n code points, it has the following form:

<UID1, UID2, ..., UIDn>

where UID1, UID2, etc. represent the short identifiers of the corresponding code points, in the same order as those code points appear in the sequence. If each of the code points in such a sequence has a character allocated to it, the USI can be used to identify the sequence of characters allocated at those code points. The syntax for UID1, UID2, etc. is specified in 7.5, with an added constraint that, within a USI, the UIDs shall not include a ‘U’, ‘u, or ‘+’ prefix. A COMMA character (optionally followed by a SPACE character) separates the UIDs. A UCS Sequence Identifier shall include at least two UIDs; it begins with a LESS-THAN SIGN and is terminated by a GREATER-THAN SIGN.

The full syntax of the notation of a UCS Sequence Identifier, in Backus-Naur form, is

“<” (xxxx | xxxxx | xxxxxx) (“,” space?) (xxxx | xxxxx | xxxxxx))+ “>”

 © ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO IEC 10646-2020

where “x” represents one hexadecimal digit (0 to 9, A to F, or a to f).

UCS Sequences Identifiers cannot be used for specification of subset content. They may be used outside this standard to identify composite sequences for mapping purposes, font repertoire, etc.

7.7 Octet sequence identifiers

To represent serialized octet in the context of the encoding schemes definition (see Clause 11), this document defines an identifier for serialized octet sequence. For a sequence of n octets, it has the following form:

<xx_1 xx_2 ... xx_n>

where xx_1, xx_2, and xx_n, represents the first, second, and n^{th} octets using two hexadecimal digits for each octet.

8 Revision and updating of the UCS

The names and code points allocation of all characters in this coded character set shall remain unchanged in all future editions and amendments of this standard. This also includes character name aliases.

NOTE – Character name aliases are created to denote errors in the character names which cannot be fixed after publication of the standard. These character name aliases are described in the file NameAliases.txt part of the Unicode character database (http://www.unicode.org/Public/UCD/latest/ucd/NameAliases.txt).

9 Subsets

9.1 General

This document provides the specification of subsets of coded graphic characters for use in interchange, by originating devices, and by receiving devices.

There are two alternatives for the specification of subsets: limited subset and selected subset. An adopted subset may comprise either of them, or a combination of the two.

9.2 Limited subset

A limited subset consists of a list of graphic characters in the specified subset. This specification allows applications and devices that were developed using other codes to interwork with this coded character set.

A claim of conformance referring to a limited subset shall list the graphic characters in the subset by the names of graphic characters or code points as defined in this document.

9.3 Selected subset

A selected subset consists of a list of collections of graphic characters as defined in this document. The collections from which the selection may be made are listed in Annex A. A selected subset shall always automatically and implicitly include the code points from 0020 to 007E, even if the corresponding collection 1 BASIC LATIN, is not explicitly listed in the subset specification.

A claim of conformance referring to a selected subset shall list the collections chosen as defined in this document.

10 UCS encoding forms

10.1 General

This document provides three encoding forms expressing each UCS scalar value in a unique sequence of one or more code units. These are named UTF-8, UTF-16, and UTF-32 respectively.

10.2 UTF-8

UTF-8 is the UCS encoding form that assigns each UCS scalar value to an octet sequence of one to four octets, as specified in table 2.

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO IEC 10646-2020

- UCS characters from the BASIC LATIN collection are represented in UTF-8 in accordance with ISO/IEC 4873, i.e. single octets with values ranging from 20 to 7E.
- Control functions in code points from 0000 to 001F, and the control character in code point 007F, are represented without the padding octets specified in Clause 12, i.e. as single octets with values ranging from 00 to 1F, and 7F respectively in accordance with ISO/IEC 4873 and with the 8-bit structure of ISO/IEC 2022.
- Octet values 00 to 7F do not otherwise occur in the UTF-8 coded representation of any character. This provides compatibility with existing file-handling systems and communications sub-systems which parse code unit sequences for these octet values.
- The first octet in the UTF-8 coded representation of any character can be directly identified when a code unit sequence is examined, one octet at a time, starting from an arbitrary location. The first code unit indicates the number of continuing octets (if any) in the multi-octet sequence that constitutes the code unit representation of that character.

Table 2 specifies the bit distribution for the UTF-8 encoding form, showing the ranges of UCS scalar values corresponding to one, two, three, and four octet sequences.

Table 2: UTF-8 Bit distribution

Scalar values	Scalar value bit distribution	1st octet	2nd octet	3rd octet	4th octet
0000 to 007F	000000000xxxxxxx	0xxxxxxx			
0080 to 07FF	00000yyyyyxxxxxx	110yyyyy	10xxxxxx		
0800 to D7FF, E000 to FFFF	zzzzyyyyyyxxxxxx	1110zzzz	10yyyyyy	10xxxxxx	
10000 to 10FFFF	000uuuuuzzzzyyyyyyxxxxxx	11110uuu	10uuzzzz	10yyyyyy	10xxxxxx

Because surrogate code points are not UCS scalar values, any UTF-8 sequence that would otherwise map to code points D800-DFFF is ill-formed.

Table 3 lists all the ranges (inclusive) of the octet sequences that are well-formed in UTF-8. Any UTF-8 sequence that does not match the patterns listed in table 3 is ill-formed.

Table 3: Well-formed UTF-8 Octet sequences

Code points	1st octet	2nd octet	3rd octet	4th octet
0000-007F	00-7F			
0080-07FF	C2-DF	80-BF		
0800-0FFF	E0	A0-BF	80-BF	
1000-CFFF	E1-EC	80-BF	80-BF	
D000-D7FF	ED	80-9F	80-BF	
E000-FFFF	EE-EF	80-BF	80-BF	
10000-3FFFF	F0	90-BF	80-BF	80-BF
40000-FFFFF	F1-F3	80-BF	80-BF	80-BF
100000-10FFFF	F4	80-8F	80-BF	80-BF

Because of the well-formedness conditions specified in table 9.2, the following octet values are disallowed in UTF-8: C0-C1, F5-FF.

10.3 UTF-16

UTF-16 is the UCS encoding form that assigns each UCS scalar value to a sequence of one to two unsigned 16-bit code units, as specified in table 4.

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO IEC 10646-2020

In the UTF-16 encoding form, code points in the range 0000-D7FF and E000-FFFF are represented as a single 16-bit code unit; code points in the range 10000-10FFFF are represented as pairs of 16-bit code units. These pairs of special code units are known as surrogate pairs.

The values of the code units used for surrogate pairs are disjoint from the code units used for the single code unit representation, thus maintaining non-overlap for all code point representations in UTF-16.

UTF-16 optimizes the representation of characters in the BMP which contains most common use characters.

Because surrogate code points are not UCS scalar values, unpaired surrogate code units are ill-formed.

Table 4 specifies the bit distribution for the UTF-16 encoding form. Calculation of the surrogate pair values involves subtraction of 10000 hexadecimal to account for the starting offset to the scalar value (expressed as 'wwww = uuuuu-1' in the table).

Table 4: UTF-16 Bit distribution

Scalar values	Scalar value bit distribution	1st code unit	2nd code unit
0000 to D7FF, E000 to FFFF	xxxxxxxxxxxxxxxx	xxxxxxxxxxxxxxxx	
10000 to 10FFFF	000uuuuuxxxxxxxxxxxxxxxx	110110wwwwxxxxxx	110111xxxxxxxxxx

NOTE – Former editions of this document included references to a two-octet BMP form called UCS-2 which would be a subset of the UTF-16 encoding form restricted to the BMP UCS scalar values. The UCS-2 form is deprecated.

10.4 UTF-32

UTF-32 is the UCS encoding form that assigns each UCS scalar value to a single unsigned 32-bit code unit.

NOTE — Former editions of this document included "UCS-4" as an alternate term synonymous with "UTF-32". Use of the term "UCS-4" to refer to this encoding form is deprecated.

Because surrogate code points are not UCS scalar values, UTF-32 code units in the range 0000 D800 - 0000 DFFF are ill-formed.

11 UCS encoding schemes

11.1 General

Encoding schemes are octet serializations specific to each UCS encoding form, including the specification of a signature, if allowed. The signature is the code unit sequence corresponding to the code point FEFF ZERO WIDTH NO-BREAK SPACE in the corresponding encoding form. When used, a signature at the beginning of a stream of serialized octets indicates the order of the octets within the encoding form used for the representation of the characters. See also 7.3.7.

This document specifies seven encoding schemes: UTF-8, UTF-16BE, UTF-16LE, UTF-16, UTF-32BE, UTF-32LE, and UTF-32.

11.2 UTF-8

The UTF-8 encoding scheme serializes a UTF-8 code unit sequence in exactly the same order as the code unit sequence itself.

When represented in UTF-8, the signature turns into the octet sequence <EF BB BF>. Its usage at the beginning of a UTF-8 data stream is neither required or recommended but does not affect conformance.

11.3 UTF-16BE

The UTF-16BE encoding scheme serializes a UTF-16 code unit sequence by ordering the octets for each code unit such that the more significant octet precedes the less significant octet (also known as big-endian ordering).

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 10646-2020

In UTF-16BE, an initial octet sequence of <FE FF> is interpreted as FEFF ZERO WIDTH NO-BREAK SPACE and does not convey a signature meaning.

NOTE — Because the code point FFFE is a noncharacter code point (see 7.3.7), the octet sequence <FF FE> is not valid in the UTF-16BE encoding scheme. If a data stream is assumed to be using the UTF-16BE encoding scheme, an initial octet sequence <FF FE> would serve as a signature strongly suggesting that the assumption of UTF-16BE is invalid. If a data stream is assumed to be using the UTF-16 encoding form but the encoding scheme has not been specified, initial octet sequences <FE FF> or <FF FE> can be used as a heuristic: <FE FF> would strongly suggest that UTF-16BE can be assumed, while <FF FE> would strongly suggest that UTF-16BE should not be assumed. See also 11.5.

11.4 UTF-16LE

The UTF-16LE encoding scheme serializes a UTF-16 code unit sequence by ordering the octets for each code unit such that the less significant octet precedes the more significant octet (also known as little-endian ordering).

In UTF-16LE, an initial octet sequence of <FF FE> is interpreted as FEFF ZERO WIDTH NO-BREAK SPACE and does not convey a signature meaning.

NOTE — Because the code point FFFE is a noncharacter code point (see 7.3.7), the octet sequence <FE FF> is not valid in the UTF-16LE encoding scheme. If a data stream is assumed to be using the UTF-16LE encoding scheme, an initial octet sequence <FE FF> would serve as a signature strongly suggesting that the assumption of UTF-16LE is invalid. If a data stream is assumed to be using the UTF-16 encoding form but the encoding scheme has not been specified, initial octet sequences <FF FE> or <FE FF> can be used as a heuristic: <FF FE> would strongly suggest that UTF-16LE can be assumed, while <FE FF> would strongly suggest that UTF-16LE should not be assumed. See also 11.5.

11.5 UTF-16

The UTF-16 encoding scheme serializes a UTF-16 code unit sequence by ordering octets in a way that either the less significant octet precedes or follows the more significant octet.

In the UTF-16 encoding scheme, the initial signature read as <FE FF> indicates that the more significant octet precedes the less significant octet, and <FF FE> the reverse. The signature is not part of the textual data.

In the absence of signature, the octet order of the UTF-16 encoding scheme is that the more significant octet precedes the less significant octet.

11.6 UTF-32BE

The UTF-32BE encoding scheme serializes a UTF-32 code unit sequence by ordering the octets for each code unit such that the more significant octets precede the less significant octets (also known as big-endian ordering).

In UTF-32BE, an initial octet sequence of <00 00 FE FF> is interpreted as FEFF ZERO WIDTH NO-BREAK SPACE and does not convey a signature meaning.

NOTE — Because the code point FFFE is a noncharacter code point (see 7.3.7), the octet sequence <00 00 FF FE> is not valid in the UTF-32BE encoding scheme. If a data stream is assumed to be using the UTF-32BE encoding scheme, an initial octet sequence <00 00 FF FE> would serve as a signature strongly suggesting that the assumption of UTF-32BE is invalid. If a data stream is assumed to be using the UTF-32 encoding form but the encoding scheme has not been specified, initial octet sequences <00 00 FE FF> or <00 FF FE> can be used as a heuristic: <00 00 FE FF> would strongly suggest that UTF-32BE can be assumed, while <00 00 FF FE> would strongly suggest that UTF-32BE should not be assumed. See also 11.8.

11.7 UTF-32LE

The UTF-32LE encoding scheme serializes a UTF-32 code unit sequence by ordering the octets for each code unit such that the less significant octets precede the more significant octets (also known as little-endian ordering).

In UTF-32LE, an initial octet sequence of <FF FE 00 00> is interpreted as FEFF ZERO WIDTH NO-BREAK SPACE and does not convey a signature meaning.

NOTE — Because the code point FFFE is a noncharacter code point (see 7.3.7), the octet sequence <00 00 FE FF> is not valid in the UTF-32LE encoding scheme. If a data stream is assumed to be using the UTF-32LE encoding scheme, an initial octet sequence <00 00 FE FF> would serve as a signature strongly suggesting that the assumption of UTF-32LE is invalid. If a data stream is assumed to be using the UTF-32 encoding form but the encoding scheme has not been specified, initial octet sequences <00 00 FF FE> or <00 00 FE

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO IEC 10646-2020

FF> can be used as a heuristic: <00 00 FF FE> would strongly suggest that UTF-32LE can be assumed, while <00 00 FE FF> would strongly suggest that UTF-32LE should not be assumed. See also 11.8.

11.8 UTF-32

The UTF-32 encoding scheme serializes a UTF-32 code unit sequence by ordering octets in a way that either the less significant octets precede or follow the more significant octets.

In the UTF-32 encoding scheme, the initial signature read as <00 00 FE FF> indicates that the more significant octet precedes the less significant octet, and <FF FE 00 00> the reverse. The signature is not part of the textual data.

In the absence of signature, the octet order of the UTF-32 encoding scheme is that the more significant octets precede the less significant octets.

12 Use of control functions with the UCS

This coded character set provides for use of control functions encoded according to ISO/IEC 6429 or similarly structured standards for control functions, and standards derived from these. A set or subset of such coded control functions may be used in conjunction with this coded character set. These standards encode a control function as a sequence of one or more octets.

When a control character of ISO/IEC 6429 is used with this coded character set, its coded representation as specified in ISO/IEC 6429 shall be padded to correspond with the number of octets in code unit of the adopted encoded form (see Clause 10). Thus, the least significant octet shall be the bit combination specified in ISO/IEC 6429, and the more significant octet(s) shall be zeros.

For example, the control character FORM FEED is represented by “000C” in the UTF-16 encoding form, and “0000 000C” in the UTF-32 encoding form.

For escape sequences, control sequences, and control strings (see ISO/IEC 6429) consisting of a coded control character followed by additional bit combinations in the range 20 to 7F, each bit combination shall be padded by octet(s) with value 00.

For example, the escape sequence “ESC 02/00 04/00” is represented by “1B 20 40” in the UTF-8 encoding form, by “001B 0020 0040” in the UTF-16 encoding form, and “0000001B 00000020 00000040” in the UTF-32 encoding form.

NOTE 1 – The notation aa/bb (where ‘a’ and ‘b’ represent decimal digits 0 to 9) is used by ISO/IEC 2022 and ISO/IEC 6429 to indicate octet value where ‘aa’ can take values from 02 to 07 and ‘bb’ can take values from 00 to 15; for example, 02/00 represents the value 20 in hexadecimal notation and 07/15 represents the value 7E in hexadecimal notation.

NOTE 2 – The term “character” appears in the definition of many of the control functions specified in ISO/IEC 6429, to identify the elements on which the control functions will act. When such control functions are applied to coded characters according to this document, the action of those control functions will depend on the type of element from this document that has been chosen, by the application, to be the element (or character) on which the control functions act. These elements may be chosen to be characters (non-combining characters and/or combining characters) or may be chosen in other ways (such as composite sequences) when applicable.

Code extension control functions for the ISO/IEC 2022 code extension techniques (such as designation escape sequences, single shift, and locking shift) shall not be used with this coded character set.

NOTE 3 – The following list provides the long names from ISO/IEC 6429 used in association with the control characters.

0000 NULL
0001 START OF HEADING
0002 START OF TEXT
0003 END OF TEXT
0004 END OF TRANSMISSION
0005 ENQUIRY
0006 ACKNOWLEDGE
0007 BELL
0008 BACKSPACE
0009 CHARACTER TABULATION
000A LINE FEED
000B LINE TABULATION
000C FORM FEED
000D CARRIAGE RETURN
000E SHIFT-OUT
000F SHIFT-IN
0010 DATA LINK ESCAPE
0011 DEVICE CONTROL ONE
0012 DEVICE CONTROL TWO
0013 DEVICE CONTROL THREE
0014 DEVICE CONTROL FOUR
0015 NEGATIVE ACKNOWLEDGE
0016 SYNCHRONOUS IDLE
0017 END OF TRANSMISSION BLOCK

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 10646-2020

0018 CANCEL
0019 END OF MEDIUM
001A SUBSTITUTE
001B ESCAPE
001C INFORMATION SEPARATOR FOUR
001D INFORMATION SEPARATOR THREE
001E INFORMATION SEPARATOR TWO
001F INFORMATION SEPARATOR ONE
007F DELETE
0082 BREAK PERMITTED HERE
0083 NO BREAK HERE
0084 INDEX
0085 NEXT LINE
0086 START OF SELECTED AREA
0087 END OF SELECTED AREA
0088 CHARACTER TABULATION SET
0089 CHARACTER TABULATION WITH JUSTIFICATION
008A LINE TABULATION SET
008B PARTIAL LINE FORWARD
008C PARTIAL LINE BACKWARD
008D REVERSE LINE FEED
008E SINGLE-SHIFT TWO
008F SINGLE-SHIFT THREE
0090 DEVICE CONTROL STRING
0091 PRIVATE USE ONE
0092 PRIVATE USE TWO
0093 SET TRANSMIT STATE
0094 CANCEL CHARACTER
0095 MESSAGE WAITING
0096 START OF GUARDED AREA
0097 END OF GUARDED AREA
0098 START OF STRING
009A SINGLE CHARACTER INTRODUCER
009B CONTROL SEQUENCE INTRODUCER
009C STRING TERMINATOR
009D OPERATING SYSTEM COMMAND
009E PRIVACY MESSAGE
009F APPLICATION PROGRAM COMMAND

The control character 0084 INDEX has been removed from ISO/IEC 6429. In addition, the control characters 000E and 000F are named SHIFT-OUT and SHIFT-IN respectively in 7-bit environment and LOCKING-SHIFT ONE and LOCKING-SHIFT ZERO respectively in 8-bit environment.

13 Declaration of identification of features

13.1 Purpose and context of identification

Code unit sequences conforming to this document are intended to form all or part of a composite unit of coded information that is interchanged between an originator and a recipient. The identification of ISO/IEC 10646 (including the encoding form and the encoding scheme) and any subset of the coding space that have been adopted by the originator should also be available to the recipient. The route by which such identification is communicated to the recipient is outside the scope of this document.

However, some standards for interchange of coded information may permit, or require, that the coded representation of the identification applicable to the code unit sequence forms a part of the interchanged information. Clause 13 specifies a coded representation for the identification of UCS and a subset of ISO/IEC 10646, and of a C0 and a C1 set of control functions from ISO/IEC 6429 for use in conjunction with ISO/IEC 10646. Such coded representations provide all or part of an identification data element, which may be included in information interchange in accordance with the relevant standard.

In the context of these identifications, because the more significant octets shall precede the less significant octets when serialized, the only encoding schemes that can be selected are UTF-8, UTF-16BE, and UTF-32BE according to the relevant encoding forms (UTF-8, UTF-16, and UTF-32 respectively).

If two or more of the identifications are present, the order of those identifications shall follow the sub-clause order as specified in Clause 13.

NOTE 1 – For example, the identification of the encoding scheme should precede the identification of the subset of graphic characters.

NOTE 2 – An alternative method of identification is described in Annex N.

13.2 Identification of a UCS encoding scheme

When the escape sequences from ISO/IEC 2022 are used, the identification of a UCS encoding scheme (see Clause 11) specified by this document shall be by a designation sequence chosen from the following list:

ESC 02/05 02/15 04/09
UTF-8 encoding form; UTF-8 encoding scheme

ESC 02/05 02/15 04/12
UTF-16 encoding form; UTF-16BE encoding scheme

ESC 02/05 02/15 04/06
UTF-32 encoding form; UTF-32BE encoding scheme

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 10646-2020

NOTE – The following designation sequences: ESC 02/05 02/15 04/00, ESC 02/05 02/15 04/01, ESC 02/05 02/15 04/03, ESC 02/05 02/15 04/04, ESC 02/05 02/15 04/07, ESC 02/05 02/15 04/08, ESC 02/05 02/15 04/10, ESC 02/05 02/15 04/11 used in previous versions of this standard to identify implementation levels 1 and 2 are deprecated. The remaining designation sequences correspond to the former level 3 which is now the only supported content definition for code unit sequences.

ESC 02/05 04/07
UTF-8 encoding form; UTF-8 encoding scheme

If such an escape sequence appears within a code unit sequence conforming to ISO/IEC 2022, it shall consist only of the sequences of bit combinations as shown above.

If such an escape sequence appears within a code unit sequence conforming to this document, it shall be padded in accordance with Clause 12 when the identified encoding form is either UTF-16 or UTF-32. No padding is necessary when the identified encoding form is UTF-8. See also 13.5.

13.3 Identification of subsets of graphic characters

When the control sequences of ISO/IEC 6429 are used, the identification of subsets (see Clause 9) specified by this document shall be by a control sequence IDENTIFY UNIVERSAL CHARACTER SUBSET (IUCS) as shown below.

CSI Ps... 02/00 06/13

NOTE – CSI is the short name for the control function CONTROL SEQUENCE INTRODUCER in ISO/IEC 6429.

Ps... means that there can be any number of selective parameters. The parameters are to be taken from the subset collection numbers as shown in Annex A. When there is more than one parameter, each parameter value is separated by an octet with value 03/11.

Parameter values are represented by digits where octet values 03/00 to 03/09 represent digits 0 to 9.

If such an escape sequence appears within a code unit sequence conforming to ISO/IEC 2022, it shall consist only of the sequences of bit combinations as shown above.

If such a control sequence appears within a code unit sequence conforming to this document, it shall be padded in accordance with Clause 12.

13.4 Identification of control function set

When the escape sequences from ISO/IEC 2022 are used, the identification of each set of control functions (see Clause 12) of ISO/IEC 6429 to be used in conjunction with ISO/IEC 10646 shall be an identifier sequence of the type shown below.

ESC 02/01 04/00	identifies the full C0 set of ISO/IEC 6429
ESC 02/02 04/03	identifies the full C1 set of ISO/IEC 6429

For other C0 or C1 sets, the final octet F shall be obtained from the International Register of Coded Character Sets. The identifier sequences for these sets shall be

ESC 02/01 F	identifies a C0 set
ESC 02/02 F	identifies a C1 set

If such an escape sequence appears within a code unit sequence conforming to ISO/IEC 2022, it shall consist only of the sequences of bit combinations as shown above.

If such an escape sequence appears within a code unit sequence conforming to this document, it shall be padded in accordance with Clause 12.

13.5 Identification of the coding system of ISO/IEC 2022

When the escape sequences from ISO/IEC 2022 are used, the identification of a return, or transfer, from UCS to the coding system of ISO/IEC 2022 shall be by the escape sequence ESC 02/05 04/00. If such an escape sequence appears within a code unit sequence conforming to this document, it shall be padded in accordance with Clause 12.

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 10646-2020

If such an escape sequence appears within a code unit sequence conforming to ISO/IEC 2022, it shall consist only of the sequence of bit combinations as shown above.

NOTE – Escape sequence ESC 02/05 04/00 is normally used for return to the restored state of ISO/IEC 2022. The escape sequence ESC 02/05 04/00 specified here is sometimes not exactly as specified in ISO/IEC 2022 due to the presence of padding octets. For this reason the escape sequences used in 13.2 for the identification of UCS (except for ESC 02/05 04/07) include the octet 02/15 to indicate that the return does not always conform to that standard.

14 Structure of the code charts and lists

Clause 34 sets out the detailed code charts and the lists of character names for the graphic characters. It specifies for each character their graphic symbol, their coded representation, and their character name.

NOTE – Clause 34 also includes additional information on characters clarifying some feature of a character, such as its naming or usage, or its associated graphic symbol.

Graphic symbols are to be regarded as typical visual representations of the corresponding graphic characters. This document does not attempt to prescribe the exact shape or colour of each character. The shape is affected by the design of the font or other representation method employed, which is out of scope. Although the representative glyphs in this document are consistently presented in black and white, it does not prevent implementations from using graphic symbols with some specific colour or even with multiple colours, fully or partly animated graphics, or both. When characters are typically associated with a particular colour, conventions of European heraldry are used to represent those colours in monochromatic line drawings in the code charts. Furthermore, the usage of 'BLACK' and 'WHITE' in character names does not imply a specific colour. It is simply a distinction between a filled character and an outline character.

Graphic characters specified in this document are uniquely identified by their names. This does not imply that the graphic symbols by which they are commonly imaged are always different. Examples of graphic characters with similar graphic symbols are LATIN CAPITAL LETTER A, GREEK CAPITAL LETTER ALPHA and CYRILLIC CAPITAL LETTER A.

The meaning attributed to any character is not specified by this document; it may differ from country to country, or from one application to another.

For the alphabetic scripts, the general principle has been to arrange the characters within any row in approximate alphabetic sequence; where the script has capital and small letters, these are arranged in pairs. However, this general principle has been overridden in some cases. For example, for those scripts for which a relevant standard exists, the characters are allocated according to that standard. This arrangement within the code charts will aid conversion between the existing standards and this coded character set. In general, however, it is anticipated that conversion between this coded character set and any other coded character set will use a table lookup technique.

It is not intended, nor will it often be the case, that the characters needed by any one user will be found all grouped together in one part of the code charts.

Furthermore, the user of any script will find that needed characters may have been coded elsewhere in this coded character set. This especially applies to the digits, to the symbols, and to the use of Latin letters in dual-script applications.

Therefore, in using this coded character set, the reader is advised to refer first to the block names list in Annex A.2 or an overview of the Planes in figures 8 to 14, and then to turn to the specific code chart for the relevant script and for symbols and digits. In addition, Annex G contains an alphabetically sorted list of character names.

15 Block and collection names

15.1 Block names

Named blocks of contiguous code points are specified within a plane for the purpose of allocation of characters sharing some common characteristic, such as script. The blocks specified within the BMP, SMP, SIP and SSP are listed in A.2 , and are illustrated in figures 8 to 14.

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO IEC 10646-2020

Rules to be used for constructing the names of blocks are given in 27.5.1.

15.2 Collection names

Collections are shown in Annex A.

Rules to be used for constructing the names of collections are given in 27.5.2.

16 Mirrored characters in bidirectional context

16.1 Mirrored characters

A class of characters has special significance in the context of bidirectional text. The interpretation and rendering of any of these characters depend on the direction of the character being rendered that is in effect at the point in the code unit sequence where the coded representation of the character appears. The list of these characters is determined by having the 'Bidi_Mirrored' property set to 'Y' in the Unicode Standard. These values shall be determined according to the Unicode Standard Bidi Mirrored property (see Clause 2).

NOTE 1 – Typically, a mirrored character has its image mirrored horizontally in text that is laid out from right to left. However, for some mathematical symbols, the 'mirrored' form is not an exact mirror image. See the Unicode Technical Report #25, "Unicode Support for Mathematics" for additional details.

This character mirroring is not limited to paired characters and shall be applied to all characters belonging to that class.

EXAMPLE

In a right-to-left text segment, the GREATER-THAN SIGN (rendered as ">" in left-to-right text) may be rendered as the "<" graphic symbol.

NOTE 2 – Many ancient scripts and some scripts in modern use can be written either right-to-left or left-to-right. It is often customary for one of these scripts to use the appropriately mirrored graphical symbol for any character represented by a graphic symbol that is not symmetric around the vertical axis. In such cases, it is up to the rendering system to display the graphic image appropriate for the writing direction employed. The directionality of the representative graphic symbol shown in the character code charts matches the default writing direction for the script. Characters belonging to these scripts have the "Bidi_Mirrored' property set to 'N' in the Unicode Standard (see reference to the Unicode Standard Bidi Mirrored property in Clause 2).

Examples of such scripts include, but are not limited to, Old Italic, an ancient script for which the default writing direction in this standard is left-to-right, and Cypriot, an ancient script for which the default writing direction in this standard is right-to-left.

16.2 Directionality of bidirectional text

The Unicode Bidirectional Algorithm (see Clause 2) describes the algorithm used to determine the directionality for bidirectional text. It shall be used in the context of this document.

17 Special characters

17.1 General

There are some characters that do not have printable graphic symbols or are otherwise special in some ways.

17.2 Space characters

The following characters are space characters. They represent all characters which have the General Category value set to 'Zs'.

Code Point	Name		
0020	SPACE	2005	FOUR-PER-EM SPACE
00A0	NO-BREAK SPACE	2006	SIX-PER-EM SPACE
1680	OGHAM SPACE MARK	2007	FIGURE SPACE
2000	EN QUAD	2008	PUNCTUATION SPACE
2001	EM QUAD	2009	THIN SPACE
2002	EN SPACE	200A	HAIR SPACE
2003	EM SPACE	202F	NARROW NO-BREAK SPACE
2004	THREE-PER-EM SPACE	205F	MEDIUM MATHEMATICAL SPACE
		3000	IDEOGRAPHIC SPACE

NOTE 1 – The character 1680 OGHAM SPACE MARK is typically represented with a visible glyph showing the central stemline, and it is only represented by a blank glyph in a "stemless" style font.

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO IEC 10646-2020

NOTE 2 – The character 202F NARROW NO-BREAK-SPACE is a non-breaking space. It is similar to 00A0 NO-BREAK SPACE, except that it is rendered with a narrower width. When used with the Mongolian script this character is usually rendered at one-third of the width of a normal space, and it separates a suffix from the Mongolian word-stem. This allows for the normal rules of Mongolian character shaping to apply, while indicating that there is no word boundary at that position.

17.3 Currency symbols

Currency symbols in this document do not necessarily identify the currency of a country. For example, YEN SIGN can be used for Japanese Yen and Chinese Yuan. Also, DOLLAR SIGN is used in numerous countries including the United States of America.

17.4 Format characters

The following characters are format characters (see 7.3.3). They represent all characters which have the General Category value set to 'Cf', 'Zl', and 'Zp'. See Annex F.

Code Point	Name
00AD	SOFT HYPHEN
0600	ARABIC NUMBER SIGN
0601	ARABIC SIGN SANAH
0602	ARABIC FOOTNOTE MARKER
0603	ARABIC SIGN SAFHA
0604	ARABIC SIGN SAMVAT
0605	ARABIC NUMBER MARK ABOVE
061C	ARABIC LETTER MARK
06DD	ARABIC END OF AYAH
070F	SYRIAC ABBREVIATION MARK
08E2	ARABIC DISPUTED END OF AYAH
180E	MONGOLIAN VOWEL SEPARATOR
200B	ZERO WIDTH SPACE
200C	ZERO WIDTH NON-JOINER
200D	ZERO WIDTH JOINER
200E	LEFT-TO-RIGHT MARK
200F	RIGHT-TO-LEFT MARK
2028	LINE SEPARATOR
2029	PARAGRAPH SEPARATOR
202A	LEFT-TO-RIGHT EMBEDDING
202B	RIGHT-TO-LEFT EMBEDDING
202C	POP DIRECTIONAL FORMATTING
202D	LEFT-TO-RIGHT OVERRIDE
202E	RIGHT-TO-LEFT OVERRIDE
2060	WORD JOINER
2061	FUNCTION APPLICATION
2062	INVISIBLE TIMES
2063	INVISIBLE SEPARATOR
2064	INVISIBLE PLUS
2066	LEFT-TO-RIGHT ISOLATE
2067	RIGHT-TO-LEFT ISOLATE
2068	FIRST STRONG ISOLATE
2069	POP DIRECTIONAL ISOLATE
206A	INHIBIT SYMMETRIC SWAPPING
206B	ACTIVATE SYMMETRIC SWAPPING
206C	INHIBIT ARABIC FORM SHAPING
206D	ACTIVATE ARABIC FORM SHAPING
206E	NATIONAL DIGIT SHAPES
206F	NOMINAL DIGIT SHAPES
FEFF	ZERO WIDTH NO-BREAK SPACE
FFF9	INTERLINEAR ANNOTATION ANCHOR
FFFA	INTERLINEAR ANNOTATION SEPARATOR
FFFB	INTERLINEAR ANNOTATION TERMINATOR
110BD	KAITHI NUMBER SIGN
110CD	KAITHI NUMBER SIGN ABOVE
13430	EGYPTIAN HIEROGLYPH VERTICAL JOINER
13431	EGYPTIAN HIEROGLYPH HORIZONTAL JOINER
13432	EGYPTIAN HIEROGLYPH INSERT AT TOP START
13433	EGYPTIAN HIEROGLYPH INSERT AT BOTTOM START
13434	EGYPTIAN HIEROGLYPH INSERT AT TOP END
13435	EGYPTIAN HIEROGLYPH INSERT AT BOTTOM END
13436	EGYPTIAN HIEROGLYPH OVERLAY MIDDLE
13437	EGYPTIAN HIEROGLYPH BEGIN SEGMENT
13438	EGYPTIAN HIEROGLYPH END SEGMENT
1BCA0	SHORTHAND FORMAT LETTER OVERLAP
1BCA1	SHORTHAND FORMAT CONTINUING OVERLAP
1BCA2	SHORTHAND FORMAT DOWN STEP
1BCA3	SHORTHAND FORMAT UP STEP
1D173	MUSICAL SYMBOL BEGIN BEAM
1D174	MUSICAL SYMBOL END BEAM
1D175	MUSICAL SYMBOL BEGIN TIE
1D176	MUSICAL SYMBOL END TIE
1D177	MUSICAL SYMBOL BEGIN SLUR
1D178	MUSICAL SYMBOL END SLUR
1D179	MUSICAL SYMBOL BEGIN PHRASE
1D17A	MUSICAL SYMBOL END PHRASE
E0001	LANGUAGE TAG
E0020-E007F	TAG SPACE to CANCEL TAG

17.5 Ideographic description characters

An Ideographic Description Character (IDC) is a graphic character, which is used with a sequence of other graphic characters to form an Ideographic Description Sequence (IDS). Such a sequence may be used to describe an ideographic character which is not specified with this document. Annex I describes them in more details. The list of IDC follows:

Code Point	Name
2FF0	IDEOGRAPHIC DESCRIPTION CHARACTER LEFT TO RIGHT
2FF1	IDEOGRAPHIC DESCRIPTION CHARACTER ABOVE TO BELOW
2FF2	IDEOGRAPHIC DESCRIPTION CHARACTER LEFT TO MIDDLE AND RIGHT
2FF3	IDEOGRAPHIC DESCRIPTION CHARACTER ABOVE TO MIDDLE AND BELOW
2FF4	IDEOGRAPHIC DESCRIPTION CHARACTER FULL SURROUND

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO IEC 10646-2020

2FF5 IDEOGRAPHIC DESCRIPTION CHARACTER SURROUND FROM ABOVE
2FF6 IDEOGRAPHIC DESCRIPTION CHARACTER SURROUND FROM BELOW
2FF7 IDEOGRAPHIC DESCRIPTION CHARACTER SURROUND FROM LEFT
2FF8 IDEOGRAPHIC DESCRIPTION CHARACTER SURROUND FROM UPPER LEFT
2FF9 IDEOGRAPHIC DESCRIPTION CHARACTER SURROUND FROM UPPER RIGHT
2FFA IDEOGRAPHIC DESCRIPTION CHARACTER SURROUND FROM LOWER LEFT
2FFB IDEOGRAPHIC DESCRIPTION CHARACTER OVERLAID

17.6 Variation selectors and variation sequences

17.6.1 General

Variation selectors are a specific class of combining characters immediately following a base character or a spacing mark for which there is no canonical mapping and no equivalent composite sequence. The purpose is to indicate a specific variant form of graphic symbol for that base character or spacing mark. The character sequence consisting of such base character or spacing mark followed by a variation selector is called a variation sequence.

NOTE 1 – The variation selector only selects a specific *appearance* among those acceptable for an already encoded character. It is not intended as a general code extension mechanism.

Variation selectors are made of all coded characters included in the blocks VARIATION SELECTORS and VARIATION SELECTORS SUPPLEMENT and the three Mongolian Free Variation Selectors (FVS1 to FVS3).

A variation sequence whose base character is a pictographic symbol and whose variation selector is FE0E VARIATION SELECTOR-15 or FE0F VARIATION SELECTOR-16 is called an Emoji variation sequence.

A variation sequence whose base character is an ideograph which is not canonically or compatibly decomposable and whose variation selector is from the VARIATION SELECTORS SUPPLEMENT block is called an ideographic variation sequence. All other variation sequences are called standardized variation sequences. The variant form of a graphic symbol specified by a standardized variation sequence is called a standardized variant.

NOTE 2 – At present the only specified ideographic variation sequences are for CJK unified ideographs.

Only the variation sequences defined or referenced in 17.6 indicate a specific variant form of graphic symbol; all other such sequences are undefined. Furthermore, variation selectors following other base characters and any non-base characters have no effect on the selection of the graphic symbol for that character.

17.6.2 Standardized variation sequences

Standardized variation sequences are defined by a machine-readable format that is accessible as a link: https://standards.iso.org/iso-iec/10646/ed-6/en/UCSVariants.txt.

The content linked to is a plain text file, encoded as UTF-8 with CARRIAGE RETURN/ LINE FEED as end of line mark that specifies Standardized Variations Sequences. Each line in the text file contains the following information organized in two or three fields:

— 1st field: Variation sequence expressed as a UCS Sequence Identifier (using a modified syntax for the USI, omitting commas and angle brackets),
— 2nd field: Description of the variation sequence,
— 3rd field (optional): Shaping environments where the variation sequence applies. The possible values are: isolate, initial, medial, final.

The fields are delimited by a SEMICOLON (';') followed optionally by zero or more SPACE characters. The last field may be followed by a comment starting with a NUMBER SIGN ('#') describing the name of the base character. Comment lines, starting with a NUMBER SIGN ('#') are informational only. Comment lines and blank lines in the text file should be ignored by any automatic process which parses the data file to extract the normative list of Standardized Variants.

The Standardized Variations Sequences contain sequences associated with allowed base characters from the following categories:

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 10646-2020

— Mathematical symbols

NOTE 1 – The VARIATION SELECTOR-1 (FE00) is the only variation selector used with mathematical symbols.

— Mongolian characters. Only some presentation forms of the base Mongolian characters used with the Mongolian free variation selectors produce variant appearances.

NOTE 2 – The Mongolian characters have various presentation forms depending on their position in a code unit sequence. These presentations forms are called isolate, initial, medial and final. The reference for Mongolian variant information for implementers is provided in https://www.unicode.org/reports/tr54/.

— Manichaean characters

NOTE 3 – These are similar to Mongolian variations sequences in depending on their position in a code unit sequence but using only isolate and final presentation forms.

— Myanmar characters. These variation sequences allow the representation of the dotted glyph used in Khamti, Aiton, and Phake style consonants and vowel signs.

— Phags-pa characters. These variation sequences do not select fixed visual representation; rather, they select a representation that is reversed from the normal form predicted by the preceding character.

— CJK Unified ideographs. Each of these variation sequences corresponds to a CJK Compatibility ideograph. Its specified appearance is that of the corresponding CJK Compatibility ideograph.

NOTE 4 – All normalization forms replace CJK Compatibility ideographs with the corresponding CJK Unified ideographs, but leave the variation sequences unchanged (see Clause 22). In contexts where normalization forms are used and the distinction between the CJK Compatibility ideographs and CJK Unified ideographs is desired, the usage of variation sequences is a mechanism to maintain that distinction. No equivalence between these variation sequences and the corresponding compatibility ideographs is defined. Conversion considerations are out of scope of this document

17.6.3 Emoji variation sequences

The Emoji Variations Sequences contain sequences where the base characters are pictographic symbols followed by specific variation selectors. The range of presentations may include a traditional black and white text style, using FE0E VARIATION SELECTOR-15, or an ‘emoji’ style, using FE0F VARIATION SELECTOR-16, whose presentation often involves colour/grayscale and/or animation.

NOTE – Standardized variations sequences involving the characters 0023 NUMBER SIGN and the range 0030-0039 (DIGIT ZERO to DIGIT NINE) are intended for use with 20E3 COMBINING ENCLOSING KEYCAP, such as in:

<0023, FE0E, 20E3> NUMBER SIGN inside a COMBINING ENCLOSING KEYCAP in text style.

Emoji variations sequences are defined by a machine-readable format that is accessible as a link:

https://standards.iso.org/iso-iec/10646/ed-6/en/emoji-variation-sequences.txt.

The content linked to is a plain text file, encoded as UTF-8 with CARRIAGE RETURN/ LINE FEED as end of line mark that specifies Emoji Variations Sequences. Each line in the text file contains the following information organized in two or three fields:

— 1st field: Variation sequence expressed as a UCS Sequence Identifier (using a modified syntax for the USI, omitting commas and angle brackets),

— 2nd field: Description of the variation sequence,

The fields are delimited by a SEMICOLON (';') followed optionally by zero or more SPACE characters. The last field may be followed by a comment starting with a NUMBER SIGN ('#') including the name of the base character. Comment lines, starting with a NUMBER SIGN ('#') are informational only. Comment lines and blank lines in the text file should be ignored by any automatic process which parses the data file to extract the normative list of Standardized Variants.

17.6.4 Ideographic variation sequences

Variations sequences composed of an ideograph as the base character and one of VARIATION SELECTOR-17 to VARIATION SELECTOR-256 from the Supplementary Special-purpose Plane (SSP) are registered in the Ideographic Variation Database defined by Unicode Technical Standard #37. The version referenced in Clause 2 shall be used in the context of this document. The purpose of IVD registration is to provide a technical solution to

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO IEC 10646-2020

represent ideographic variants that are considered unifiable (see Annex S for the unification rules for the CJK Unified ideographs).

NOTE – This document incorporates by reference the variation sequences listed in version 2017-12-12 of the Ideographic Variation Database, as described at http://www.unicode.org/ivd/data/2017-12-12/.

18 Presentation forms of characters

Each presentation form of a character provides an alternative form, for use in a particular context, to the nominal form of the character or sequence of characters from the other zones of graphic characters. The transformation from the nominal form to the presentation forms may involve substitution, superimposition, or combination.

The rules for the superimposition, choice of differently shaped characters, or combination into ligatures, or conjuncts, which are often of extreme complexity, are not specified in this document.

In general, presentation forms are not intended to be used as a substitute for the nominal forms of the graphic characters specified elsewhere within this coded character set. However, specific applications may encode these presentation forms instead of the nominal forms for specific reasons among which is compatibility with existing devices. The rules for searching, sorting, and other processing operations on presentation forms are outside the scope of this document.

Within the BMP these characters are mostly allocated to code points within rows from FB to FF.

19 Compatibility characters

Compatibility characters are included in this document primarily for compatibility with existing coded character sets to allow two-way code conversion without loss of information.

Within the BMP many of these characters are allocated to code points within rows F9, FA, FE, and FF, and within rows 31 and 33. Some compatibility characters are also allocated within other rows.

NOTE – There are twelve code points in the row FA of the BMP which are allocated to CJK Unified ideographs.

Within the Supplementary Ideographic Plane (SIP) these characters are allocated to code points within rows F8 to FA.

The CJK Compatibility ideographs are ideographs that should have been unified with one of the CJK Unified ideographs, per the unification rule described in Annex S. However, they are included in this document as separate characters, because, based on various national, cultural, or historical reasons for some specific country and region, some national and regional standards assign separate code points for them.

For this reason, compatibility ideographs should only be used for maintaining and guaranteeing a round trip conversion with the specific national, regional, or other standard. Other usage is strongly discouraged.

Because compatibility ideographs are not preserved through any normalization forms, use of standardized variation sequences for CJK Unified ideographs (see 17.6) may be preferred in contexts where normalization forms are used and the distinction between CJK Compatibility ideographs and the corresponding CJK Unified ideographs needs to be maintained. In context where compatibility ideographs should be preserved, normalization forms cannot be used.

20 Order of characters

Usually, coded characters appear in a code unit sequence in logical order (logical or backing store order corresponds approximately to the order in which characters are entered from the keyboard, after corrections such as insertions, deletions, and overtyping have taken place). This applies even when characters of different dominant direction are mixed: left-to-right (Greek, Latin, Thai) with right-to-left (Arabic, Hebrew), or with vertical (Mongolian) script.

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO IEC 10646-2020

Some characters may not appear linearly in final rendered text. For example, the medial form of DEVANAGARI VOWEL SIGN I is displayed before the character that it logically follows in the code unit sequence.

21 Combining characters

21.1 Order of combining characters

Coded representations of combining characters shall follow that of the graphic character with which they are associated (for example, coded representations of LATIN SMALL LETTER A followed by COMBINING TILDE represent a composite sequence for Latin “ã”).

If a combining character is to be regarded as a composite sequence in its own right, it shall be coded as a composite sequence by association with the character 00A0 NO-BREAK SPACE. For example, grave accent can be composed as 00A0 NO-BREAK SPACE followed by 0300 COMBINING GRAVE ACCENT.

Indic combining marks for vowels form a special category of combining characters, since the presentation can depend on more than one of the surrounding characters. Thus, it might not be desirable to associate these Indic combining marks with the character NO-BREAK SPACE.

21.2 Combining class and canonical ordering

Each combining character has a combining class value determined by the Unicode Standard. These values shall be determined according to the Unicode Standard Combining Class property (see Clause 2). The combining class is used to determine the canonical ordering which is part of the normalization process (see Clause 22). Canonical ordering consists in ordering combining characters in the increasing order of their combining class. Combining characters with a combining class value of zero are not re-ordered relative to other characters.

21.3 Appearance in code charts

Combining characters intended to be positioned relative to the associated character are depicted within the character code charts above, below, to the right of, to the left of, in, around, or through a dotted circle to show their position relative to the base character. In presentation, these characters are intended to be positioned relative to the preceding base character in some manner, and not to stand alone or function as base characters. This is the motivation for the term “combining”.

NOTE – Diacritics are the principal class of combining characters used in European alphabets. For many other scripts used in India and South East Asia, combining characters encode vowel letters; as such they are not generally referred to as “diacritical marks”.

21.4 Alternate coded representations

Alternate coded representations of text are generated by using multiple combining characters in different orders, or using various equivalent combinations of characters and composite sequences. These alternate coded representations result in multiple representations of the same text. Normalizing (see Clause 22) these coded representations reduces significantly, but does not eliminate, the occurrences of these multiple representations.

NOTE – For example, the French word “là” may be represented by the characters LATIN SMALL LETTER L followed by LATIN SMALL LETTER A WITH GRAVE, or may be represented by the characters LATIN SMALL LETTER L followed by LATIN SMALL LETTER A followed by COMBINING GRAVE ACCENT. When the normalization forms are applied on those alternate coded representations, only one representation remains. The form of the remaining representation depends on the normalization form used.

21.5 Multiple combining characters

There are instances where more than one combining character is applied to a single graphic character. This document does not restrict the number of combining characters that can follow a base character. The following rules apply to the presentation of these characters:

a) If the combining characters can interact in presentation (for example, COMBINING MACRON and COMBINING DIAERESIS), then the position of the combining characters in the resulting graphic display is determined by the order of the coded representation of the combining characters. The presentations of combining characters are to be positioned from the base character outward. For example, combining characters placed above a base character are stacked vertically, starting with the first encountered in the sequence of coded representations and continuing for as many marks above as are required by the coded combining characters following the coded base character. For combining characters placed below a base

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO IEC 10646-2020

character, the situation is inverted, with the combining characters starting from the base character and stacking downward.

An example of multiple combining characters above the base character is found in Thai, where a consonant letter can have above it one of the vowels 0E34 to 0E37 and, above that, one of four tone marks 0E48 to 0E4B. The order of the coded representation is: base consonant, followed by a vowel, followed by a tone mark.

b) Some specific combining characters override the default stacking behaviour by being positioned horizontally rather than stacking, or by forming a ligature with an adjacent combining character. When positioned horizontally, the order of coded representations is reflected by positioning in the dominant order of the script with which they are used. For example, horizontal accents in a left-to-right script are coded left-to-right.

   Prominent characters that show such override behaviour are associated with specific scripts or alphabets. For example, the COMBINING GREEK KORONIS (0343) requires that, together with a following acute or grave accent, they be rendered side-by-side above a letter, rather than the accent marks being stacked above the COMBINING GREEK KORONIS. The order of the coded representations is: the letter itself, followed by that of the breathing mark, followed by that of the accent marks. Two Vietnamese tone marks, which have the same graphic appearance as the Latin acute and grave accent marks, do not stack above the three Vietnamese vowel letters which already contain the circumflex diacritic (â, ê, ô). Instead, they form ligatures with the circumflex component of the vowel letters.

c) If the combining characters do not interact in presentation (for example, when one combining character is above a graphic character and another is below), the resultant graphic symbol from the base character and combining characters in different orders may appear the same. For example, the coded representations of LATIN SMALL LETTER A, followed by COMBINING CARON, followed by COMBINING OGONEK may result in the same graphic symbol as the coded representations of LATIN SMALL LETTER A, followed by COMBINING OGONEK, followed by COMBINING CARON.

   Combining characters in Hebrew or Arabic scripts do not normally interact. Therefore, the sequence of their coded representations in a composite sequence does not affect its graphic symbol. The rules for forming the combined graphic symbol are beyond the scope of this document.

21.6 Collections containing combining characters

In some collections of characters listed in Annex A, such as collections 14 (BASIC ARABIC) or 25 (THAI), both combining characters and non-combining characters are included.

Other collections of characters listed in Annex A comprise only combining characters, for example collection 7 (COMBINING DIACRITICAL MARKS).

21.7 Combining Grapheme Joiner

The character 034F COMBINING GRAPHEME JOINER is used to indicate that adjacent characters are to be treated as a unit for the purpose of language-sensitive collation and searching. In language-sensitive collation and searching, the combining grapheme joiner shall be ignored unless it specifically occurs with a tailored collation element mapping. For rendering, the combining grapheme joiner is invisible.

NOTE – The combining grapheme joiner may be used to differentiate two usages of a combining character by using it for one of the two cases. For example, where a distinction is needed between the German umlaut and the tréma, the COMBINING GRAPHEME JOINER (034F) followed by the COMBINING DIAERESIS (0308) should be used to represent the tréma while the COMBINING DIAERESIS (0308) alone should be used to represent the German umlaut.

22 Normalization forms

Normalization forms are the mechanisms allowing the selection of a unique coded representation among alternative; but equivalent coded text representations of the same text. Normalization forms for use with this document are specified in the Unicode Standard UAX#15 (see Clause 2) and shall be used in the context of this document. There are four normalization forms:

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO IEC 10646-2020

a) Normalization Form D (NFD),

b) Normalization Form C (NFC),

c) Normalization Form KD (NFKD),

d) Normalization Form KC (NFKC).

NOTE 1 – The result of applying any of these normalization forms onto a code unit sequence is intended to stay stable over time. It means that the normalized representation of a code unit sequence consisting of characters assigned in this version of the standard remains normalized even when the standard is amended.

NOTE 2 – Some normalization forms favour composite sequences over shorter representations of text, others favour the shorter representations. The backward compatibility requirement is provided by establishing ISO/IEC 10646-1:2000 (2nd Edition) and ISO/IEC 10646-2:2001 (1st Edition) as the reference versions for the definition of the shorter representation of text. The union of their repertoire is identical to the fixed collection UNICODE 3.2 (see A.6).

NOTE 3 – The goal of normalization is to provide a unique normalized result for any given code unit sequence to facilitate, among other things, identity matching. A normalized form does not necessarily represent the optimal sequence from a linguistic point of view.

NOTE 4 – In all four normalization forms, CJK Compatibility ideographs are replaced with the corresponding CJK Unified ideographs. Normalization, however, does not alter variation selectors, and variation sequences are preserved. Because of this, the use of standardized variation sequences for CJK Unified ideographs over the CJK Compatibility ideographs is preferred in the context of normalization (see 17.6).

23 Special features of individual scripts and symbol repertoires

23.1 Hangul syllable composition method

In rendering, a sequence of Hangul Jamo (from HANGUL JAMO block: 1100 to 11FF, HANGUL JAMO EXTENDED-A block: A960 to A97F, and HANGUL JAMO EXTENDED-B block: D7B0 to D7FF) is displayed as a series of syllable blocks. Jamo can be classified into three classes: Choseong (syllable-initial characters or initial consonants), Jungseong (syllable-peak characters or medial vowels), and Jongseong (syllable-final characters or final consonants). A complete syllable block is composed of a Choseong and a Jungseong, and optionally a Jongseong.

An incomplete syllable is a string of one or more characters which does not constitute a complete syllable (for example, a Choseong alone, a Jungseong alone, a Jongseong alone, or a Jungseong followed by a Jongseong). An incomplete syllable which starts with a Jungseong shall be preceded by a CHOSEONG FILLER (115F). An incomplete syllable composed of a Jongseong alone shall be preceded by a CHOSEONG FILLER (115F) and JUNGSEONG FILLER (1160). An incomplete syllable composed of a Choseong alone shall be followed by a JUNGSEONG FILLER (1160).

NOTE 1 – Hangul Jamo are not combining characters.

When a combining character such as HANGUL SINGLE DOT TONE MARK (302E) is intended to apply to a sequence of Hangul Jamo it should be placed at the end of the sequence, after the Hangul Jamo character which completes the syllable block.

NOTE 2 – Hangul text can be represented in several different ways in this standard. Korean Standard KS X 1026-1: In-formation Technology - Universal Multiple-Octet Coded Character set (UCS) - Hangul - Part 1, Hangul processing guide for information interchange, provides guidelines on how to ensure interoperability in information interchange.

23.2 Features of scripts used in India and some other South Asian countries

In the code charts for Rows 09 to 0D and 0F, and for the MYANMAR block in Row 10, of the BMP (see Clause 34) the graphic symbols shown for some characters appear to be formed as compounds of the graphic symbols for two other characters in the same table.

EXAMPLE 1 Row 09 Devanagari

The graphic symbol for 0906 DEVANAGARI LETTER AA appears as if it is constructed from the graphic symbols for 0905 DEVANAGARI LETTER A and 093E DEVANAGARI VOWEL SIGN AA

EXAMPLE 2 Row 0D Malayalam

The graphic symbol for 0D08 MALAYALAM LETTER II appears as if it is constructed from the graphic symbols for 0D07 MALAYALAM LETTER I and 0D57 MALAYALAM AU LENGTH MARK

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO IEC 10646-2020

In such cases a single coded character may appear to the user to be equivalent to the sequence of two coded characters whose graphic symbols, when combined, are visually similar to the graphic symbol of that single character, as in a composite sequence (see 3.17).

A "unique-spelling" rule is defined as follows. According to this rule, no coded character from a table for Rows 09 to 0D or 0F, or for the MYANMAR block in Row 10, with the list of exceptions mentioned below, shall be regarded as equivalent to a sequence of two or more other coded characters taken from the same table.

— Two-part dependent vowel signs,

— Independent vowel 1026 MYANMAR LETTER UU,

— Consonants including a nukta sign.

NOTE – All these characters have canonical mapping consisting of a sequence of two characters.

23.3 Byzantine musical symbols

The Byzantine Musical Notation System makes use of the so-called 'three-stripe' effect. There are signs that appear in the Upper, Middle or Lower stripes. Other signs are known as musical characters and appear in the textual part of the notation system. Multiple signs can be stacked together in their appropriate stripe.

23.4 Source references for pictographic symbols

Some symbols are associated by reference with various industrial sources. Unlike CJK Unified ideographs, these references do not establish character identity. These sources may be associated with either single code points or sequences of code points.

The symbol sources are:

— DoCoMo Shift-JIS code

— KDDI Shift-JIS code

— SoftBank Shift-JIS code

The source references are defined by a machine-readable format that is accessible as a link: https://standards.iso.org/iso-iec/10646/ed-6/en/EmojiSrc.txt.

The content linked to is a plain text file, encoded as UTF-8 with CARRIAGE RETURN/ LINE FEED as end of line mark, that specifies, after a 7-lines header and a variable number of comment lines (starting with '#'), source reference lines; each line containing the following information organized in fields delimited by ';':

— 1st field: UCS code point or sequence, in the format `(hhhh | hhhhh ) (<space> (hhhh | hhhhh))*`

— 2nd field: DoCoMo Shift-JIS code, in the format `(hhhh)`

— 3rd field: KDDI Shift-JIS code, in the format `(hhhh)`

— 4th field: SoftBank Shift-JIS code, in the format `(hhhh)`

The format definition uses `'h'` as a hexadecimal unit and `<space>` as the SPACE character. An ASTERISK indicates zero, one, or more iteration of the preceding pattern.

NOTE 1 – The content of the file is for reference only and is not intended for cross mapping between vendor sets.

NOTE 2 – This content provides mappings between UCS code points and sequences on one hand and Shift-JIS codes for cell phone carrier symbols on the other hand. Each mapping is symmetric ("round trip"), for equivalent UCS and carrier symbols or sequences. It does not include best-fit ("fallback") mappings to similar but not equivalent symbols in either mapping direction.

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO IEC 10646-2020

24 Source references for CJK ideographs

24.1 List of source references

A CJK ideograph is always referenced by at least one source reference. These source references are provided in a machine-readable format that is accessible as a link from this document. The content pointed by these links is also normative.

The source reference information establishes the character identity for CJK ideographs. A source reference is established by associating a CJK ideograph code point with one or several values in the source standards listed below. Such a source standard originates from the following categories:

— Hanzi G sources,
— Hanzi H sources,
— Hanzi M sources,
— Hanzi T sources,
— Kanji J sources,
— Hanja K sources,
— Hanja KP sources,
— ChuNom V sources,
— Unicode U sources,
— UK UK sources, and
— SAT S sources.

For a given code point, only one source reference can be created for each of the source standard category (G, H, M, T, J, K, KP, V, U, UK, and S). In order to provide a comprehensive coverage for a source standard category, when a source standard is referenced, all its unique associations with existing CJK ideographs are documented.

NOTE 1 – Some source references use a special type to denote self-referencing (i.e. the standard code point is the reference). These sources may correspond to initial sources that could not be confirmed, or to sources used in previous versions of this standard. The categories using this notation are 'G', 'H', 'K', 'KP', 'T', and 'V'. The corresponding sources are 'GU', 'HU', 'KU', 'KPU', 'TU', and 'VU'.

The following list identifies all sources referenced by the CJK ideographs in the BMP, the SIP, and the TIP.

The Hanzi G sources are

G0	GB2312-80
G1	GB12345-90
G3	GB7589-87 unsimplified forms
G5	GB7590-87 unsimplified forms
G7	General Purpose Hanzi List for Modern Chinese Language, and General List of Simplified Hanzi
GS	Singapore Characters
G8	GB8565-88
G9	GB18030-2000
GE	GB16500-95
GH	GB15564-1995 Code of Chinese Ideogram set for teltext broadcasting Hong Kong subset
GK	GB12052-89 Korean Character Coded Character Set for Information Interchange
G4K	Siku Quanshu（四庫全書）
GBK	Chinese Encyclopedia（中國大百科全書）
GCE	Chemistry Elements（化学元素中文用字）
GCH	Ci Hai（辞海）
GCY	Ci Yuan（辭源）
GCYY	Chinese Academy of Surveying and Mapping Ideographs（中国测绘科学研究院用字）

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 10646-2020

GDZ	Geographic Publishing House Ideographs（地质出版社用字）
GFC	Modern Chinese Standard Dictionary（现代汉语规范词典第二版。主编：李行健。北京：外语教学与研究出版社）2010, ISBN：978-7-5600-9518-9
GFZ	Founder Press System（方正排版系统）
GGFZ	Tongyong Guifan Hanzi Zidian（通用规范汉字字典）
GGH	Gudai Hanyu Cidian（古代汉语词典）
GHC	Hanyu Dacidian（漢語大詞典）
GHF	鄭賢章:《漢文佛典疑難俗字彙釋與研究》, 成都: 巴蜀書社, 2016, ISBN 978-7-5531-0700-4
GHZ	Hanyu Dazidian ideographs（漢語大字典）
GHZR	Hanyu Dazidian (2nd Edition)《漢語大字典》（第二版），2010 年，湖北长江出版集团・崇文书局、四川出版集团・四川辞书出版社，ISBN978-7-54-3-1744-7
GIDC	ID system of the Ministry of Public Security of China, 2009
GKJ	China National Committee for Terms in Sciences and Technologies (CNCTST), names of geology, botanical and zoology terms, WG2 N5072
GJZ	Commercial Press Ideographs（商务印书馆用字）
GKX	Kangxi Dictionary ideographs（康熙字典）9th edition (1958) including the addendum（康熙字典）補遺
GLGYJ	Zhuang Liao Songs Research《壮族嘹歌研究》2008 年广西民族出版社, ISBN 978-7-5363-5069-4
GLK	《龍龕手鑑》（續古逸叢書）
GOCD	Oxford English-Chinese Chinese-English Dictionary (牛津英汉汉英词典。主编：Julie Kleeman, 于海江。牛津：牛津大学出版社。2010 年。ISBN：978-0-19-920761-9)
GPGLG	Zhuang Folk Song Culture Series - Pingguo County Liao Songs（壮族民歌文化丛书•平果嘹歌）2004-2006, ISBN 7-5363-[4820-7 | 5012-0 | 5013-9 |5014-7 | 5015-5]
GRM	People's Daily Ideographs（人民日报用字）
GT	Standard Telegraph Codebook (revised), 1983（标准电码本（修订本））
GXC	Xiandai Hanyu Cidian（现代汉语词典）
GXH	Xinhua Zidian（新华字典）
GXHZ	Xinhua Big Dictionary（新华大字典：彩色本修订本。《新华字典》编委会编。北京：商务印书馆国际有限公司）2011, ISBN：978-7-80103-718-3
GWZ	Hanyu Dacidian Publishing House Ideographs（漢語大詞典出版社用字）
GZ	Ancient Zhuang Character Dictionary,（古壮字字典）1989, ISBN 7-5363-0614-8
GZFY	Hanyu Fangyan Dacidian（汉语方言大词典）
GZH	Zhonghua Zihai（中华字海）
GZJW	Yinzhou Jinwen Jicheng Yinde（殷周金文集成引得）
GZYS	Chinese Ancient Ethnic Characters Research（中国民族古文字研究）, 1984

NOTE 2 – The graphic symbol shown on the code charts for a character referenced by a Kangxi Dictionary (GKX) are in modern Chinese style which may differ slightly from the corresponding graphic symbol used in the dictionary.

The Hanzi H sources are

H	Hong Kong Supplementary Character Set – 2008
HB0	Big-5: Computer Chinese Glyph and Character Code Mapping Table, Technical Report C-26, 電腦用中文字型與字碼對照表, 技術通報 C-26, 1984, Symbols
HB1	Big-5, Level 1
HB2	Big-5, Level 2
HD	Hong Kong Supplementary Character Set – 2016

The Hanzi M source is

MAC	Macao Information System Character Set（澳門資訊系統字集）

The Hanzi T sources are

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 10646-2020

T1	TCA-CNS 11643-1992 1st plane
T2	TCA-CNS 11643-1992 2nd plane
T3	TCA-CNS 11643-1992 3rd plane with some additional characters
T4	TCA-CNS 11643-1992 4th plane
T5	TCA-CNS 11643-1992 5th plane
T6	TCA-CNS 11643-1992 6th plane
T7	TCA-CNS 11643-1992 7th plane
TA	TCA-CNS 11643-2007 10th plane
TB	TCA-CNS 11643-2007 11th plane
TC	TCA-CNS 11643-2007 12th plane
TD	TCA-CNS 11643-2007 13th plane
TE	TCA-CNS 11643-2007 14th plane
TF	TCA-CNS 11643-2007 15th plane
T13	TCA-CNS 11643-2007 19th plane

The Kanji J sources are

J0	JIS X 0208-1990
J1	JIS X 0212-1990
J3	JIS X 0213:2004 level-3
J3A	JIS X 0213:2004 level-3 addendum from JIS X 0213:2000 level-3
J13A	JIS X 0213:2004 level-3 addendum from JIS X 0213:2000 level-3 replacing J1 characters
J13	JIS X 0213:2004 level-3 characters replacing J1 characters
JA3	JIS X 0213:2004 level-3 characters replacing JA characters
J4	JIS X 0213:2004 level-4
J14	JIS X 0213:2004 level-4 characters replacing J1 characters
JA4	JIS X 0213:2004 level-4 characters replacing JA characters
JA	Unified Japanese IT Vendors Contemporary Ideographs, 1993
JARIB	Association of Radio Industries and Businesses (ARIB) ARIB STD-B24 Version 5.1, March 14 2007
JH	Hanyo-Denshi Program (汎用電子情報交換環境整備プログラム), 2002-2009
JMJ	Character Information Development and Maintenance Project for e-Government "Moji-Joho-Kiban Project" (文字情報基盤整備事業), 2010-
JK	Japanese KOKUJI Collection

NOTE 3 – When a source such as JIS X 0213:2004 includes encoded characters already included in older referenced sources, the more recent source references are typically used to reflect contemporary usage. This also reflects the fact that some of these more recent references were accompanied by minor adjustment for the graphic representation of the related characters.

NOTE 4 – J Sources that have been partially superseded (J1 and JA) are described in Annex A (see A.4.3 and A.4.4). They can also be found as provisional sources in the Unihan Data base in http://www.unicode.org/reports/tr38/ fully described as kJis1 for J1: JIS X 0212-1990 and kJA for the JA source.

The Hanja K sources are

K0	KS C 5601-1987 (now known as KS X 1001:2004)
K1	KS C 5657-1991 (now known as KS X 1002:2001)
K2	PKS C 5700-1 1994 (Reedited and standardized as KS X 1027-1:2011)
K3	PKS C 5700-2 1994 (Reedited and standardized as KS X 1027-2:2011)
K4	PKS 5700-3:1998 (Reedited and standardized as KS X 1027-3:2011)
K5	Korean IRG Hanja Character Set 5th Edition: 2001 (Reedited and standardized as KS X 1027-4:2011)
K6	KS X1027-5:2014
KC	Korean History On-Line (한국 역사 정보 통합 시스템)

The Hanja KP sources are

KP0	KPS 9566-97
KP1	KPS 10721:2000 and KPS 10721:2003

The ChuNom V sources are

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO IEC 10646-2020

V0 TCVN 5773:1993
V1 TCVN 6056:1995
V2 VHN 01:1998
V3 VHN 02: 1998
V4 Dictionary on Nom 2006, Dictionary on Nom of Tay ethnic 2006, Lookup Table for Nom in the South 1994

The Unicode U source is

UTC The Unicode Technical Report #45, U-source Ideographs, February 15, 2019

The UK UK source is

UK The Unicode Technical Report #45, UK-source Ideographs, February 15, 2019

The SAT S source is

SAT Taishō Shinshū Daizōkyō (大正新脩大藏經), 1924-1934

24.2 Source references file for CJK ideographs

The source references for CJK ideographs are defined by a machine-readable format that is accessible as a link: https://standards.iso.org/iso-iec/10646/ed-6/en/CJKSrc.txt.

The content linked to is a plain text file, encoded as UTF-8 with CARRIAGE RETURN/ LINE FEED as end of line mark. The file specifies the sources references data for all CJK ideographs. The file also contains information concerning Radical-Stroke index, the corresponding CJK Unified ideograph code point value for each CJK Compatibility ideograph entry, and information about the IICORE collection (see A.4.1) for characters belonging to that collection. Each line in the text file contains the following information organized in three fields:

— 1st field: UCS code point in the format `(U+hhhh)` or `(U+hhhhh)`
— 2nd field: Tag indicating the type of information in the third field `(kIRG_GSource, kIRG_HSource, kIRG_MSource, kIRG_TSource, kIRG_JSource, kIRG_KSource, kIRG_KPSource, kIRG_VSource, kIRG_USource, kIRG_UKSource, kIRG_SSsource, kIICore, kCompatibilityVariant, kRSUnicode, kTotalStrokes)`
— 3rd field: Information corresponding to the tag value specified by the second field. Table 5 provides the format details.

The format definition for the first field uses `'h'` as a hexadecimal unit. The three fields are delimited by a CHARACTER TABULATION control character (0009). Comment lines, starting with a NUMBER SIGN ('#') are informational only. Comment lines and blank lines in the text file should be ignored by any automatic process which parses the data file to extract the source reference information.

Table 5: Format details of the tags used in the source reference file for CJK ideographs

Tag Value	Tag description	Third field format
`kIRG_GSource`	Hanzi G source	`(G0-hhhh), (G1-hhhh), (G3-hhhh), (G5-hhhh), (G7-hhhh), (GS-hhhh), (G8-hhhh), (G9-hhhh), (G9-hhhhhhhh), (GE-hhhh), (GH-hhhh), (GK-hhhh), (G4K), (G4K-ddddd), (GBK), (GBK-dddd.dd), (GCE-ddd), (GCH), (GCH-dddd.dd), (GCY), (GCY-dddd.dd), (GCYY-ddddd), (GDZ-dddd.dd), (GFC-ddd), (GFZ-hhhh), (GGFZ-dddddd), (GGH-ddddd.dd), (GHC), (GHC-dddd.dd), (GHF-dddd), (GHZ-ddddd.dd), (GHZR-ddddd.dd), (GIDC-ddd), (GJZ-ddddd), (GKJ-ddddd), (GKX-dddd.dd),(GLGYJ-dddd), (GLK-ddddddd), (GOCD-ddd), (GPGLG-dddd), (GRM-dddd.dd), (GT-dddd), (GU-hhhhh), (GXC-dddd.dd), (GXH-dddd.dd), (GXHZ-ddd), (GWZ-dddd.dd),`

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 10646-2020

Tag Value	Tag description	Third field format
		(GZ-ddddddd), (GZFY-ddddd), (GZH-dddd.dd), (GZYS-ddddd), or (GZJW-ddddd)
kIRG_HSource	Hanzi H source	(H-hhhh), (HB0-hhhh), (HB1-hhhh), (HB2-hhhh), (HD-hhhh), or (HU-hhhhh)
kIRG_MSource	Hanzi M source	(MAC-ddddd)
kIRG_TSource	Hanzi T source	(T1-hhhh), (T2-hhhh), (T3-hhhh), (T4-hhhh), (T5-hhhh), (T6-hhhh), (T7-hhhh), (TB-hhhh), (TC-hhhh), (TD-hhhh), (TE-hhhh), (TF-hhhh), (T13-hhhh), or (TU-hhhhh)
kIRG_JSource	Kanji J source	(J0-hhhh), (J1-hhhh), (J3-hhhh), (J3A-hhhh), (J13-hhhh), (J13A-hhhh), (J4-hhhh), (J14-hhhh), (JA-hhhh), (JA3-hhhh), (JA4-hhhh), (JARIB-hhhh), (JH-xxxxxx), (JH-xxxxxxS), (JMJ-dddddd), or (JK-ddddd)
kIRG_KSource	Hanja K source	(K0-hhhh), (K1-hhhh), (K2-hhhh), (K3-hhhh), (K4-hhhh), (K5-hhhh), (KC-ddddd), or (KU-hhhhh)
kIRG_KPSource	Hanja KP source	(KP0-hhhh),(KP1-hhhh), or (KPU-hhhhh)
kIRG_VSource	ChuNom V source	(V0-hhhh), (V1-hhhh), (V2-hhhh), (V3-hhhh), (V4-hhhh), or (VU-hhhhh)
kIRG_USource	Unicode U source	(UTC-ddddd)
kIRG_UKSource	UK UK source	(UK-ddddd)
kIRG_SSource	SAT S source	(SAT-ddddd)
kIICore	IICORE info	([ABC]{1}[GTJHKMP]{1,7})
kCompatibility Variant	Compatibility info	(U+hhhh), or (U+hhhhh)
kRSUnicode	Radical Stroke index	((d{1,3}'.d{1,2})) (<space> (d{1,3}'.d{1,2}))*
kTotalStrokes	Total Strokes	((d{1,3}) (<space> (d{1,3}))*

The format definition uses 'd' as a decimal unit, 'h' as a hexadecimal unit, 'x' as an alphanumerical unit (0 to 9 and A to Z), and <space> as the SPACE character. Uppercase characters, digits and all other symbols between parentheses appear as shown. An ASTERISK indicates zero, one, or more iteration of the preceding pattern.

The IICORE value ([ABC]{1}[GTJHKMP]) is a field starting with one of the letters A, B, or C indicating a decreasing order of priority, followed by one or several letters (G,T,J,H,K,M,P) indicating usage from the Hanzi G source, Hanzi T source, Kanji J source, Hanzi H source, Hanja K source, Hanzi M source, and Hanja KP source respectively. The presence of an IICORE tag for a given CJK ideograph indicates that the character is part of the IICORE collection.

The Compatibility information is a field specified for all CJK Compatibility ideographs that provides the code point of the corresponding CJK Unified ideograph.

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub

Full document details: ISO IEC 10646-2020

Tag Value	Tag description	Third field format
The Radical-Stroke index `((d{1,3}'.d{1,2})) (space (d{1,3}'.d{1,2}))*` is a field contains one or more of the following space separated construct: a radical index (one to three digits), optionally followed by an apostrophe for simplified radicals, followed by a full stop, and ending by one or two digits for the stroke count. NOTE 1 – Concerning JIS X 0213: 2004 sources, level-4 references correspond to the second plane; other level references correspond to the first plane. NOTE 2 – The original source references in the Hanja K4 and K5 sources are described using a single decimal index without section or position values. For better consistency with the other sources, those indexes have been converted into hexadecimal values in the source reference file. Unlike K0-K3 indexes, K4 and K5 indexes do not decompose in section, position values. NOTE 3 – The reference types 'GU', 'HU', 'KU', 'KPU', 'TU', and 'VU' identifies self-referenced code points with no other sources within their source types. The value associated is simply the character code point. NOTE 4 – The UCS code points are using the U+ prefix for UCS short identifiers in the format included in the source reference file to be identical to the similar file included in the Unicode Standard.		

The following examples shows file entries for the CJK ideographs 3687, 4E00, 4E07, and F928. The three first definitions correspond to CJK Unified ideographs, with 4E00 and 4E07 part of the IICORE collection, while the fourth definition corresponds to a CJK Compatibility ideograph.

EXAMPLES

```
U+3687        kIRG_GSource           G3-3A36
U+3687        kIRG_KPSource          KP1-3C87
U+3687        kIRG_KSource           K3-2339
U+3687        kIRG_TSource           T4-2861
U+3687        kRSUnicode             35.6 66.6
U+3687        kTotalStrokes          9

U+4E00        kIRG_GSource           G0-523B
U+4E00        kIRG_HSource           HB1-A440
U+4E00        kIRG_JSource           J0-306C
U+4E00        kIRG_KPSource          KP0-FCD6
U+4E00        kIRG_KSource           K0-6C69
U+4E00        kIRG_TSource           T1-4421
U+4E00        kIRG_VSource           V1-4A21
U+4E00        kRSUnicode             1.0
U+4E00        kIICore                AGTJHKMP
U+4E00        kTotalStrokes          1

U+4E07        kIRG_GSource           G0-4D72
U+4E07        kIRG_HSource           HB2-C945
U+4E07        kIRG_JSource           J0-4B7C
U+4E07        kIRG_KPSource          KP0-DAB9
U+4E07        kIRG_KSource           K0-5832
U+4E07        kIRG_TSource           T2-2126
U+4E07        kIRG_VSource           V1-4A24
U+4E07        kRSUnicode             1.2
U+4E07        kIICore                AGJKP
U+4E07        kTotalStrokes          3

U+F928        kIRG_JSource           J3-742E
U+F928        kIRG_KSource           K0-5227
U+F928        kRSUnicode             53.9
U+F928        kCompatibilityVariant  U+5ECA
U+F928        kTotalStrokes          13
```

24.3 Source reference presentation for CJK Unified ideographs

24.3.1 General

Unlike many other character repertoires which only show one graphic symbol per character, the code charts for CJK Unified ideographs show one graphic symbol per source reference with additional information related to their identity such as radical, stroke count, and numeric value of the various source references.

NOTE – The presentation of the twelve CJK Unified ideographs included in the CJK COMPATIBILITY block follows a different model described in 24.4.

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 10646-2020

The graphic representation for the radical is shown immediately below the code point, along with the radical number and the stroke count. That stroke count does not include the radical itself.

The code chart for the CJK UNIFIED IDEOGRAPHS block (4E00-9FFF) uses a fixed column format (i.e. source references from a given source always appear in the same column) while the code charts for the other CJK Unified blocks show graphic symbols per the following order of appearance: G, T, J, K, KP, V, H, M, and U.

24.3.2 Source reference presentation for CJK UNIFIED IDEOGRAPHS block

For the presentation of the CJK UNIFIED IDEOGRAPH block, the graphic representations for the Hanzi G, H, and T sources, the Kanji J source, the Hanja K source, and the ChuNom V source are shown in that order when present. Unicode U sources are shown in the second column (instead of the Hanzi H source). No characters in this block have both a H and U source.

Figure 2 shows an example for characters 4E00-4E09 and 4E14-4E1D.

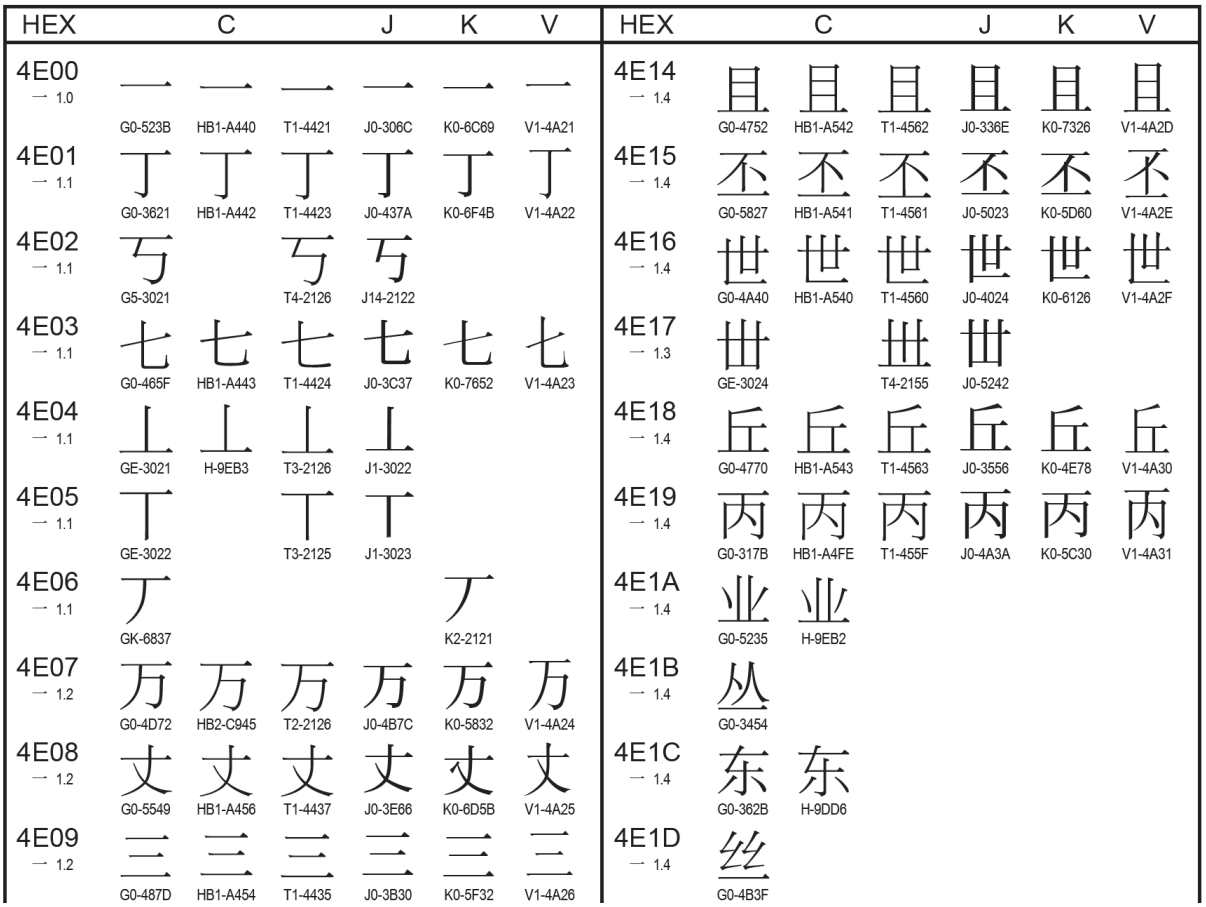


Figure 2 – Chart presentation for CJK UNIFIED IDEOGRAPHS

24.3.3 Source reference presentation for CJK UNIFIED IDEOGRAPHS EXTENSION A

For the presentation of the CJK UNIFIED IDEOGRAPH EXTENSION A block, up to three sources per characters are represented in a single row. If more than three sources exist, an additional row is used.

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 10646-2020

Figure 3 shows an example for characters 41C9-41CC, 41DB-41DD, and 41EE-41F0.

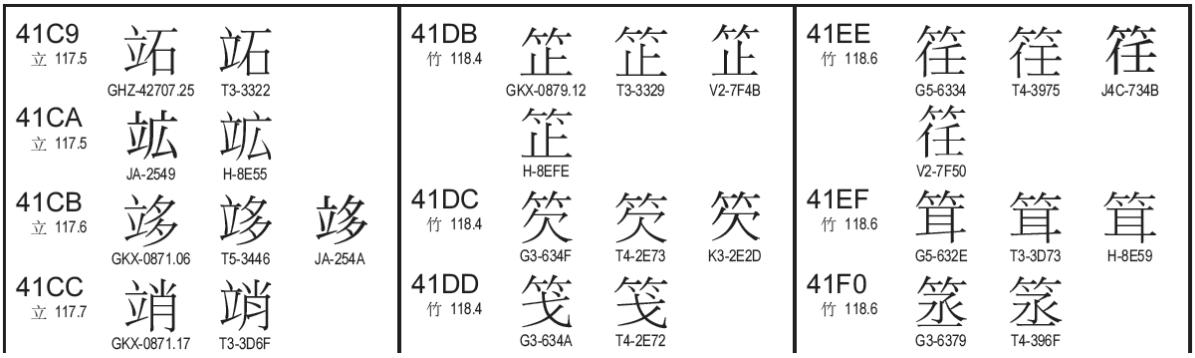


Figure 3 – Chart presentation for CJK UNIFIED IDEOGRAPHS EXTENSION A

24.3.4 Source reference presentation for CJK UNIFIED IDEOGRAPHS EXTENSION B

For the presentation of the CJK UNIFIED IDEOGRAPH EXTENSION B block, the first graphic symbol shows the glyph used for the first and second edition of this document (2003 and 2011 respectively) referenced by a 'UCS2003' notation. Up to two sources per characters are represented in a single row. If more than two sources exist, an additional row is used.

Figure 4 shows an example for characters 200E8-200EB, 200FC-200FF, and 2010E-20110.

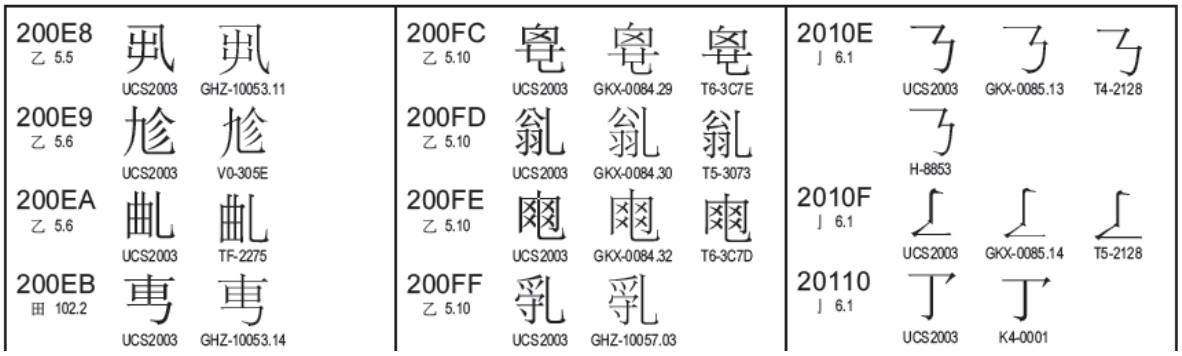


Figure 4 – Chart presentation for CJK UNIFIED IDEOGRAPHS EXTENSION B

24.3.5 Source reference presentation for CJK UNIFIED IDEOGRAPHS EXTENSION C, D, E, F, and G

For the presentation of the CJK UNIFIED IDEOGRAPH EXTENSION C, D, E, F, and G blocks, up to two sources per characters are represented in a single row. If more than two sources exist, an additional row is used.

Figure 5 shows an example for characters 2A7A0-2A7A2, 2A7B4-2A7B6, 2A7C8-2A7CA, and 2A7DC-2A7DE.

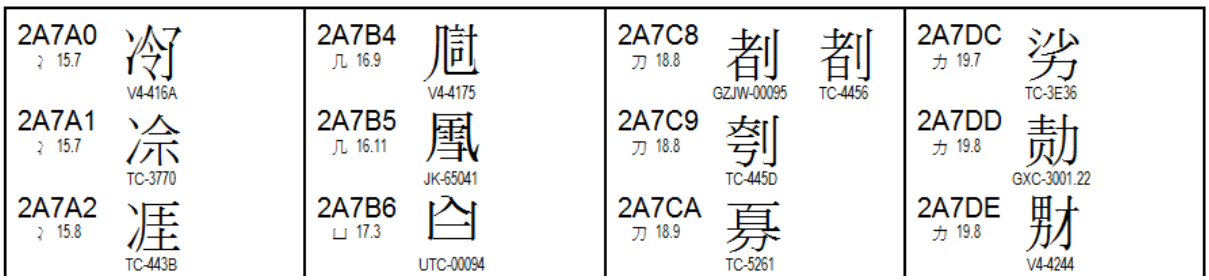


Figure 5 – Chart presentation for CJK UNIFIED IDEOGRAPHS EXTENSION C, D, E, F and G

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 10646-2020

24.4 Source references presentation for CJK Compatibility ideographs

Similarly to CJK Unified ideographs, the code charts for CJK Compatibility ideographs show one graphic symbol per source reference with additional information related to their identity such as radical, stroke count, and numeric value of the various source references.

The graphic representation for the radical is also shown immediately below the code point, along with the radical number and the stroke count. That stroke count does not include the radical itself.

Additional information, corresponding to informative items described in 34.3, such as decomposition mapping, and standardized variation sequences may also appear in the presentation.

NOTE 1 – The presentation of the twelve CJK Unified ideographs included in the CJK COMPATIBILITY IDEOGRAPHS block follows this model. It does not include decomposition mapping.

NOTE 2 – Unlike the convention used for other blocks, the standardized variation sequences for CJK Unified ideographs are not shown in the blocks where the base character characters are described but instead in the block of the CJK Compatibility ideographs to which they are associated.

Figure 6 shows an example for characters F9F8-F9FC, FA04-FA08, and FA0F-FA13.

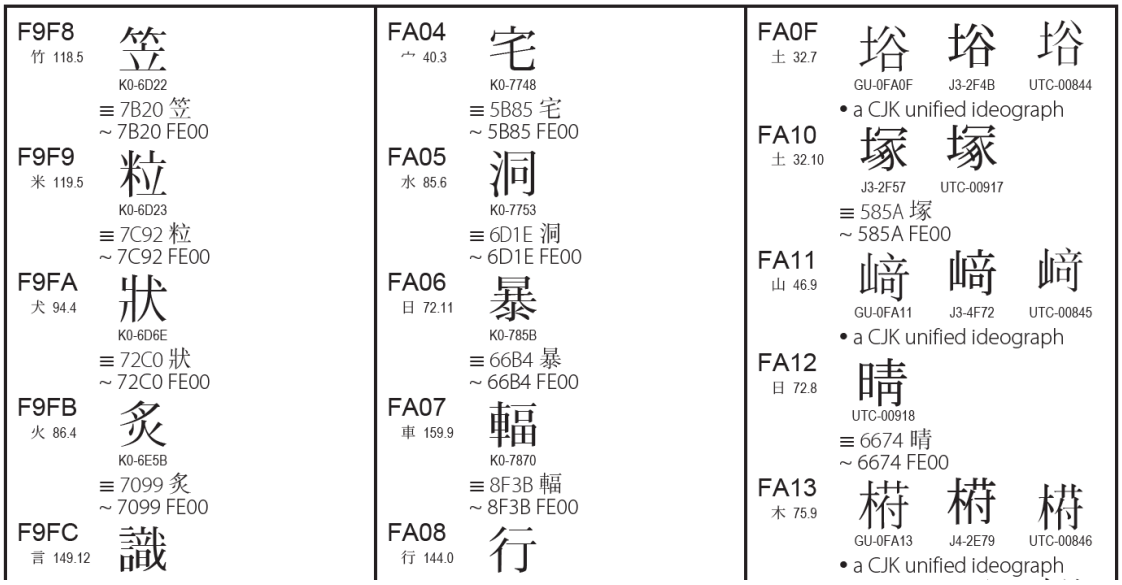


Figure 6 – Chart presentation for CJK COMPATIBILITY IDEOGRAPHS

25 Source references for Tangut ideographs

25.1 List of source references

A Tangut ideograph is always referenced by one source reference. These source references are provided in a machine-readable format that is accessible as a link from this document. The content pointed by this link is also normative.

The source reference information establishes the character identity for Tangut ideographs. A source reference is established by associating a Tangut ideograph code point with one value in the source standards listed below. The Tangut sources are:

H2004-A	Hán Xiǎománg 2004 (PhD dissertation on the correct form of Tangut ideographs), H1 subdivision
H2004-B	Hán Xiǎománg 2004 (PhD dissertation on the correct form of Tangut ideographs), H2 subdivision
L1986	Lǐ Fànwén 1986 (Study of the Homophones)

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 10646-2020

L1997 Lǐ Fànwén 1997 (Tangut-Chinese Dictionary, 1st edition)
L2006 Lǐ Fànwén 2006 (Comparative Study of Wuyin Qieyun and Wenhai Baoyun)
L2008 Lǐ Fànwén 2008 (Tangut-Chinese Dictionary, 2nd edition)
L2012 Lǐ Fànwén 2012 (Tangut-Chinese Dictionary, 3rd edition)
N1966 Nishida 1966 (Little Dictionary of Tangut)
S1968 Sofronov 1968 (Grammar of the Tangut Language)
UTN42 Unicode Technical Note #42: Tangut Character Additions and Glyphs Corrections

25.2 Source reference file for Tangut ideographs

The source references for Tangut ideographs are defined by a machine-readable format that is accessible as a link: https://standards.iso.org/iso-iec/10646/ed-6/en/TangutSrc.txt.

The content linked to is a plain text file, encoded as UTF-8 with CARRIAGE RETURN/ LINE FEED as end of line mark. The file specifies the sources references data for all Tangut ideographs. The file also contains information concerning Radical-Stroke index. Each line in the text file contains the following information organized in three fields:

— 1st field: UCS code point in the format `(U+hhhhh)`

— 2nd field: Tag indicating the type of information in the third field `(kTGT_MergedSrc, kRSTU-nicode)`

— 3rd field: Information corresponding to the tag value specified by the second field. Table 6 provides the format details.

The format definition for the first field uses 'h' as a hexadecimal unit. The three fields are delimited by a CHARACTER TABULATION control character (0009). Comment lines, starting with a NUMBER SIGN ('#') are informational only. Comment lines and blank lines in the text file should be ignored by any automatic process which parses the data file to extract the source reference information.

Table 6: Format details of the tags used in the source reference file for Tangut ideographs

Tag Value	Tag description	Third field format
`kTGT_MergedSrc`	Tangut source	`(H2004-A-dddd)`, `(H2004-B-dddd)`, `(L1986-dddd)`, `(L1997-dddd)`, `(l2006-dddd)`, `(L2008-dddd)`, `(L2008-dddd-dddd)`, `(L2012-dddd)`, `(N1966-ddd-ddx)`, `(N1966-ddd-ddxx)`, `(S1968-dddd)`, or `(UT42-ddd)`
`kRSTUnicode`	Radical Stroke index	`(d{1,3}.d{1,2})`
The format definition uses `'d'` as a decimal unit, and `'x'` as an alphanumerical unit (0 to 9 and A to Z). Uppercase characters, digits and all other symbols between parentheses appear as shown. The Radical-Stroke index `(d{1,3}.d{1,2})` is a field contains one construct: a radical index (one to three digits), followed by a full stop, and ending by one or two digits for the stroke count. The stroke count includes all strokes for the character, including the radical. The radical index is an entry on the Tangut Components block at 18800-18AFF, with the index '1' representing the code point 18800. NOTE – Concerning Lǐ Fànwén 2008 entries, the notation `(L2008-dddd)` denotes a character mapped to a single entry while the notation `(L2008-dddd-dddd)` denotes a character mapped to two entries.		

The following examples shows file entries for the Tangut ideographs 17000, 17001, 17A7A, and 17E91.

EXAMPLES

```
U+17000     kTGT_MergedSrc      L2008-0008
U+17000     kRSTUnicode         1.6
U+17001     kTGT_MergedSrc      L2008-0030
U+17001     kRSTUnicode         1.7
```

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 10646-2020

```
U+17A7A      kTGT_MergedSrc        N1966-204-10GG
U+17A7A      kRSTUnicode           141.14

U+17E91      kTGT_MergedSrc        L2008-2027-2350
U+17E91      kRSTUnicode           259.10
```

25.3 Source reference presentation for Tanguts ideographs

The code charts for Tangut ideographs show one graphic symbol per character, along with additional information related to their identity such as radical, stroke count, and numeric value of the various source references.

The graphic representation for the radical is shown immediately below the code point, along with the radical number and the stroke count. That stroke count includes the radical itself.

NOTE – The radicals shown in this representation are part of the Tangut Components block at 18800-18AFF.

Figure 7 shows an example for characters 177D0-177D2, 177E4-177E6, 177F8-177FA, 1780C-1780E, and 17820-17822.

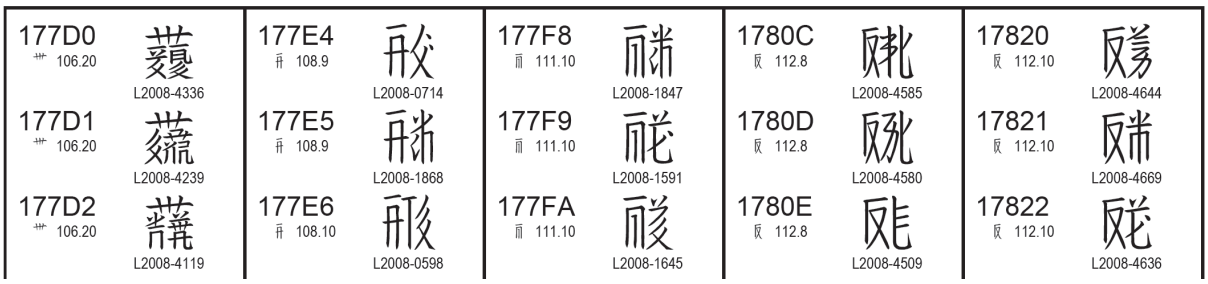


Figure 7 – Chart presentation for TANGUT and TANGUT SUPPLEMENT

26 Source references for Nüshu characters

26.1 List of source references

A Nüshu character is always referenced by one or more source references. These source references are provided in a machine-readable format that is accessible as a link from this document. The content pointed by this link is also normative.

The source reference information establishes the character identity for Nüshu characters. A source reference is established by associating a Nüshu character code point with one value in the source standards listed below. The Nüshu sources are:

NushuDuben	Nüshu duben 女书读本 (2008)
NushuYongziBijiao	Nüshu yongzi bijiao 女书用字比较 (2006)

NOTE – Only NushuDuben is used in this version for references. NushuYongziBijjiao and other sources may be added in the future.

26.2 Source reference file for Nüshu characters

The source references for Nüshu characters are defined by a machine-readable format that is accessible as a link: https://standards.iso.org/iso-iec/10646/ed-6/en/NushuSrc.txt.

The content linked to is a plain text file, encoded as UTF-8 with CARRIAGE RETURN/ LINE FEED as end of line mark. The file specifies the sources references data for all Nüshu characters. Each line in the text file contains the following information organized in three fields:

— 1st field: UCS code point in the format `(U+hhhhh)`

— 2nd field: Tag indicating the type of information in the third field `(kSrc_NushuDuben, kReading)`

— 3rd field: Information corresponding to the tag value specified by the second field. Table 7 provides the format details.

© ISO/IEC 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: ISO IEC 10646-2020