This PDF file is an excerpt from *The Unicode Standard, Version 4.0*, issued by the Unicode Consortium and published by Addison-Wesley. The material has been modified slightly for this online edition, however the PDF files have not been modified to reflect the corrections found on the Updates and Errata page (http://www.unicode.org/errata/). For information on more recent versions of the standard, see http://www.unicode.org/standard/versions/enumeratedversions.html.

Dai Kan-Wa Jiten used as the source of reference Kanji codes was written by Tetsuji Morohashi and published by Taishukan Shoten.

Cover and CD-ROM label design: Steve Mehallo, http://www.mehallo.com

The publisher offers discounts on this book when ordered in quantity for bulk purchases and special sales. For more information, customers in the U.S. please contact U.S. Corporate and Government Sales, (800) 382-3419, corpsales@pearsontechgroup.com. For sales outside of the U.S., please contact International Sales, +1 317 581 3793, international@pearsontechgroup.com

Visit Addison-Wesley on the Web: http://www.awprofessional.com

Library of Congress Cataloging-in-Publication Data
The Unicode Standard, Version 4.0 : the Unicode Consortium /Joan Aliprand... [et al.].
p. cm.
Includes bibliographical references and index.
ISBN 0-321-18578-1 (alk. paper)
1. Unicode (Computer character set). I. Aliprand, Joan.

QA268.U545 2004
005.7'2—dc21

2003052158

ISBN 0-321-18578-1
Text printed on recycled paper
1 2 3 4 5 6 7 8 9 10—CRW—0706050403
First printing, August 2003

Preface

This book, *The Unicode Standard, Version 4.0*, is the authoritative source of information on the Unicode character encoding standard.

Version 4.0 expands on and supersedes all other previous versions. The text of the standard has been extensively rewritten to improve its structure and clarity.

Major additions to Version 4.0 since Version 3.0 include:

- Extensive additions of CJK characters to cover dictionaries and historic usage
- Many new symbols for mathematical and technical publication
- Substantially improved specification of conformance requirements, incorporating the character encoding model
- Encoding of supplementary characters
- Formalized policies for stability of the standard
- Clarification of semantics of special characters, including the byte order mark
- Major expansion of Unicode Character Database properties and of specifications for text boundaries and casing
- More minority scripts, including Limbu, Tai Le, Osmanya, and Philippine scripts
- More historic scripts, including Linear B, Cypriot, and Ugaritic
- Tightened definition of encoding terms, including UTF-32

Furthermore, many individual characters were added to meet the requirements of users and implementers alike.

The Unicode Standard maintains consistency with the international standard ISO/IEC 10646. Version 4.0 of the Unicode Standard corresponds to ISO/IEC 10646:2003.

0.1 About the Unicode Standard

This book, together with the Unicode Standard Annexes (described in Appendix B) and the Unicode Character Database, defines Version 4.0 of the Unicode Standard. The book gives the general principles, requirements for conformance, and guidelines for implementers, followed by character code charts and names.

Concepts, Architecture, Conformance, and Guidelines

The first five chapters of Version 4.0 introduce the Unicode Standard and provide the fundamental information needed to produce a conforming implementation. Basic text processing, working with combining marks, and encoding forms are all described. A special

chapter on implementation guidelines answers many common questions that arise when implementing Unicode.

Chapter 1 introduces the standard's basic concepts, design basis, and coverage, and discusses basic text handling requirements.

Chapter 2 sets forth the fundamental principles underlying the Unicode Standard and covers specific topics such as text processes, overall character properties, and the use of combining marks.

Chapter 3 constitutes the formal statement of conformance. This chapter also presents the normative algorithms for two processes: the canonical ordering of combining marks and the encoding of Korean Hangul syllables by conjoining *jamo*.

Chapter 4 describes character properties in detail, both normative (required) and informative. Tables giving additional character property information appear in the Unicode Character Database.

Chapter 5 discusses implementation issues, including compression, strategies for dealing with unknown and unsupported characters, and transcoding to other standards.

Character Block Descriptions

Chapters 6 through 15 contain the character block descriptions that give basic information about each script or group of symbols and may discuss specific characters or pertinent layout information. Some of this information is required to produce conformant implementations of these scripts and other collections of characters.

Chapter 6 introduces writing systems and describes the general punctuation characters.

Chapter 7 presents the European Alphabetic scripts, including Latin, Greek, Cyrillic, Armenian, Georgian, and associated combining marks.

Chapter 8 presents the Middle Eastern, right-to-left scripts: Hebrew, Arabic, Syriac, and Thaana.

Chapter 9 covers the South Asian scripts, including Devanagari, Bengali, Gurmukhi, Gujarati, Oriya, Tamil, Telugu, Kannada, Malayalam, Sinhala, Tibetan, and Limbu.

Chapter 10 covers the Southeast Asian scripts, including Thai, Lao, Tai Le, Myanmar, Khmer, and Philippine scripts.

Chapter 11 presents the East Asian scripts, including Han, Hiragana, Katakana, Hangul, Bopomofo, and Yi.

Chapter 12 presents other scripts, including Ethiopic, Mongolian, Osmanya, Cherokee, Canadian Aboriginal Syllabics, Deseret, and Shavian.

Chapter 13 describes archaic scripts, including Ogham, Old Italic, Runic, Gothic, Ugaritic, Linear B, and Cypriot.

Chapter 14 presents symbols, including currency, letterlike and technical symbols, mathematical operators, and musical symbols.

Chapter 15 describes other topics such as private-use characters, surrogate code points, and special characters.

Charts and Han Radical-Stroke Index

The next two chapters document the Unicode Standard's character code assignments, their names, and important descriptive information, and provide a Han radical-stroke index that aids in locating specific ideographs encoded in Unicode.

Chapter 16 gives the code charts and the Character Names List. The code charts contain the normative character encoding assignments, and the names list contains normative information as well as useful cross references and informational notes.

Chapter 17 provides a radical-stroke index to East Asian ideographs.

Appendices

The appendices contain detailed background information on important topics regarding the history of the Unicode Standard and its relationship to ISO/IEC 10646.

Appendix A describes the history of Han Unification in the Unicode Standard.

Appendix B provides abstracts of Unicode Technical Reports and lists other important Unicode resources.

Appendix C details the relationship between the Unicode Standard and ISO/IEC 10646.

Appendix D lists the changes to the Unicode Standard since Version 3.0.

The appendices are followed by a glossary of terms, a bibliography, and two indices: an index to Unicode characters and an index to the text of the book.

0.2 The Unicode Character Database and Technical Reports

The Unicode Character Database is a collection of data files that contain character code points, character names, and character property data. It is described more fully in *Section 4.1, Unicode Character Database*. All versions, including the most up-to-date version of the Unicode Character Database, are found on the Unicode Web site:

http://www.unicode.org/ucd/

The files for Version 4.0.0 of the Unicode Character Database are also supplied on the CD-ROM that accompanies this book.

Information on versions of the Unicode Standard can be found on the Unicode Web site:

http://www.unicode.org/versions/

All versions of all Unicode Technical Reports, Unicode Technical Standards, and Unicode Standard Annexes are available on the Unicode Web site:

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

The latest available version of each document at the time of publication is included on the CD-ROM. See Appendix B for a summary overview of important Unicode Technical Standards, Unicode Technical Reports, and Unicode Standard Annexes.

On the CD-ROM

The CD-ROM contains additional information, such as sample code, which is maintained on the Unicode FTP site:

ftp.unicode.org

It is also available via HTTP:

http://www.unicode.org/Public/

For the complete contents of the CD-ROM, see its ReadMe.txt file.

0.3 Notational Conventions

Throughout this book, certain typographic conventions are used.

Code Points

In running text, an individual Unicode code point can be expressed as U+*n*, where *n* is four to six hexadecimal digits, using the digits 0–9 and uppercase letters A–F (for 10 through 15, respectively). There should be no leading zeros, unless the code point would have fewer than four hexadecimal digits—for example, U+0001, U+0012, U+0123, U+1234, U+12345, U+102345.

- U+0416 is the Unicode code point for the character named CYRILLIC CAPITAL LETTER ZHE.

In tables, the *U+* may be omitted for brevity.

A range of Unicode code points is expressed as *U+xxxx–U+yyyy* or *xxxx..yyyy*, where *xxxx* and *yyyy* are the first and last Unicode values in the range, and the long dash or two dots indicate a contiguous range inclusive of the endpoints. For ranges involving supplementary characters, the code points in the ranges are expressed with five or six hexadecimal digits.

- The range U+0900–U+097F contains 128 Unicode code points.
- The Plane 16 private-use characters are in the range 100000..10FFFD.

Character Names

All Unicode characters have unique names, which are identical to those of the English-language edition of International Standard ISO/IEC 10646. Unicode character names contain only uppercase Latin letters A through Z, digits, space, and hyphen-minus; this convention makes it easy to generate computer-language identifiers automatically from the names. Unified CJK ideographs are named CJK UNIFIED IDEOGRAPH-X, where X is replaced with the hexadecimal Unicode code point—for example, CJK UNIFIED IDEOGRAPH-4E00. The names of Hangul syllables are generated algorithmically; for details, see "Hangul Syllable Names" in *Section 3.12, Conjoining Jamo Behavior*.

In running text, a formal Unicode name is shown in small capitals (for example, GREEK SMALL LETTER MU), and alternative names (aliases) appear in italics (for example, *umlaut*). Italics are also used to refer to a text element that is not explicitly encoded (for example, *pasekh alef*) or to set off a non-English word (for example, the Welsh word *ynghyd*).

Sequences

A sequence of two or more code points may be represented by a comma-delimited list, set off by angle brackets. For this purpose angle brackets consist of U+003C LESS-THAN SIGN and U+003E GREATER-THAN SIGN. Spaces are optional after the comma, and U+ notation for the code point is also optional—for example, "<U+0061, U+0300>".

When the usage is clear from the context, a sequence of characters may also be represented with generic short names, as in "<a, grave>", or the angle brackets may be omitted.

In contrast to sequences of code points, a sequence of one or more code *units* may be represented by a list set off by angle brackets, but without comma delimitation or U+ notation. For example, the notation "<nn nn nn nn>" represents a sequence of bytes, as for the UTF-8 encoding form of a Unicode character. The notation "<nnnn nnnn>" represents a sequence of 16-bit code units, as for the UTF-16 encoding form of a Unicode character.

Miscellaneous

Phonemic transcriptions are shown between slashes, as in Khmer /khnyom/.

Phonetic transcriptions are shown between square brackets, using the International Phonetic Alphabet. (Full details on the IPA can be found on the International Phonetic Association's Web site, http://www2.arts.gla.ac.uk/IPA/ipa.html.)

A leading asterisk is used to represent an incorrect or nonoccurring linguistic form.

The symbols used in the character names list are described at the beginning of *Chapter 16, Code Charts*.

In the text of this book, the word "Unicode" when used alone as a noun refers to the Unicode Standard.

Unambiguous dates of the current common era, such as 1999, are unlabeled. In cases of ambiguity, CE is used. Dates before the common era are labeled with BCE.

The term *byte*, as used in this standard, always refers to a unit of eight bits. This corresponds to the use of the term *octet* in some other standards.

Extended BNF

The Unicode Standard and technical reports use an extended BNF format for describing syntax. As different conventions are used for BNF, *Table 0-1, Extended BNF*, lists the notation used here.

Table 0-1. Extended BNF

Symbols	Meaning
`x := ...`	production rule
`x y`	the sequence consisting of `x` then `y`
`x*`	zero or more occurrences of `x`
`x?`	zero or one occurrence of `x`
`x+`	one or more occurrences of `x`
`x | y`	either `x` or `y`
`( x )`	for grouping
`x || y`	equivalent to `(x | y | (x y))`
`{ x }`	equivalent to `(x)?`
`"abc"`	string literals ("_" is sometimes used to denote space for clarity)
`'abc'`	string literals (alternative form)

Table 0-1. Extended BNF (Continued)

Symbols	Meaning
sot	start of text
eot	end of text
\u1234	Unicode code points within string literals or character classes
\U00101234	Unicode code points within string literals or character classes
U+HHHH	Unicode character literal: equivalent to '\uHHHH'
U-HHHHHHHH	Unicode character literal: equivalent to '\UHHHHHHHH'
charClass	character class (syntax below)

In other environments, such as programming languages or markup, alternative notation for sequences of code points or code units may be used.

Character Classes. A *code point class* is a specification of an unordered set of code points. Whenever the code points are all assigned characters, it can also be referred to as a *character class*. The specification consists of any of the following:

- A literal code point
- A range of literal code points
- A set of code points having a given Unicode character property value, as defined in the Unicode Character Database (see PropertyAliases.txt and PropertyValueAliases.txt)
- Non-Boolean properties given as an expression <property> = <property_value> or <property> ≠ <property_value>, such as "General_Category=Titlecase_Letter"
- Boolean properties given as an expression <property> = true or <property> ≠ true, such as "Uppercase=true"
- Combinations of logical operations on classes

Further extensions to this specification of character classes are used in some Unicode Standard Annexes and Unicode Technical Reports. Such extensions are described in those documents, as appropriate.

A partial formal BNF syntax for character classes as used in this standard is given by the following:

```
char_class  := "[" char_class - char_class "]"
               // set difference
            := "[" item_list "]"
            := "[" property ("=" | "≠") property_value "]"
item_list   := item ("," item)?
item        := code_point  // either literal or escaped
            := code_point - code_point   // inclusive range
```

Whenever any character could be interpreted as a syntax character, it must be escaped. Where no ambiguity would result (with normal operator precedence), extra square brackets can be discarded. If a space character is used as a literal, it is escaped. Examples are found in *Table 0-2, Character Class Examples*.

Table 0-2. Character Class Examples

Syntax	Matches
`[a-z]`	English lowercase letters
`[a-z]-[c]`	English lowercase letters except for c
`[0-9]`	European decimal digits
`[\u0030-\u0039]`	(same as above, using Unicode escapes)
`[0-9,A-F,a-f]`	hexadecimal digits
`[{gc=letter},{gc=non-spacing_ mark}]`	all letters and nonspacing marks
`[{gc=L},{gc=Mn}]`	(same as above, using abbreviated notation)
`[{gc≠unassigned}]`	all assigned Unicode characters
`[\u0600-\u06FF]-[{gc=unassigned}]`	all assigned Arabic characters
`[Alphabetic=true]`	all alphabetic characters
`[Line_Break≠Infix_Numeric]`	all code points that do not have the line break property of Infix_Numeric

For more information about character classes, see Unicode Technical Report #18, "Unicode Regular Expression Guidelines."

Operators

Operators used in this standard are listed in *Table 0-3, Operators*.

Table 0-3. Operators

→	is transformed to, or behaves like
↛	is not transformed to
/	integer division (rounded down)
%	modulo operation; equivalent to the integer remainder for positive numbers
¬	logical not

0.4 Resources

The Unicode Consortium provides a number of online resources for obtaining information and data about the Unicode Standard, as well as updates and corrigenda. They are listed below.

Unicode Web Site

- http://www.unicode.org

Unicode Anonymous FTP Site

- ftp://ftp.unicode.org

Unicode E-mail Discussion List

- unicode@unicode.org

Subscription instructions for the e-mail discussion list are posted on the Unicode Web site.

How to Contact the Unicode Consortium

Contact the Unicode Consortium for membership information and to order publications (including additional copies of this book).

- Postal address:

  P.O. Box 391476

  Mountain View, CA 94039-1476

  USA

Please check the Web site for up-to-date contact information, including telephone, fax, and courier delivery address.