

Reference number of working document: **ISO/IEC JTC1/SC22/WG20 N753**

Date: 2000-05-21

Reference number of document: **ISO/IEC draft DTR 14652**

Committee identification: ISO/IEC JTC1/SC22

Secretariat: ANSI

Information technology —

Specification method for cultural conventions

Technologies de l'information —

Méthode de modélisation des conventions culturelles

1

		Page
2	Contents	
3		
4	1 SCOPE	1
5	2 NORMATIVE REFERENCES	1
6	3 TERMS, DEFINITIONS AND NOTATIONS	2
7	4 FDCC-set	6
8	4.1 FDCC-set definition	6
9	4.2 LC_IDENTIFICATION	10
10	4.3 LC_CTYPE	11
11	4.4 LC_COLLATE	27
12	4.5 LC_MONETARY	42
13	4.6 LC_NUMERIC	46
14	4.7 LC_TIME	47
15	4.8 LC_MESSAGES	53
16	4.9 LC_XLITERATE	53
17	4.10 LC_NAME	55
18	4.11 LC_ADDRESS	57
19	4.12 LC_TELEPHONE	57
20	5 CHARMAP	58
21	6 REPERTOIREMAP	62
22	7 CONFORMANCE	89
23	Annex A (informative) DIFFERENCES FROM POSIX	90
24	Annex B (informative) RATIONALE	92
25	Annex C (informative) BNF GRAMMAR	106
26	Annex D (informative) ISSUES	106
27	Annex E (informative) INDEX	111
28	BIBLIOGRAPHY	114

29 **Foreword**

30

31 ISO (the International Organization for Standardization) and IEC (the International
32 Electrotechnical Commission) form the specialized system for worldwide standardization.
33 National bodies that are members of ISO or IEC participate in the development of
34 International Standards through technical committees established by the respective
35 organization to deal with particular fields of technical activity. ISO and IEC technical
36 committees collaborate in fields of mutual interest. Other international organizations,
37 governmental and non-governmental, in liaison with ISO and IEC, also take part in the
38 work. In the field of information technology, ISO and IEC have established a joint
39 technical committee, ISO/IEC JTC 1.

40

41 The main task of a technical committee is to prepare International Standards but in
42 exceptional circumstances, the publication of a Technical Report of one of the following
43 types may be proposed:

44

45 - type 1, when the required support cannot be obtained for the publication of an
46 International Standard, despite repeated efforts;

47

48 - type 2, when the subject is still under technical development or where for any
49 other reason there is the future but not immediate possibility of an agreement on an
50 International Standard;

51

52 - type 3, when a technical committee has collected data of a different kind from
53 that which is normally published as an International Standard ("state of the art", for
54 example).

55

56 Technical Reports are drafted in accordance with the rules given in the ISO/IEC
57 Directives, Part 3.

58

59 Technical Reports of types 1 and 2 are subject to review within three years of publication,
60 to decide whether they can be transformed into International Standards. Technical Report
61 of type 3 do not necessarily have to be reviewed until the date they provide are considered
62 to be no longer valid or useful.

63

64 ISO/IEC TR 14652 is a Technical Report type 1, and it was prepared by Joint Technical
65 Committee ISO/IEC JTC 1, *Information technology, Subcommittee 22, Programming
66 languages, their environments and system software interfaces.*

67

68 The Annexes A, B, C, D and E of this Technical Report are for information only.

69 Introduction

70

71 This Technical Report defines a general mechanism to specify cultural conventions, and it
72 defines formats for a number of specific cultural conventions in the areas of character
73 classification and conversion, sorting, number formatting, monetary formatting, date
74 formatting, message display, addressing of persons, postal address formatting, and
75 telephone number handling.

76

77 There are a number of benefits coming from this Technical Report:

78

79 Rigid specification Using this Technical Report, a user can rigidly specify a
80 number of the cultural conventions that apply to the
81 information technology environment of the user.

82

83 Cultural adaptability If an application has been designed and built in a
84 culturally neutral manner, the application may use the
85 specifications as data to its APIs, and thus the same
86 application may accommodate different users in a
87 culturally acceptable way to each of the users, without
88 change of the binary application.

89

90 Productivity This Technical Report specifies those cultural
91 conventions and how to specify data for them. With that
92 data an application developer is relieved from getting the
93 different information to support all the cultural
94 environments for the expected customers of the product.
95 The application developer is thus ensured of culturally
96 correct behavior as specified by the customer, and
97 possibly more markets may be reached as customers may
98 have the possibility to provide the data themselves for
99 markets that were not targeted.

100

101 Uniform behaviour When a number of applications share one cultural
102 specification, which may be supplied from the user or
103 provided by the application or operating system, their
104 behaviour for cultural adaptation becomes uniform.

105

106 The specification format is independent of platforms and specific encoding, and targeted to
107 be usable from a wide range of programming languages.

108

109 A number of cultural conventions, such as spelling, hyphenation rules and terminology, are
110 not specifiable with this Technical Report, but it provides mechanisms to define new
111 categories and also new keywords within existing categories. An internationalized
112 application may take advantage of information provided with the FDCC-set (such as the
113 language) to provide further internationalized services to the user.

114

115 This Technical Report defines a format compatible with the one used in the International
116 string ordering standard, ISO/IEC 14651. This Technical Report is upward compatible
117 with the ISO/IEC 9945-2:1993 POSIX shell and utilities standard, particularly its clauses
118 2.4 and 2.5. The major extensions from that text are listed in annex A. This Technical
119 Report has enhanced functionality in a number of areas such as ISO/IEC 10646 support,
120 more classification of characters, transliteration, dual (multi) currency support, enhanced

121 date and time formatting, personal name writing, postal address formatting, telephone
122 number handling, and management of categories. There is enhanced support for character
123 sets including ISO/IEC 2022 handling and an enhanced method to separate the
124 specification of cultural conventions from an actual encoding via a description of the
125 character repertoire employed. A standard set of values for all the categories has been
126 defined covering the repertoire of ISO/IEC 10646-1, as referenced in the normative
127 references clause.

128
129 The Technical report was originally scheduled for adoption as an International Standard,
130 but a number of members of ISO and IEC found the specification problematical. It was
131 then decided to convert the specification into a Technical Report type I. Annex D lists a
132 number of issues that some members of ISO and IEC have with the specification.
133

Information technology — Specification method for cultural conventions

1 SCOPE

This Technical Report specifies a description format for the specification of cultural conventions, a description format for character sets, and a description format for binding character names to ISO/IEC 10646, plus a set of default values for some of these items.

The specification is upward compatible with POSIX locale specifications - a locale conformant to POSIX specifications will also be conformant to the specifications in this Technical Report, while the reverse condition will not hold. The descriptions are intended to be coded in text files to be used via Application Programming Interfaces, that are expected to be developed for a number of programming languages.

2 NORMATIVE REFERENCES

The following normative documents contain provisions which, through reference in this text, constitute provisions of this Technical Report. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this Technical Report are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid Technical Reports.

ISO 639 (all parts), *Codes for the representation of names of languages*.

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

ISO 3166 (all parts), *Codes for the representation of names of countries and their subdivisions*.

ISO 4217, *Codes for the representation of currencies and funds*.

ISO 8601, *Data elements and interchange formats - Information interchange - Representation of dates and times*.

ISO/IEC 9945-2:1993, *Information technology - Portable Operating System Interface (POSIX) - Part 2: Shell and Utilities*.

ISO/IEC 10646-1:1993, *Information technology - Universal Multiple-Octet Coded Character Set (UCS) - Part 1: Architecture and Basic Multilingual Plane (including Cor.1 and AMD 1-9)*.

ISO/IEC 14651, *Information technology - International string ordering - Method for comparing character strings and description of a default tailorable ordering*.

ISO/IEC 15897:1999, *Information technology - Procedures for registration of cultural conventions*.

3 TERMS, DEFINITIONS AND NOTATIONS

187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238

3.1 Terms and definitions

For the purposes of this Technical Report, the terms and definitions given in the following apply.

3.1.1 Bytes and characters

3.1.1.1

byte:

An individually addressable unit of data storage that is equal to or larger than an octet, used to store a character or a portion of a character.

A byte is composed of a contiguous sequence of bits, the number of which is implementation defined. The least significant bit is called the low-order bit; the most significant bit is called the high-order bit.

3.1.1.2

character:

A member of a set of elements used for the organization, control or representation of data.

3.1.1.3

coded character:

A sequence of one or more bytes representing a single character.

3.1.1.4

text file:

A file that contains characters organized into one or more lines.

3.1.2 cultural and other major concepts

3.1.2.1

cultural convention:

A data item for information technology that may vary dependent on language, territory, or other cultural habits.

3.1.2.2

FDCC

A Formal Definition of a Cultural Convention, that is a cultural convention put into a formal definition scheme.

3.1.2.3

FDCC-set:

A Set of Formal Definitions of Cultural Conventions (FDCC's). The definition of the subset of a user's information technology environment that depends on language and cultural conventions. Note: the FDCC-set is a superset of the "locale" term in C and POSIX.

3.1.2.4

charmap:

A definition of a mapping between symbolic character names and character codes, plus related information.

239

240

3.1.2.5

241

repertoiremap:

242

A definition of a mapping between symbolic character names and characters for the repertoire of characters used in a FDCC-set, further described in clause 6.

243

244

245

3.1.3 FDCC categories related

246

247

3.1.3.1

248

character class:

249

A named set of characters sharing an attribute associated with the name of the class.

250

251

3.1.3.2

252

collation:

253

The logical ordering of strings according to defined precedence rules.

254

255

3.1.3.3

256

collating element:

257

The smallest entity used to determine logical ordering.

258

259

See collating sequence. A collating element consists of either a single character, or two or more characters collating as a single entity. The LC_COLLATE category in the associated FDCC-set determines the set of collating elements.

260

261

262

263

3.1.3.4

264

multicharacter collating element:

265

A sequence of two or more characters that collate as an entity.

266

267

For example, in some languages two characters are sorted as one letter, as in the case for Danish and Norwegian "aa".

268

269

270

3.1.3.5

271

collating sequence:

272

The relative order of collating elements as determined by the setting of the LC_COLLATE category in the applied FDCC-set.

273

274

275

3.1.3.6

276

equivalence class:

277

A set of collating elements with the same primary collation weight.

278

279

Elements in an equivalence class are typically elements that naturally group together, such as all accented letters based on the same letter.

280

281

282

The collation order of elements within an equivalence class is determined by the weights assigned on any subsequent levels after the primary weight.

283

284

285

3.2 Notations

286

287

The following notations and common conventions for specifications apply to this Technical Report:

288

289

290

3.2.1 Notation for defining syntax

291
292 In this Technical Report, the description of an individual record in a FDCC-set is done
293 using the syntax notation given in the following.
294

295 The syntax notation looks as follows:

296 "<format>",[<arg1>,<arg2>,....,<argn>]
297
298

299 The <format> is given in a format string enclosed in double quotes, followed by a number
300 of parameters, separated by commas. It is similar to the format specification defined in
301 clause 2.12 in the ISO/IEC 9945-2:1993 standard and the format specification used in C
302 language printf() function. The format of each parameter is given by an escape sequence
303 as follows:

304 %s specifies a string
305 %d specifies a decimal integer
306 %c specifies a character
307 %o specifies an octal integer
308 %x specifies a hexadecimal integer
309

310
311 A " " (an empty character position) in the syntax string represents one or more <blank>
312 characters.
313

314 All other characters in the format string except

315 %% specifies a single %
316 \n specifies an end-of-line
317
318

319 represent themselves.

320
321 The notation "... " is used to specify that repetition of the previous specification is optional,
322 and this is done in both the format string and in the parameter list.
323
324

325 **3.2.3 Portable character set**

326
327 A set of symbolic names for characters in Table 1, which is called the portable character
328 set, is used in character description text of this specification. The first eight entries in
329 Table 1 are defined in ISO/IEC 6429 and the rest is defined in ISO/IEC 9945-2 with some
330 definitions from ISO/IEC 10646-1.
331

332 **Table 1: Portable character set**

334 Symbolic name	335 Glyph	336 UCS	337 Description
338 <NUL>		<U0000>	NULL (NUL)
339 <alert>		<U0007>	BELL (BEL)
340 <backspace>		<U0008>	BACKSPACE (BS)
341 <tab>		<U0009>	CHARACTER TABULATION (HT)
342 <carriage-return>		<U000D>	CARRIAGE RETURN (CR)
343 <newline>		<U000A>	LINE FEED (LF)
344 <vertical-tab>		<U000B>	LINE TABULATION (VT)
345 <form-feed>		<U000C>	FORM FEED (FF)
346 <space>		<U0020>	SPACE
<exclamation-mark>	!	<U0021>	EXCLAMATION MARK
<quotation-mark>	"	<U0022>	QUOTATION MARK

347	<number-sign>	#	<U0023>	NUMBER SIGN
348	<dollar-sign>	\$	<U0024>	DOLLAR SIGN
349	<percent-sign>	%	<U0025>	PERCENT SIGN
350	<ampersand>	&	<U0026>	AMPERSAND
351	<apostrophe>	'	<U0027>	APOSTROPHE
352	<left-parenthesis>	(<U0028>	LEFT PARENTHESIS
353	<right-parenthesis>)	<U0029>	RIGHT PARENTHESIS
354	<asterisk>	*	<U002A>	ASTERISK
355	<plus-sign>	+	<U002B>	PLUS SIGN
356	<comma>	,	<U002C>	COMMA
357	<hyphen-minus>	-	<U002D>	HYPHEN-MINUS
358	<hyphen>	-	<U002D>	HYPHEN-MINUS
359	<full-stop>	.	<U002E>	FULL STOP
360	<period>	.	<U002E>	FULL STOP
361	<slash>	/	<U002F>	SOLIDUS
362	<solidus>	/	<U002F>	SOLIDUS
363	<zero>	0	<U0030>	DIGIT ZERO
364	<one>	1	<U0031>	DIGIT ONE
365	<two>	2	<U0032>	DIGIT TWO
366	<three>	3	<U0033>	DIGIT THREE
367	<four>	4	<U0034>	DIGIT FOUR
368	<five>	5	<U0035>	DIGIT FIVE
369	<six>	6	<U0036>	DIGIT SIX
370	<seven>	7	<U0037>	DIGIT SEVEN
371	<eight>	8	<U0038>	DIGIT EIGHT
372	<nine>	9	<U0039>	DIGIT NINE
373	<colon>	:	<U003A>	COLON
374	<semicolon>	;	<U003B>	SEMICOLON
375	<less-than-sign>	<	<U003C>	LESS-THAN SIGN
376	<equals-sign>	=	<U003D>	EQUALS SIGN
377	<greater-than-sign>	>	<U003E>	GREATER-THAN SIGN
378	<question-mark>	?	<U003F>	QUESTION MARK
379	<commercial-at>	@	<U0040>	COMMERCIAL AT
380	<A>	A	<U0041>	LATIN CAPITAL LETTER A
381		B	<U0042>	LATIN CAPITAL LETTER B
382	<C>	C	<U0043>	LATIN CAPITAL LETTER C
383	<D>	D	<U0044>	LATIN CAPITAL LETTER D
384	<E>	E	<U0045>	LATIN CAPITAL LETTER E
385	<F>	F	<U0046>	LATIN CAPITAL LETTER F
386	<G>	G	<U0047>	LATIN CAPITAL LETTER G
387	<H>	H	<U0048>	LATIN CAPITAL LETTER H
388	<I>	I	<U0049>	LATIN CAPITAL LETTER I
389	<J>	J	<U004A>	LATIN CAPITAL LETTER J
390	<K>	K	<U004B>	LATIN CAPITAL LETTER K
391	<L>	L	<U004C>	LATIN CAPITAL LETTER L
392	<M>	M	<U004D>	LATIN CAPITAL LETTER M
393	<N>	N	<U004E>	LATIN CAPITAL LETTER N
394	<O>	O	<U004F>	LATIN CAPITAL LETTER O
395	<P>	P	<U0050>	LATIN CAPITAL LETTER P
396	<Q>	Q	<U0051>	LATIN CAPITAL LETTER Q
397	<R>	R	<U0052>	LATIN CAPITAL LETTER R
398	<S>	S	<U0053>	LATIN CAPITAL LETTER S
399	<T>	T	<U0054>	LATIN CAPITAL LETTER T
400	<U>	U	<U0055>	LATIN CAPITAL LETTER U
401	<V>	V	<U0056>	LATIN CAPITAL LETTER V
402	<W>	W	<U0057>	LATIN CAPITAL LETTER W
403	<X>	X	<U0058>	LATIN CAPITAL LETTER X
404	<Y>	Y	<U0059>	LATIN CAPITAL LETTER Y
405	<Z>	Z	<U005A>	LATIN CAPITAL LETTER Z
406	<left-square-bracket>	[<U005B>	LEFT SQUARE BRACKET
407	<backslash>	\	<U005C>	REVERSE SOLIDUS
408	<reverse-solidus>	\	<U005C>	REVERSE SOLIDUS
409	<right-square-bracket>]	<U005D>	RIGHT SQUARE BRACKET
410	<circumflex-accent>	^	<U005E>	CIRCUMFLEX ACCENT
411	<circumflex>	^	<U005E>	CIRCUMFLEX ACCENT
412	<low-line>	_	<U005F>	LOW LINE
413	<underscore>	_	<U005F>	LOW LINE
414	<grave-accent>	`	<U0060>	GRAVE ACCENT
415	<a>	a	<U0061>	LATIN SMALL LETTER A
416		b	<U0062>	LATIN SMALL LETTER B

417	<c>	c	<U0063>	LATIN SMALL LETTER C
418	<d>	d	<U0064>	LATIN SMALL LETTER D
419	<e>	e	<U0065>	LATIN SMALL LETTER E
420	<f>	f	<U0066>	LATIN SMALL LETTER F
421	<g>	g	<U0067>	LATIN SMALL LETTER G
422	<h>	h	<U0068>	LATIN SMALL LETTER H
423	<i>	i	<U0069>	LATIN SMALL LETTER I
424	<j>	j	<U006A>	LATIN SMALL LETTER J
425	<k>	k	<U006B>	LATIN SMALL LETTER K
426	<l>	l	<U006C>	LATIN SMALL LETTER L
427	<m>	m	<U006D>	LATIN SMALL LETTER M
428	<n>	n	<U006E>	LATIN SMALL LETTER N
429	<o>	o	<U006F>	LATIN SMALL LETTER O
430	<p>	p	<U0070>	LATIN SMALL LETTER P
431	<q>	q	<U0071>	LATIN SMALL LETTER Q
432	<r>	r	<U0072>	LATIN SMALL LETTER R
433	<s>	s	<U0073>	LATIN SMALL LETTER S
434	<t>	t	<U0074>	LATIN SMALL LETTER T
435	<u>	u	<U0075>	LATIN SMALL LETTER U
436	<v>	v	<U0076>	LATIN SMALL LETTER V
437	<w>	w	<U0077>	LATIN SMALL LETTER W
438	<x>	x	<U0078>	LATIN SMALL LETTER X
439	<y>	y	<U0079>	LATIN SMALL LETTER Y
440	<z>	z	<U007A>	LATIN SMALL LETTER Z
441	<left-brace>	{	<U007B>	LEFT CURLY BRACKET
442	<left-curly-bracket>	{	<U007B>	LEFT CURLY BRACKET
443	<vertical-line>		<U007C>	VERTICAL LINE
444	<right-brace>	}	<U007D>	RIGHT CURLY BRACKET
445	<right-curly-bracket>	}	<U007D>	RIGHT CURLY BRACKET
446	<tilde>	~	<U007E>	TILDE

447
448 This Technical Report may use other symbolic character names than the above in
449 examples, to illustrate the use of the range of symbols allowed by the syntax specified in
450 4.1.1.

451 **4 FDCC-set**

452
453 A FDCC-set is the definition of the subset of a user's information technology environment
454 that depends on language and cultural conventions. It is made up from one or more
455 categories. Each category is identified by its name and controls specific aspects of the
456 behaviour of components of the system. This Technical Report defines the following
457 categories:
458

459		
460	LC_IDENTIFICATION	Versions and status of categories
461	LC_CTYPE	Character classification, case conversion and code 462 transformation.
463	LC_COLLATE	Collation order.
464	LC_TIME	Date and time formats.
465	LC_NUMERIC	Numeric, non-monetary formatting.
466	LC_MONETARY	Monetary formatting.
467	LC_MESSAGES	Formats of informative and diagnostic messages and 468 interactive responses.
469	LC_XLITERATE	Character transliteration.
470	LC_NAME	Format of writing personal names.
471	LC_ADDRESS	Format of postal addresses.
472	LC_TELEPHONE	Format for telephone numbers, and other telephone 473 information.
474		

475 Note: In future editions of this Technical Report further categories may be added.
476

477 Other category names beginning with the 3 characters "LC_" are reserved for future
478 standardization, except for category names beginning with the five characters "LC_X_"
479 which is not used for future addition of categories specified in this Technical Report. An
480 application may thus use category names beginning with the five characters "LC_X_" for
481 application defined categories to avoid clashes with future standardized categories.
482

483 This Technical Report also defines an FDCC-set named "i18n" with values for some of
484 the above categories in order to simplify FDCC-set descriptions for a number of cultures.
485 The contents of "i18n" categories should not necessarily be considered as the most
486 commonly accepted values, while in many cases it could be the recommended values.
487

488 **4.1 FDCC-set description**

489
490 FDCC-sets are described with the syntax presented in this subclause. For the purposes of
491 this Technical Report, the text is referred to as the FDCC-set definition text or FDCC-set
492 source text.
493

494 The **FDCC-set definition text** contains one or more FDCC-set category source definitions,
495 and does not contain more than one definition for the same FDCC-set category. If the text
496 contains source definitions for more than one category, application-defined categories, if
497 present, appears after the categories defined by this clause. A category source definition
498 contains either the definition of a category or a copy directive. In the event that some of
499 the information for a FDCC-set category, as specified in this Technical Report, is missing
500 from the FDCC-set source definition, the behaviour of that category, if it is referenced, is
501 unspecified. A FDCC-set category is the normal way of specifying a single FDCC.
502

503 There are no **naming conventions** for FDCC-sets specified in this Technical Report, but
504 clause 6.8 in ISO/IEC 15897:1999 specifies naming rules for POSIX locales, charmaps
505 and repertoire maps, that may also be applied to FDCC-sets, charmaps and repertoire maps
506 specified according to this Technical Report.
507

508 A **category source definition** consists of a category header, a category body, and a
509 category trailer. A category header consists of the character string naming of the category,
510 beginning with the characters "LC_". The category trailer consists of the string "END",
511 followed by one or more "blank"s and the string used in the corresponding category
512 header.
513

514 The **category body** consists of one or more lines of text. Each line is one of the
515 following:
516

- 517 - a line containing an identifier, optionally followed by one or more operands. Identifiers
518 are either keywords, identifying a particular FDCC, or collating elements, or section
519 symbols,
- 520 - one of transliteration statements defined in 4.3.
521

522 In addition to the keywords defined in this Technical Report, the source can contain
523 application-defined keywords. Each **keyword** within a category has a unique name (i.e.,
524 two categories can have a commonly-named keyword); no keyword starts with the
525 characters "LC_". Identifiers are separated from the operands by one or more "blank"s.
526

527 **Operands** are characters, collating elements, section symbols, or strings of characters.
528 Strings are enclosed in double-quotes. Literal double-quotes within strings are preceded by

529 the <escape character>, described below. When a keyword is followed by more than one
 530 operand, the operands are separated by semicolons; "blank"s are allowed before and/or
 531 after a semicolon.

532

533

534 4.1.1 Character representation

535

536 Individual characters, characters in strings, and collating elements are represented using
 537 symbolic names, UCS notation or characters themselves, or as octal, hexadecimal, or
 538 decimal constants as defined below. When constant notation is used, the resultant
 539 FDCC-set definitions need not be portable between systems.

540

541 (0) The left angle bracket (<) is a reserved symbol, denoting the
 542 start of a symbolic name; when used to represent itself
 543 outside a symbolic name it is preceded by the escape
 544 character.

545

546 (1) A character can be represented via a **symbolic name**,
 547 enclosed within angle brackets (< and >). The symbolic
 548 name, including the angle brackets, exactly matches a
 549 symbolic name defined in a charmap or a repertoiremap to
 550 be used, and is replaced by a character value determined
 551 from the value associated with the symbolic name in the
 552 charmap or a value associated via a repertoiremap.
 553 Repertoiremaps have predefined symbolic names for UCS
 554 characters, see clause 6. A FDCC-set may also use the UCS
 555 notation of clause 6 to represent characters, without a
 556 repertoiremap being defined for the FDCC-set. Use of the
 557 escape character or a right angle bracket within a symbolic
 558 name is invalid unless the character is preceded by the
 559 escape character.

560

561 Example: <c>;<c-cedilla> "<M><a><y>"

562

563 The items (2), (3), (4) and (5) are deprecated and are retained for compatibility with the
 564 POSIX standard. FDCC-sets should be specified in a coded character set independent way,
 565 using symbolic names. To make actual use of the FDCC-set, it is used together with
 566 charmaps and/or repertoiremaps, so that the symbolic character names can be resolved into
 567 the actual character encoding used.

568

569 (2) A character can be represented by the character itself, in
 570 which case the value of the character is application-defined.
 571 Within a string, the double-quote character, the escape
 572 character, and the right angle bracket character are escaped
 573 (preceded by the escape character) to be interpreted as the
 574 character itself. Outside strings, the characters

575

576 , ; < > escape_char

577

578 are escaped by the escape character to be interpreted as the character itself.

579

580 Example: c ä "May"

581 (3) A character can be represented as an octal constant. An octal
582 constant is specified as the escape character followed by two
583 or more octal digits. Each constant represents a byte value.

584 Example: `\143; \347; "\115"`
585

586
587 (4) A character can be represented as a hexadecimal constant. A
588 hexadecimal constant is specified as the escape character
589 followed by an x followed by two or more hexadecimal
590 digits. Each constant represents a byte value.

591 Example: `\x63;\xe7;`
592

593
594 (5) A character can be represented as a decimal constant. A
595 decimal constant is specified as the escape character
596 followed by a d followed by two or more decimal digits.
597 Each constant represents a byte value.

598 Example: `\d99; \d231;`
599

600
601 (6) Multibyte characters can be represented by concatenated
602 constants specified in byte order with the last constant
603 specifying the least significant byte of the character.
604 Concatenated constants can include a mix of the above
605 character representations.

606 Example: `\143\xe7; "\115\xe7\d171"`
607

608
609 Only characters existing in the character set for which the FDCC-set definition is created
610 are specified, whether using symbolic names, the characters themselves, or octal, decimal,
611 or hexadecimal constants. If a charmap is present, only characters defined in the charmap
612 can be specified using octal, decimal, or hexadecimal constants. Symbolic names not
613 present in the charmap can be specified and are ignored, as specified under item (1)
614 above.

615
616 Note: The <character> symbolic character notation is recommended for use of specifying
617 all characters in a FDCC-set, to facilitate portability of the FDCC-sets, as the coded
618 character set of the application of the FDCC-set may be different from the coded character
619 set of the FDCC-set source. This is also recommended for format effectors in strings, such
620 as in LC_DATE or LC_ADDRESS, where the format effectors are allowed to be stored
621 together with the rest of the string, in a binary string with a different encoding from that
622 of the source FDCC-set.

623 624 **4.1.2 Continuation of lines**

625
626 A line in a specification can be continued by placing an escape character as the last visible
627 graphic character on the line; this continuation character is discarded from the input. The
628 line is continued to the next non-comment line.

629 630 **4.1.3 Names for copy keyword**

631
632 In most of the categories a "copy" keyword is allowed. The name specified with this copy
633 keyword is one of:

- 634 - "i18n" which indicate the "i18n" FDCC-set defined in this specification,
635 - the name of a FDCC-set or POSIX locale registered by the process defined in ISO/IEC
636 15897,
637 - any other name which may be recognized in some local context - not being
638 recommended as an international specification.
639

640 **4.1.4 Pre-category statements**

641
642 In a FDCC-set the following statements can precede category specifications, and they
643 apply to all categories in the specified FDCC-set.
644

645 **4.1.4.1 comment_char**

646
647 The following line in a FDCC-set modifies the comment character. It has the following
648 syntax, starting in column 1:
649

```
650 "comment_char %c\n", <comment_character>
```

651
652 The comment character defaults to the number-sign (#). All examples in this Technical
653 Report use "%" as the <comment_character>, except where otherwise noted. Blank lines
654 and lines containing the <comment_character> in the first position are ignored. In collating
655 statements a <comment_character> occurring where the delimiter ";" may occur,
656 terminates the collating statement.
657

658 **4.1.4.2 escape_char**

659
660 The following line in a FDCC-set modifies the escape character to be used in the text. It
661 has the following syntax, starting in column 1:
662

```
663 "escape_char %c\n", <escape_character>
```

664
665 The escape character is used for representing characters in 4.1.1 and for continuing lines.
666 The escape character defaults to backslash "\". All examples in this Technical Report uses
667 "/" as the escape character, except where otherwise noted.
668

669 **4.1.4.3 repertoiremap**

670
671 The following line in a FDCC-set specifies the name of a repertoiremap used to define the
672 symbolic character names in the FDCC-set. There may be at most one "repertoiremap"
673 line. It has the following syntax, starting in column 1:
674

```
675 "repertoiremap %s\n", <repertoiremap>
```

676
677 The name is one of:

- 678 - "i18nrep" which indicates the "i18nrep" repertoiremap defined in this specification,
679 - the name of a <repertoiremap> registered by the process defined in ISO/IEC 15897,
680 - any other name which may be recognized in some local context - not being
681 recommended as an international specification.
682

683 **4.1.4.4 charmap**

684
685 The following line in a FDCC-set specifies the name of a charmap which may be used

686 with the FDCC-set. It has the following syntax, starting in column 1:

687
688 "charmap %s\n",<charmap>
689

690 This keyword gives a hint on which charmaps a FDCC-set is meant to be supported by.
691 There may be more than one charmap specification useful with a FDCC-set. It is an
692 application's responsibility to decide what charmap specification is to be used with that
693 application.

694

695 The name is one of:

- 696 - the name of a <charmap> registered by the process defined in ISO/IEC 15897,
- 697 - any other name which may be recognized in some local context - not being
- 698 recommended as an international specification.

699

700 4.2 LC_IDENTIFICATION

701

702 The LC_IDENTIFICATION category defines properties of the FDCC-set, and which
703 specification methods the FDCC-set is conforming to. All keywords are mandatory unless
704 otherwise noted, and the operands are strings. The following keywords are defined:

705

706	title	Title of the FDCC-set.
707	source	Organization name of provider of the source.
708	address	Organization postal address.
709	contact	Name of contact person. This keyword is optional.
710	email	Electronic mail address of the organization, or contact
711		person.
712	tel	Telephone number for the organization, in international
713		format.
714	fax	Fax number for the organization, in international format.
715	language	Natural language to which the FDCC-set applies, as specified
716		in ISO 639.
717	territory	The geographic extent where the FDCC-set applies (where
718		applicable), as two-letter form of ISO 3166.
719	audience	If not for general use, an indication of the intended user
720		audience. This keyword is optional.
721	application	If for use of a special application, a description of the
722		application. This keyword is optional.
723	abbreviation	Short name for provider of the source. This keyword is
724		optional.
725	revision	Revision number consisting of digits and zero or more full
726		stops (".").
727	date	Revision date in the format according to this example:
728		"1995-02-05" meaning the 5th of February, 1995.

729

730 If information required for any of the mandatory keywords above is not available, then the
731 corresponding string is an empty string. If required information is not present in ISO 639
732 or ISO 3166, the relevant Maintenance Authority should be approached to get the needed
733 item registered.

734

735 Note: Only one language per territory can be addressed with a single FDCC-set; an
736 additional FDCC-set is required for each additional language for that territory.

737

738 **category** Is used to define that a category is present and what
 739 specification the category is claiming conformance to. The
 740 first operand is a string in double-quotes that describes the
 741 specification that the category is claiming conformance to,
 742 and the following values are defined:
 743 "i18n:2000"
 744 "posix:1993"
 745 The second operand is a string with the category name,
 746 where the category names of clause 4 are defined. More than
 747 one "category" keyword may be given, but only one per
 748 category name.
 749

750 The "i18n" LC_IDENTIFICATION category is:

```

751 LC_IDENTIFICATION
752 % This is the ISO/IEC TR 14652 "i18n" definition for
753 % the LC_IDENTIFICATION category.
754 %
755 title "ISO/IEC TR 14652 i18n FDCC-set"
756 source "ISO/IEC Copyright Office"
757 address "Case postale 56, CH-1211 Geneve 20, Switzerland"
758 contact ""
759 email ""
760 tel ""
761 fax ""
762 language ""
763 territory ""
764 revision "1.0"
765 date "2000-05-21"
766 %
767 category "i18n:2000";LC_IDENTIFICATION
768 category "i18n:2000";LC_CTYPE
769 category "i18n:2000";LC_COLLATE
770 category "i18n:2000";LC_TIME
771 category "i18n:2000";LC_NUMERIC
772 category "i18n:2000";LC_MONETARY
773 category "i18n:2000";LC_MESSAGES
774 category "i18n:2000";LC_NAME
775 category "i18n:2000";LC_ADDRESS
776 category "i18n:2000";LC_TELEPHONE
777
778 END LC_IDENTIFICATION
779
780
781
```

782 4.3 LC_CTYPE

783
 784 The LC_CTYPE category defines character classification, case conversion, character
 785 transformation, and other character attribute mappings. Support for the portable character
 786 set is required.
 787

788 A series of characters in a specification can be represented by the hexadecimal symbolic
 789 ellipsis symbol ".." (two dots), the decimal symbolic ellipses symbols "...." (4 dots), the
 790 double increment hexadecimal symbolic ellipses "..(2).." , or the absolute ellipses "..."
 791 (3 dots).
 792

793 The **hexadecimal symbolic ellipsis** ("..") specification is only valid between symbolic
 794 character names. The symbolic names consists of zero or more nonnumeric characters
 795 from the set shown with visible glyphs in Table 1, followed by an integer formed by one
 796 or more hexadecimal digits, using uppercase letters only for the range "A" to "F". The
 797 characters preceding the hexadecimal integer are identical in the two symbolic names, and

798 the integer formed by the hexadecimal digits in the second symbolic name are identical to
 799 or greater than the integer formed by the hexadecimal digits in the first name. This is
 800 interpreted as a series of symbolic names formed from the common part and each of the
 801 integers in hexadecimal format using uppercase letters only between the first and the
 802 second integer, inclusive, and with a length of the symbolic names generated that is equal
 803 to the length of the first (and also the second) symbolic name. As an example,
 804 <U010E>..
 805 and <U0111>, in that order.

806
 807 The **decimal symbolic ellipsis** ("...") specification is only valid between symbolic
 808 character names. The symbolic names consist of zero or more nonnumeric characters from
 809 the set shown with visible glyphs in Table 1, followed by an integer formed by one or
 810 more decimal digits. The characters preceding the decimal integer are identical in the two
 811 symbolic names, and the integer formed by the decimal digits in the second symbolic
 812 name is identical to or greater than the integer formed by the decimal digits in the first
 813 name. This is interpreted as a series of symbolic names formed from the common part and
 814 each of the integers in decimal format between the first and the second integer, inclusive,
 815 and with a length of the symbolic names generated that is equal to the length of the first
 816 (and also the second) symbolic name. As an example, <j0101>...<j0104> is interpreted as
 817 the symbolic names <j0101>, <j0102>, <j0103>, and <j0104>, in that order.

818
 819 The **double increment hexadecimal symbolic ellipses** ("..(2)..") works like the
 820 hexadecimal symbolic ellipses, but generates only every other of the symbolic character
 821 names. As an example. <U01AC>..(2)..<U01B2> is interpreted as the symbolic character
 822 names <U01AC>, <U01AE>, <U01B0>, and <U01B2>, in that order.

823
 824 The **absolute ellipsis** specification is only valid within a single encoded character set. An
 825 ellipsis is interpreted as including in the list all characters with an encoded value higher
 826 than the encoded value of the character preceding the ellipsis and lower than the encoded
 827 value of the character following the ellipsis. The absolute ellipsis specification is
 828 deprecated, as this is only relevant to FDCC-sets not using symbolic characters.
 829 As an example, \x30;...;\x39 includes in the character class all characters with encoded
 830 values between the endpoints.

831 832 **4.3.1 Character classification keywords**

833
 834 The following keywords are recognized. In the descriptions, the term "automatically
 835 included" means that it is not an error to either include the referenced characters or to
 836 omit them; the interpreting system provides them if missing and accept them silently if
 837 present.

838	
839	copy Specify the name of an existing FDCC-set to be used as the source for the
840	definition of this category. If this keyword is specified, no other keyword is
841	specified.
842	upper Define characters to be classified as uppercase letters. No character
843	specified for the keywords "cntrl", "digit", "punct", or "space" is specified.
844	The uppercase letters A through Z of the portable character set,
845	automatically belong to this class, with application-defined character values.
846	The keyword may be omitted.
847	lower Define characters to be classified as lowercase letters. No character
848	specified for the keywords "cntrl", "digit", "punct", or "space" is specified.
849	The lowercase letters a through z of the portable character set, automatically

850		belong to this class, with application-defined character values. The keyword
851		may be omitted.
852	alpha	Define characters to be classified as used to spell out the words for natural
853		languages; such as letters, syllabic or ideographic characters. No character
854		specified for the keywords "cntrl", "digit", "punct", or "space" is specified.
855		In addition, characters classified as either "upper" or "lower" automatically
856		belong to this class. The keyword may be omitted.
857	digit	Define the characters to be classified as numeric digits. Digits
858		corresponding to the values 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9 can be specified
859		in groups of 10 digits, and in ascending order of the values they represent.
860		The digits of the portable character set are automatically included. If this
861		keyword is not specified, the digits 0 through 9 of the portable character set
862		automatically belong to this class, with application-defined character values.
863		The "digit" keyword is used to specify which characters are accepted as
864		digits in input to an application, such as characters typed in or scanned in
865		from an input text file, and should list digits used with all the scripts
866		supported by the FDCC-set. The keyword may be omitted.
867	alnum	Define the characters to be classified as used to spell out the words for
868		natural languages, and numeric digits. The characters of the "alpha" and
869		"digits" classes are automatically included in this class. The keyword may
870		be omitted.
871	outdigit	Define the characters to be classified as numeric digits for output from an
872		application, such as to a printer or a display or a output text file. Digits
873		corresponding to the values <0>, <1>, <2>, <3>, <4>, <5>, <6>, <7>, <8>,
874		and <9> can be specified, and in ascending order of the values they
875		represent. The intended use is for all places where digits are used for
876		output, including numeric and monetary formatting, and date and time
877		formatting. Only one set of 10 digits may be specified. If this keyword is
878		not specified, the digits 0 through 9 of the portable character set automati-
879		cally belong to this class, with application-defined character values. The
880		keyword may be omitted.
881	blank	Define characters to be classified as "blank" characters. If this keyword is
882		unspecified, the characters <space> and <tab>, with application-defined
883		character values, belong to this character class.
884	space	Define characters to be classified as white-space characters, to find
885		syntactical boundaries. No character specified for the keywords "upper",
886		"lower", "alpha", "digit", "graph", or "xdigit" is specified. If this keyword is
887		not specified, the characters <space>, <form-feed>, <newline>, <carriage-
888		return>, <tab>, and <vertical-tab>, automatically belong to this class, with
889		application-defined character values. Any characters included in the class
890		"blank" are automatically included. The class should not include the NO-
891		BREAK spaces characters <U00A0>, <U2007>, <UFEFF>, as these
892		characters should not be used for word boundaries. The keyword may be
893		omitted.
894	cntrl	Define characters to be classified as control characters. No character
895		specified for the keywords "upper", "lower", "alpha", "digit", "punct",
896		"graph", "print", or "xdigit" is specified. The keyword is specified.
897	punct	Define characters to be classified as punctuation characters. No character
898		specified for the keywords "upper", "lower", "alpha", "digit", "cntrl",
899		"xdigit", or as the <space> character is specified. The keyword is specified.
900	xdigit	Define the characters to be classified as hexadecimal digits. Only the
901		characters defined for the class "digit" are specified, in ascending sequence

902		by numerical value, followed by sets of six characters representing the
903		hexadecimal digits 10 through 15 in ascending order (for example <A>
904		, <C>, <D>, <E>, <F>, <a>, , <c>, <d>, <e>, <f>). If this keyword
905		is not specified, the digits <0> through <9>, the uppercase letters "A"
906		through <F>, and the lowercase letters <a> through <f>, automatically
907		belong to this class, with application-defined character values.
908	graph	Define characters to be classified as printable characters, not including the
909		<space> character. If this keyword is not specified, characters specified for
910		the keywords "upper", "lower", "alpha", "digit", "xdigit", and "punct" belong
911		to this character class. No character specified for the keyword "cntrl" is
912		specified.
913	print	Define characters to be classified as printable characters, including the
914		<space> character. If this keyword is not provided, characters specified for
915		the keywords upper, lower, alpha, digit, xdigit, punct, graph, and the
916		<space> character belong to this character class. No character specified for
917		the keyword "cntrl" is specified.
918	toupper	Define the mapping of lowercase letters to uppercase letters. The operand
919		consists of character pairs, separated by semicolons. The characters in each
920		character pair are separated by a comma and the pair enclosed by paren-
921		theses. The first character in each pair is the lowercase letter, the second the
922		corresponding uppercase letter. Only characters specified for the keywords
923		"lower" and "upper" are specified. If this keyword is not specified, the
924		lowercase letters <a> through <z>, and their corresponding uppercase letters
925		<A> through <Z>, are automatically included, with application-defined
926		character values.
927	tolower	Define the mapping of uppercase letters to lowercase letters. The operand
928		consists of character pairs, separated by semicolons. The characters in each
929		character pair are separated by a comma and the pair enclosed by
930		parentheses. The first character in each pair is the uppercase letter, the
931		second the corresponding lowercase letter. Only characters specified for the
932		keywords "lower" and "upper" are specified. If this keyword is specified,
933		the uppercase letters <A> through <Z>, and their corresponding lowercase
934		letter, are specified. If this keyword is not specified, the mapping is the
935		reverse mapping of the one specified for toupper.
936	class	Define characters to be classified in the class with the name given in the
937		first operand, which is a string. This string only contains characters of the
938		portable character set that either has the string "LETTER" in its description,
939		or is a digit or <hyphen-minus> or <low-line>. The following operands are
940		characters. This keyword is optional. The keyword can only be specified
941		once per named class. The following two names are recognized:
942	combining	Characters to form composite graphic symbols, such
943		as characters listed in ISO/IEC 10646:1993 annex B.1.
944	combining_level3	Characters to form composite graphic symbols, that
945		may also be represented by other characters, such as
946		characters listed in ISO/IEC 10646-1:1993 annex B.2.
947		The class names "upper", "lower", "alpha", "digit", "space", "cntrl", "punct",
948		"graph", "print", "xdigit", and "blank" are taken to mean the classes defined
949		by the respective keywords.
950	width	Define the column width of characters, for example for use of the C
951		function wwidth(). The operands are first a list for characters, possibly
952		using various ellipses, and semicolon separated, then a <colon>, and then
953		the width of these characters given as an unsigned positive integer. Such

954 width-lists separated by <semicolon> may be given for the various widths.
 955 The default value of width of characters in class "cntrl" and class
 956 "combining is 0, else the default value of width is 1. A width for a
 957 character may be overridden by a WIDTH specification in a charmap. This
 958 keyword is optional.

959 **map** Define the mapping of characters. The first operand is a string, defining the
 960 name of the mapping. The string only contains letters, digits and <hyphen-
 961 minus> and <low-line> from the portable character set. The following ope-
 962 rands consist of character pairs, separated by semicolons. The characters in
 963 each character pair are separated by a comma and the pair enclosed by
 964 parentheses. The first character in each pair is the character to map from,
 965 the second the corresponding character to map to. This keyword is optional.
 966 The keyword can only be specified once per named mapping.

967
 968 The mapping names "toupper", and "tolower" are taken to mean the
 969 mapping defined by the respective keywords.

970
 971 Example of use of the "map" keyword:

```
972 map "kana",(<U30AB>,<U304B>);(<U30AC>,<U304C>);(<U30AD>,<U304D>)
```

973
 974 This example introduces a new mapping "kana" that maps three Katakana characters to corresponding Hiragana
 975 characters.
 976

977
 978 Table 2 shows the allowed character class combinations.

979
 980 **Table 2: Valid Character Class Combinations**

Class	upper	lower	alpha	digit	space	cntrl	punct	graph	print	xdigit	blank
upper		+	A	x	x	x	x	A	A	+	x
lower	+		A	x	x	x	x	A	A	+	x
alpha	+	+		x	x	x	x	A	A	+	x
digit	x	x	x		x	x	x	A	A	A	x
space	x	x	x	x		+	*	*	*	x	+
cntrl	x	x	x	x	+		x	x	x	x	+
punct	x	x	x	x	+	x		A	A	x	+
graph	+	+	+	+	+	x	+		A	+	+
print	+	+	+	+	+	x	+	+		+	+
xdigit	+	+	+	+	x	x	x	A	A		x
blank	x	x	x	x	A	+	*	*	*	x	

995
 996 NOTES:

997 Note 1: Explanation of codes:

998 A Automatically included; see text

999 + Permitted

1000 x Mutually exclusive

1001 * See note 2

1002

1003 Note 2: The <space> character, which is part of the "space" and "blank" class, cannot
 1004 belong to "punct" or "graph", but automatically belong to the "print" class. Other "space"
 1005 or "blank" characters can be classified as "punct", "graph", and/or "print".
 1006

1007 **4.3.2 "i18n" LC_CTYPE category**

1008

1009 The "i18n" FDCC-set for the LC_CTYPE is defined as follows:

1010

1011

LC_CTYPE

1012

% The following is the ISO/IEC TR 14652 i18n fdcc-set LC_CTYPE category.

1013

% It covers ISO/IEC 10646-1 including Cor.1 and AMD 1 thru 9

1014

%

1015

% The "upper" class reflects the uppercase characters of class "alpha"

1016

upper /

1017

% TABLE 1 BASIC LATIN/

1018

<U0041>..<U005A>;/

1019

% TABLE 2 LATIN-1 SUPPLEMENT/

1020

<U00C0>..<U00D6>;<U00D8>..<U00DE>;/

1021

% TABLE 3 LATIN EXTENDED-A/

1022

<U0100>..(2)..<U0136>;/

1023

<U0139>..(2)..<U0147>;/

1024

<U014A>..(2)..<U0178>;/

1025

<U0179>..(2)..<U017D>;/

1026

% TABLE 4 LATIN EXTENDED-B/

1027

<U0181>;<U0182>..(2)..<U0186>;<U0187>;/

1028

<U0189>..<U018B>;<U018E>..<U0191>;<U0193>;<U0194>;/

1029

<U0196>..<U0198>;<U019C>;<U019D>;<U019F>;/

1030

<U01A0>..(2)..<U01A4>;/

1031

<U01A7>;<U01A9>;<U01AC>;<U01AE>;<U01AF>;<U01B1>..<U01B3>;/

1032

<U01B5>;<U01B7>;<U01B8>;<U01BC>;<U01C4>;<U01C5>;<U01C7>;<U01C8>;/

1033

<U01CA>;<U01CB>;/

1034

<U01CD>..(2)..<U01DB>;/

1035

<U01DE>..(2)..<U01EE>;/

1036

<U01F1>;<U01F2>;<U01F4>;<U01FA>..(2)..<U01FE>;/

1037

% TABLE 5 LATIN EXTENDED-B/

1038

<U0200>..(2)..<U0216>;/

1039

% TABLE 6 IPA EXTENSIONS/

1040

% TABLE 9 BASIC GREEK/

1041

<U0386>;<U0388>..<U038A>;<U038C>;<U038E>;<U038F>;<U0391>..<U03A1>;/

1042

<U03A3>..<U03AB>;<U03D2>..<U03D4>/

1043

% TABLE 10 GREEK SYMBOLS AND COPTIC/

1044

<U03E2>..(2)..<U03EE>;/

1045

% TABLE 11 CYRILLIC/

1046

<U0401>..<U040C>;<U040E>..<U042F>;<U0460>..(2)..<U047E>;/

1047

% TABLE 12 CYRILLIC/

1048

<U0480>;<U0490>..(2)..<U04BE>;<U04C1>;<U04C3>;<U04C7>;<U04CB>;/

1049

<U04D0>..(2)..<U04EA>;<U04EE>..(2)..<U04F4>;<U04F8>;/

1050

% TABLE 13 ARMENIAN/

1051

<U0531>..<U0556>;/

1052

% TABLE 28 GEORGIAN/

1053

<U10A0>..<U10C5>;/

1054

% TABLE 31 LATIN EXTENDED ADDITIONAL/

1055

<U1E00>..(2)..<U1E7E>;/

1056

% TABLE 32 LATIN EXTENDED ADDITIONAL/

1057

<U1E80>..(2)..<U1E94>;/

1058

<U1EA0>..(2)..<U1EF8>;/

1059

% TABLE 33 GREEK EXTENDED/

1060

<U1F08>..<U1F0F>;<U1F18>..<U1F1D>;<U1F28>..<U1F2F>;<U1F38>..<U1F3F>;/

1061

<U1F48>..<U1F4D>;<U1F59>..(2)..<U1F5F>;<U1F68>..<U1F6F>;/

1062

% TABLE 34 GREEK EXTENDED/

1063

<U1F88>..<U1F8F>;<U1F98>..<U1F9F>;<U1FA8>..<U1FAF>;<U1FB8>..<U1FBC>;/

1064

<U1FC8>..<U1FCC>;<U1FD8>..<U1FDB>;<U1FE8>..<U1FEC>;<U1FF8>..<U1FFC>

1065

% TABLE 28 GEORGIAN is not addressed as the letters does not have

1066

a uppercase/lowercase relation

1067

%

1068

% The "lower" class reflects the lowercase characters of class "alpha"

1069

lower /

1070

% TABLE 1 BASIC LATIN/

1071

<U0061>..<U007A>;/

1072

% TABLE 2 LATIN-1 SUPPLEMENT/

1073

<U00DF>..<U00F6>;<U00F8>..<U00FF>;/

1074

% TABLE 3 LATIN EXTENDED-A/

1075

<U0101>..(2)..<U0137>;<U0138>..(2)..<U0148>;/

1076

<U0149>..(2)..<U0177>;<U017A>..(2)..<U017E>;<U017F>;/

1077

% TABLE 4 LATIN EXTENDED-B/

1078

<U0180>;<U0183>;<U0185>;<U0188>;<U018C>;<U018D>;<U0192>;<U0195>;/

1079

<U0199>..<U019B>;<U019E>;<U01A1>;<U01A3>;<U01A5>;<U01A8>;<U01AB>;<U01AD>;/

1080

<U01B0>;<U01B4>;<U01B6>;<U01B9>;<U01BA>;<U01BD>;<U01C5>;<U01C6>;/

1081

<U01C8>;<U01C9>;<U01CB>;<U01CC>..(2)..<U01DC>;/

1082

<U01DD>..(2)..<U01F1>;<U01F3>;<U01F5>;<U01FB>;<U01FD>;<U01FF>;/

```

1083 % TABLE 5 LATIN EXTENDED-B/
1084 <U0201>..(2)..<U0217>;/
1085 % TABLE 6 IPA EXTENSIONS/
1086 <U0250>..<U0293>;<U0299>..<U02A0>;<U02A3>..<U02A8>;/
1087 % TABLE 9 BASIC GREEK/
1088 <U0390>;<U03AC>..<U03CE>;/
1089 % TABLE 10 GREEK SYMBOLS AND COPTIC/
1090 <U03E2>..(2)..<U03EE>;/
1091 % TABLE 11 CYRILLIC/
1092 <U0430>..<U044F>;<U0451>..<U045C>;<U045E>;<U045F>;<U0460>..(2)..<U047F>;/
1093 % TABLE 12 CYRILLIC/
1094 <U04801>;<U0490>..(2)..<U04BF>;<U04C2>;<U04C4>;<U04C8>;<U04CC>;/
1095 <U04D1>..(2)..<U04EB>;<U04EF>..(2)..<U04F5>;<U04F9>;/
1096 % TABLE 13 ARMENIAN/
1097 <U0561>..<U0587>;/
1098 % TABLE 28 GEORGIAN/
1099 <U10D0>..<U10F6>;/
1100 % TABLE 31 and 32 LATIN EXTENDED ADDITIONAL/
1101 <U1E01>..(2)..<U1E95>;<U1EA1>..(2)..<U1EF9>;/
1102 % TABLE 33 and 34 GREEK EXTENDED/
1103 <U1F08>..<U1F0F>;<U1F18>..<U1F1D>;<U1F28>..<U1F2F>;<U1F38>..<U1F3F>;/
1104 <U1F48>..<U1F4D>;<U1F59>..(2)..<U1F5F>;<U1F68>..<U1F6F>;/
1105 % TABLE 34 GREEK EXTENDED/
1106 <U1F00>..<U1F07>;<U1F10>..<U1F15>;<U1F20>..<U1F27>;<U1F30>..<U1F37>;/
1107 <U1F40>..<U1F45>;<U1F50>..<U1F57>;<U1F60>..<U1F67>;<U1F70>..<U1F7D>;/
1108 <U1F80>..<U1F87>;<U1F90>..<U1F97>;<U1FA0>..<U1FA7>;<U1FB0>..<U1FB4>;/
1109 <U1FB6>;<U1FB7>;<U1FC2>..<U1FC4>;<U1FC6>;<U1FC7>;<U1FD0>..<U1FD3>;/
1110 <U1FD6>;<U1FD7>;<U1FE0>..<U1FE7>;<U1FF2>..<U1FF4>;<U1FF6>;<U1FF7>;/
1111 % TABLE 35 SUPERSCRIPTS AND SUBSCRIPTS, CURRENCY SYMBOLS/
1112 <U207F>
1113 %
1114 % The "alpha" class of the "i18n" FDCC-set is reflecting
1115 % the recommendations in TR 10176 annex A
1116 alpha /
1117 % TABLE 1 BASIC LATIN/
1118 <U0041>..<U005A>;<U0061>..<U007A>;/
1119 % TABLE 2 LATIN-1 SUPPLEMENT/
1120 <U00AA>;<U00BA>;<U00C0>..<U00D6>;<U00D8>..<U00F6>;<U00F8>..<U00FF>;/
1121 % TABLE 3 LATIN EXTENDED-A/
1122 <U0100>..<U017F>;/
1123 % TABLE 4 and 5 LATIN EXTENDED-B/
1124 <U0180>..<U01F5>;<U01FA>..<U0217>;/
1125 % TABLE 6 IPA EXTENSIONS/
1126 <U0250>..<U02A8>;/
1127 % TABLE 31 and 32 LATIN EXTENDED ADDITIONAL/
1128 <U1E00>..<U1E9B>;<U1EA0>..<U1EF9>;/
1129 % TABLE 35 SUPERSCRIPTS AND SUBSCRIPTS, CURRENCY SYMBOLS/
1130 <U207F>;/
1131 % TABLE 9 BASIC GREEK/
1132 <U0386>;<U0388>..<U038A>;<U038C>;<U038E>..<U03A1>;<U03A3>..<U03CE>;/
1133 % TABLE 10 GREEK SYMBOLS AND COPTIC/
1134 <U03D0>..<U03D6>;<U03DA>;<U03DC>;<U03DE>;<U03E0>;<U03E2>..<U03F3>;/
1135 % TABLE 33 and 34 GREEK EXTENDED/
1136 <U1F00>..<U1F15>;<U1F18>..<U1F1D>;<U1F20>..<U1F45>;<U1F48>..<U1F4D>;/
1137 <U1F50>..<U1F57>;<U1F59>;<U1F5B>;<U1F5D>;<U1F5F>..<U1F7D>;/
1138 <U1F80>..<U1FB4>;<U1FB6>..<U1FBC>;<U1FC2>..<U1FC4>;<U1FC6>..<U1FCC>;/
1139 <U1FD0>..<U1FD3>;<U1FD6>..<U1FDB>;<U1FE0>..<U1FEC>;<U1FF2>..<U1FF4>;/
1140 <U1FF6>..<U1FFC>;/
1141 % TABLE 11 and 12 CYRILLIC/
1142 <U0401>..<U040C>;<U040E>..<U044F>;<U0451>..<U045C>;<U045E>..<U0481>;/
1143 <U0490>..<U04C4>;<U04C7>..<U04C8>;<U04CB>..<U04CC>;<U04D0>..<U04EB>;/
1144 <U04EE>..<U04F5>;<U04F8>..<U04F9>;/
1145 % TABLE 13 ARMENIAN/
1146 <U0531>..<U0556>;<U0561>..<U0587>;/
1147 % TABLE 14 HEBREW/
1148 <U05B0>..<U05B9>;<U05BB>..<U05BD>;<U05BF>;<U05C1>..<U05C2>;/
1149 <U05D0>..<U05EA>;<U05F0>..<U05F2>;/
1150 % TABLE 15 and 16 ARABIC/
1151 <U0621>..<U063A>;<U0641>..<U064A>;<U0670>..<U06B7>;<U06BA>..<U06BE>;/
1152 <U06C0>..<U06CE>;<U06D0>..<U06D3>;<U06D5>..<U06DC>;<U06E5>..<U06E8>;/
1153 % TABLE 17 DEVANAGARI/
1154 <U0901>..<U0903>;<U0905>..<U0939>;<U093E>..<U094D>;<U0950>..<U0952>;/
1155 <U0958>..<U0963>;/
1156 % TABLE 18 BENGALI/
1157 <U0981>..<U0983>;<U0985>..<U098C>;<U098F>..<U0990>;/
1158 <U0993>..<U09A8>;<U09AA>..<U09B0>;<U09B2>;<U09B6>..<U09B9>;/
1159 <U09BE>..<U09C4>;<U09C7>..<U09C8>;<U09CB>..<U09CD>;<U09DC>..<U09DD>;/
1160 <U09DF>..<U09E3>;<U09F0>..<U09F1>;/

```

```

1161 % TABLE 19 GURMUKHI/
1162 <U0A02>;<U0A05>..<<U0A0A>;<U0A0F>..<<U0A10>;<U0A13>..<<U0A28>;/
1163 <U0A2A>..<<U0A30>;<U0A32>..<<U0A33>;<U0A35>..<<U0A36>;<U0A38>..<<U0A39>;/
1164 <U0A3E>..<<U0A42>;<U0A47>..<<U0A48>;<U0A4B>..<<U0A4D>;<U0A59>..<<U0A5C>;/
1165 <U0A5E>;<U0A74>;/
1166 % TABLE 20 GUJARATI/
1167 <U0A81>..<<U0A83>;<U0A85>..<<U0A8B>;<U0A8D>;<U0A8F>..<<U0A91>;/
1168 <U0A93>..<<U0AA8>;<U0AAA>..<<U0AB0>;<U0AB2>..<<U0AB3>;<U0AB5>..<<U0AB9>;/
1169 <U0ABD>..<<U0AC5>;<U0AC7>..<<U0AC9>;<U0ACB>..<<U0ACD>;<U0AD0>;<U0AE0>;/
1170 % TABLE 21 ORIYA/
1171 <U0B01>..<<U0B03>;<U0B05>..<<U0B0C>;<U0B0F>..<<U0B10>;<U0B13>..<<U0B28>;/
1172 <U0B2A>..<<U0B30>;<U0B32>..<<U0B33>;<U0B36>..<<U0B39>;<U0B3E>..<<U0B43>;/
1173 <U0B47>..<<U0B48>;<U0B4B>..<<U0B4D>;<U0B5C>..<<U0B5D>;<U0B5F>..<<U0B61>;/
1174 % TABLE 22 TAMIL/
1175 <U0B82>..<<U0B83>;<U0B85>..<<U0B8A>;<U0B8E>..<<U0B90>;<U0B92>..<<U0B95>;/
1176 <U0B99>..<<U0B9A>;<U0B9C>;<U0B9E>..<<U0B9F>;<U0BA3>..<<U0BA4>;/
1177 <U0BA8>..<<U0BAA>;<U0BAE>..<<U0BB5>;<U0BB7>..<<U0BB9>;<U0BBE>..<<U0BC2>;/
1178 <U0BC6>..<<U0BC8>;<U0BCA>..<<U0BCD>;/
1179 % TABLE 23 TELUGU/
1180 <U0C01>..<<U0C03>;<U0C05>..<<U0C0C>;<U0C0E>..<<U0C10>;<U0C12>..<<U0C28>;/
1181 <U0C2A>..<<U0C33>;<U0C35>..<<U0C39>;<U0C3E>..<<U0C44>;<U0C46>..<<U0C48>;/
1182 <U0C4A>..<<U0C4D>;<U0C60>..<<U0C61>;/
1183 % TABLE 24 KANNADA/
1184 <U0C82>..<<U0C83>;<U0C85>..<<U0C8C>;<U0C8E>..<<U0C90>;<U0C92>..<<U0CA8>;/
1185 <U0CAA>..<<U0CB3>;<U0CB5>..<<U0CB9>;<U0CBE>..<<U0CC4>;<U0CC6>..<<U0CC8>;/
1186 <U0CCA>..<<U0CCD>;<U0CDE>;<U0CE0>..<<U0CE1>;/
1187 % TABLE 25 MALAYALAM/
1188 <U0D02>..<<U0D03>;<U0D05>..<<U0D0C>;<U0D0E>..<<U0D10>;<U0D12>..<<U0D28>;/
1189 <U0D2A>..<<U0D39>;<U0D3E>..<<U0D43>;<U0D46>..<<U0D48>;<U0D4A>..<<U0D4D>;/
1190 <U0D60>..<<U0D61>;/
1191 % TABLE 26 THAI/
1192 <U0E01>..<<U0E3A>;<U0E40>..<<U0E4E>;/
1193 % TABLE 27 LAO/
1194 <U0E81>..<<U0E82>;<U0E84>;<U0E87>..<<U0E88>;<U0E8A>;<U0E8D>;/
1195 <U0E94>..<<U0E97>;<U0E99>..<<U0E9F>;<U0EA1>..<<U0EA3>;<U0EA5>;<U0EA7>;/
1196 <U0EAA>..<<U0EAB>;<U0EAD>..<<U0EAE>;<U0EB0>..<<U0EB9>;<U0EBB>..<<U0EBD>;/
1197 <U0EC0>..<<U0EC4>;<U0EC6>;<U0EC8>..<<U0ECD>;<U0EDC>..<<U0EDD>;/
1198 % TIBETAN Amendment 6/
1199 <U0F00>;<U0F18>..<<U0F19>;<U0F35>;<U0F37>;<U0F39>;<U0F40>..<<U0F47>;/
1200 <U0F49>..<<U0F69>;/
1201 <U0F71>..<<U0F84>;<U0F86>..<<U0F8B>;<U0F90>..<<U0F95>;<U0F97>;/
1202 <U0F99>..<<U0FAD>;<U0FB1>..<<U0FB7>;<U0FB9>;/
1203 % TABLE 28 GEORGIAN/
1204 <U10A0>..<<U10C5>;<U10D0>..<<U10F6>;/
1205 % TABLE 50 HIRAGANA/
1206 <U3041>..<<U3093>;<U309B>..<<U309C>;/
1207 % TABLE 51 KATAKANA/
1208 <U30A1>..<<U30F6>;<U30FB>..<<U30FC>;/
1209 % TABLE 52 BOPOMOFO/
1210 <U3105>..<<U312C>;/
1211 % CJK unified ideographs/
1212 <U4E00>..<<U9FA5>;/
1213 % HANGUL amendment 5/
1214 <UAC00>..<<UD7A3>;/
1215 % Miscellaneous/
1216 <U00B5>;<U02B0>..<<U02B8>;<U02BB>;<U02BD>..<<U02C1>;/
1217 <U02D0>..<<U02D1>;<U02E0>..<<U02E4>;<U037A>;<U0559>;<U093D>;<U0B3D>;/
1218 <U1FBF>;<U2160>..<<U2182>;<U3021>..<<U3029>
1219 %
1220 % The "digit" class of the "i18n" FDCC-set is reflecting
1221 % the recommendations in TR 10176 annex A
1222 digit /
1223 % TABLE 1 BASIC LATIN/
1224 <U0030>..<<U0039>;/
1225 % TABLE 15 and 16 ARABIC/
1226 <U0660>..<<U0669>;<U06F0>..<<U06F9>;/
1227 % TABLE 17 DEVANAGARI/
1228 <U0966>..<<U096F>;/
1229 % TABLE 18 BENGALI/
1230 <U09E6>..<<U09EF>;/
1231 % TABLE 19 GURMUKHI/
1232 <U0A66>..<<U0A6F>;/
1233 % TABLE 20 GUJARATI/
1234 <U0AE6>..<<U0AEF>;/
1235 % TABLE 21 ORIYA/
1236 <U0B66>..<<U0B6F>;/
1237 % TABLE 22 TAMIL/
1238 <0>;<U0BE7>..<<U0BEF>;/

```



```

1239 % TABLE 23 TELUGU/
1240 <U0C66>..

```

```

1317 <U0BC6>..<U0BC8>;<U0BCA>..<U0BCD>;<U0BD7>;<U0BE7>..<U0BF2>;<U0C01>..<U0C03>;/
1318 <U0C05>..<U0C0C>;<U0C0E>..<U0C10>;<U0C12>..<U0C28>;<U0C2A>..<U0C33>;/
1319 <U0C35>..<U0C39>;<U0C3E>..<U0C44>;<U0C46>..<U0C48>;<U0C4A>..<U0C4D>;/
1320 <U0C55>..<U0C56>;<U0C60>;<U0C61>;<U0C66>..<U0C6F>;<U0C82>;<U0C83>;/
1321 <U0C85>..<U0C8C>;<U0C8E>..<U0C90>;<U0C92>..<U0CA8>;<U0CAA>..<U0CB3>;/
1322 <U0CB5>..<U0CB9>;<U0CBE>..<U0CC4>;<U0CC6>..<U0CC8>;<U0CCA>..<U0CCD>;/
1323 <U0CD5>;<U0CD6>;<U0CDE>;<U0CE0>;<U0CE1>;<U0CE6>..<U0CEF>;<U0D02>;<U0D03>;/
1324 <U0D05>..<U0D0C>;<U0D0E>..<U0D10>;<U0D12>..<U0D28>;<U0D2A>..<U0D39>;/
1325 <U0D3E>..<U0D43>;<U0D46>..<U0D48>;<U0D4A>..<U0D4D>;<U0D57>;<U0D60>;<U0D61>;/
1326 <U0D66>..<U0D6F>;<U0E01>..<U0E3A>;<U0E3F>..<U0E5B>;<U0E81>;<U0E82>;<U0E84>;/
1327 <U0E87>;<U0E88>;<U0E8A>;<U0E8D>;<U0E94>..<U0E97>;<U0E99>..<U0E9F>;/
1328 <U0EA1>..<U0EA3>;<U0EA5>;<U0EA7>;<U0EAA>;<U0EAB>;<U0EAD>..<U0EB9>;/
1329 <U0EBB>..<U0EBD>;<U0EC0>..<U0EC4>;<U0EC6>;<U0EC8>..<U0ECD>;<U0ED0>..<U0ED9>;/
1330 <U0EDC>;<U0EDD>;/
1331 <U0F00>..<U0F47>;<U0F49>..<U0F69>;<U0F71>..<U0F7F>;/
1332 <U10A0>..<U10C5>;<U10D0>..<U10F6>;<U10FB>;<U1100>..<U1159>;/
1333 <U115F>..<U11A2>;<U11A8>..<U11F9>;<U1E00>..<U1E9B>;<U1EA0>..<U1EF9>;/
1334 <U1F00>..<U1F15>;<U1F18>..<U1F1D>;<U1F20>..<U1F45>;<U1F48>..<U1F4D>;/
1335 <U1F50>..<U1F57>;<U1F59>;<U1F5B>;<U1F5D>;<U1F5F>..<U1F7D>;<U1F80>..<U1FB4>;/
1336 <U1FB6>..<U1FC4>;<U1FC5>..<U1FD3>;<U1FD6>..<U1FDB>;<U1FDD>..<U1FEF>;/
1337 <U1FF2>..<U1FF4>;<U1FF6>..<U1FFE>;<U2000>..<U202E>;<U2030>..<U2046>;/
1338 <U206A>..<U2070>;<U2074>..<U208E>;<U20A0>..<U20AB>;<U20D0>..<U20E1>;/
1339 <U2100>..<U2138>;<U2153>..<U2182>;<U2190>..<U21EA>;<U2200>..<U22F1>;<U2300>;/
1340 <U2302>..<U237A>;<U2400>..<U2424>;<U2440>..<U244A>;<U2460>..<U24EA>;/
1341 <U2500>..<U2595>;<U25A0>..<U25EF>;<U2600>..<U2613>;<U261A>..<U266F>;/
1342 <U2701>..<U2704>;<U2706>..<U2709>;<U270C>..<U2727>;<U2729>..<U274B>;<U274D>;/
1343 <U274F>..<U2752>;<U2756>;<U2758>..<U275E>;<U2761>..<U2767>;<U2776>..<U2794>;/
1344 <U2798>..<U27AF>;<U27B1>..<U27BE>;<U3000>..<U3037>;<U303F>;<U3041>..<U3094>;/
1345 <U3099>..<U309E>;<U30A1>..<U30FE>;<U3105>..<U312C>;<U3131>..<U318E>;/
1346 <U3190>..<U319F>;<U3200>..<U321C>;<U3220>..<U3243>;<U3260>..<U327B>;/
1347 <U327F>..<U32B0>;<U32C0>..<U32CB>;<U32D0>..<U32FE>;<U3300>..<U3376>;/
1348 <U337B>..<U33DD>;<U33E0>..<U33FE>;<UFB00>..<UFB06>;<UFB13>..<UFB17>;/
1349 <UFB1E>..<UFB36>;<UFB38>..<UFB3C>;<UFB3E>;<UFB40>;<UFB41>;<UFB43>;<UFB44>;/
1350 <UFB46>..<UFB81>;<UFB83>..<UFD3F>;<UFD50>..<UFD8F>;<UFD92>..<UFD9C>;/
1351 <UFD9E>..<UFD9F>;<UFE20>..<UFE23>;<UFE30>..<UFE44>;<UFE49>..<UFE52>;/
1352 <UFE54>..<UFE66>;<UFE68>..<UFE6B>;<UFE70>..<UFE72>;<UFE74>;<UFE76>..<UFEFC>;/
1353 <UFEFF>;<UFF01>..<UFF5E>;<UFF61>..<UFFBE>;<UFFC2>..<UFFC7>;/
1354 <UFFCA>..<UFFCF>;<UFFD2>..<UFFD7>;<UFFDA>..<UFFDC>;<UFFE0>..<UFFE6>;/
1355 <UFFE8>..<UFFEE>;<UFFFD>
1356 %
1357 % "print" is by default "graph", and the <space> character
1358 %
1359 %
1360 %
1361 %
1362 %
1363 %
1364 %
1365 %
1366 %
1367 %
1368 %
1369 %
1370 %
1371 %
1372 %
1373 %
1374 %
1375 %
1376 %
1377 %
1378 %
1379 %
1380 %
1381 %
1382 %
1383 %
1384 %
1385 %
1386 %
1387 %
1388 %
1389 %
1390 %
1391 %
1392 %
1393 %
1394 %

```

1395 (<U0199>, <U0198>); (<U01A1>, <U01A0>); (<U01A3>, <U01A2>); (<U01A5>, <U01A4>); /
 1396 (<U01A8>, <U01A7>); (<U01AD>, <U01AC>); (<U01B0>, <U01AF>); (<U01B4>, <U01B3>); /
 1397 (<U01B6>, <U01B5>); (<U01B9>, <U01B8>); (<U01BD>, <U01BC>); (<U01C5>, <U01C4>); /
 1398 (<U01C6>, <U01C4>); (<U01C8>, <U01C7>); /
 1399 (<U01C9>, <U01C7>); (<U01CB>, <U01CA>); (<U01CC>, <U01CA>); /
 1400 (<U01CE>, <U01CD>); (<U01D0>, <U01CF>); (<U01D2>, <U01D1>); (<U01D4>, <U01D3>); /
 1401 (<U01D6>, <U01D5>); (<U01D8>, <U01D7>); (<U01DA>, <U01D9>); (<U01DC>, <U01DB>); /
 1402 (<U01DD>, <U018E>); (<U01DF>, <U01DE>); (<U01E1>, <U01E0>); (<U01E3>, <U01E2>); /
 1403 (<U01E5>, <U01E4>); (<U01E7>, <U01E6>); (<U01E9>, <U01E8>); (<U01EB>, <U01EA>); /
 1404 (<U01ED>, <U01EC>); (<U01EF>, <U01EE>); (<U01F2>, <U01F1>); /
 1405 (<U01F3>, <U01F1>); (<U01F5>, <U01F4>); (<U01FB>, <U01FA>); (<U01FD>, <U01FC>); /
 1406 (<U01FF>, <U01FE>); (<U0201>, <U0200>); (<U0203>, <U0202>); (<U0205>, <U0204>); /
 1407 (<U0207>, <U0206>); (<U0209>, <U0208>); (<U020B>, <U020A>); (<U020D>, <U020C>); /
 1408 (<U020F>, <U020E>); (<U0211>, <U0210>); (<U0213>, <U0212>); (<U0215>, <U0214>); /
 1409 (<U0217>, <U0216>); (<U0253>, <U0181>); (<U0254>, <U0186>); (<U0256>, <U0189>); /
 1410 (<U0257>, <U018A>); (<U0259>, <U018F>); (<U025B>, <U0190>); (<U0260>, <U0193>); /
 1411 (<U0263>, <U0194>); (<U0268>, <U0197>); (<U0269>, <U0196>); (<U026F>, <U019C>); /
 1412 (<U0272>, <U019D>); (<U0275>, <U019F>); (<U0283>, <U01A9>); (<U0288>, <U01AE>); /
 1413 (<U028A>, <U01B1>); (<U028B>, <U01B2>); (<U0292>, <U01B7>); (<U03AC>, <U0386>); /
 1414 (<U03AD>, <U038E>); (<U03AE>, <U0389>); (<U03AF>, <U038A>); (<U03B1>, <U0391>); /
 1415 (<U03B2>, <U0392>); (<U03B3>, <U0393>); (<U03B4>, <U0394>); (<U03B5>, <U0395>); /
 1416 (<U03B6>, <U0396>); (<U03B7>, <U0397>); (<U03B8>, <U0398>); (<U03B9>, <U0399>); /
 1417 (<U03BA>, <U039A>); (<U03BB>, <U039B>); (<U03BC>, <U039C>); (<U03BD>, <U039D>); /
 1418 (<U03BE>, <U039E>); (<U03BF>, <U039F>); (<U03C0>, <U03A0>); (<U03C1>, <U03A1>); /
 1419 (<U03C2>, <U03A3>); (<U03C3>, <U03A3>); (<U03C4>, <U03A4>); (<U03C5>, <U03A5>); /
 1420 (<U03C6>, <U03A6>); (<U03C7>, <U03A7>); (<U03C8>, <U03A8>); (<U03C9>, <U03A9>); /
 1421 (<U03CA>, <U03AA>); (<U03CB>, <U03AB>); (<U03CC>, <U038C>); (<U03CD>, <U038E>); /
 1422 (<U03CE>, <U038F>); /
 1423 (<U03E3>, <U03E2>); (<U03E5>, <U03E4>); (<U03E7>, <U03E6>); (<U03E9>, <U03E8>); /
 1424 (<U03EB>, <U03EA>); (<U03ED>, <U03EC>); (<U03EF>, <U03EE>); /
 1425 (<U0430>, <U0410>); (<U0431>, <U0411>); (<U0432>, <U0412>); /
 1426 (<U0433>, <U0413>); (<U0434>, <U0414>); (<U0435>, <U0415>); (<U0436>, <U0416>); /
 1427 (<U0437>, <U0417>); (<U0438>, <U0418>); (<U0439>, <U0419>); (<U043A>, <U041A>); /
 1428 (<U043B>, <U041B>); (<U043C>, <U041C>); (<U043D>, <U041D>); (<U043E>, <U041E>); /
 1429 (<U043F>, <U041F>); (<U0440>, <U0420>); (<U0441>, <U0421>); (<U0442>, <U0422>); /
 1430 (<U0443>, <U0423>); (<U0444>, <U0424>); (<U0445>, <U0425>); (<U0446>, <U0426>); /
 1431 (<U0447>, <U0427>); (<U0448>, <U0428>); (<U0449>, <U0429>); (<U044A>, <U042A>); /
 1432 (<U044B>, <U042B>); (<U044C>, <U042C>); (<U044D>, <U042D>); (<U044E>, <U042E>); /
 1433 (<U044F>, <U042F>); (<U0451>, <U0401>); (<U0452>, <U0402>); (<U0453>, <U0403>); /
 1434 (<U0454>, <U0404>); (<U0455>, <U0405>); (<U0456>, <U0406>); (<U0457>, <U0407>); /
 1435 (<U0458>, <U0408>); (<U0459>, <U0409>); (<U045A>, <U040A>); (<U045B>, <U040B>); /
 1436 (<U045C>, <U040C>); (<U045E>, <U040E>); (<U045F>, <U040F>); (<U0461>, <U0460>); /
 1437 (<U0463>, <U0462>); (<U0465>, <U0464>); (<U0467>, <U0466>); (<U0469>, <U0468>); /
 1438 (<U046B>, <U046A>); (<U046D>, <U046C>); (<U046F>, <U046E>); (<U0471>, <U0470>); /
 1439 (<U0473>, <U0472>); (<U0475>, <U0474>); (<U0477>, <U0476>); (<U0479>, <U0478>); /
 1440 (<U047B>, <U047A>); (<U047D>, <U047C>); (<U047F>, <U047E>); (<U0481>, <U0480>); /
 1441 (<U0491>, <U0490>); (<U0493>, <U0492>); (<U0495>, <U0494>); (<U0497>, <U0496>); /
 1442 (<U0499>, <U0498>); (<U049B>, <U049A>); (<U049D>, <U049C>); (<U049F>, <U049E>); /
 1443 (<U04A1>, <U04A0>); (<U04A3>, <U04A2>); (<U04A5>, <U04A4>); (<U04A7>, <U04A6>); /
 1444 (<U04A9>, <U04A8>); (<U04AB>, <U04AA>); (<U04AD>, <U04AC>); (<U04AF>, <U04AE>); /
 1445 (<U04B1>, <U04B0>); (<U04B3>, <U04B2>); (<U04B5>, <U04B4>); (<U04B7>, <U04B6>); /
 1446 (<U04B9>, <U04B8>); (<U04BB>, <U04BA>); (<U04BD>, <U04BC>); (<U04BF>, <U04BE>); /
 1447 (<U04C2>, <U04C1>); (<U04C4>, <U04C3>); (<U04C8>, <U04C7>); (<U04CC>, <U04CB>); /
 1448 (<U04D1>, <U04D0>); (<U04D3>, <U04D2>); (<U04D5>, <U04D4>); (<U04D7>, <U04D6>); /
 1449 (<U04D9>, <U04D8>); (<U04DB>, <U04DA>); (<U04DD>, <U04DC>); (<U04DF>, <U04DE>); /
 1450 (<U04E1>, <U04E0>); (<U04E3>, <U04E2>); (<U04E5>, <U04E4>); (<U04E7>, <U04E6>); /
 1451 (<U04E9>, <U04E8>); (<U04EB>, <U04EA>); (<U04EF>, <U04EE>); (<U04F1>, <U04F0>); /
 1452 (<U04F3>, <U04F2>); (<U04F5>, <U04F4>); (<U04F9>, <U04F8>); (<U0561>, <U0531>); /
 1453 (<U0562>, <U0532>); (<U0563>, <U0533>); (<U0564>, <U0534>); (<U0565>, <U0535>); /
 1454 (<U0566>, <U0536>); (<U0567>, <U0537>); (<U0568>, <U0538>); (<U0569>, <U0539>); /
 1455 (<U056A>, <U053A>); (<U056B>, <U053B>); (<U056C>, <U053C>); (<U056D>, <U053D>); /
 1456 (<U056E>, <U053E>); (<U056F>, <U053F>); (<U0570>, <U0540>); (<U0571>, <U0541>); /
 1457 (<U0572>, <U0542>); (<U0573>, <U0543>); (<U0574>, <U0544>); (<U0575>, <U0545>); /
 1458 (<U0576>, <U0546>); (<U0577>, <U0547>); (<U0578>, <U0548>); (<U0579>, <U0549>); /
 1459 (<U057A>, <U054A>); (<U057B>, <U054B>); (<U057C>, <U054C>); (<U057D>, <U054D>); /
 1460 (<U057E>, <U054E>); (<U057F>, <U054F>); (<U0580>, <U0550>); (<U0581>, <U0551>); /
 1461 (<U0582>, <U0552>); (<U0583>, <U0553>); (<U0584>, <U0554>); (<U0585>, <U0555>); /
 1462 (<U0586>, <U0556>); /
 1463 (<U10D0>, <U10A0>); (<U10D1>, <U10A1>); (<U10D2>, <U10A2>); (<U10D3>, <U10A3>); /
 1464 (<U10D4>, <U10A4>); (<U10D5>, <U10A5>); (<U10D6>, <U10A6>); (<U10D7>, <U10A7>); /
 1465 (<U10D8>, <U10A8>); (<U10D9>, <U10A9>); (<U10DA>, <U10AA>); (<U10DB>, <U10AB>); /
 1466 (<U10DC>, <U10AC>); (<U10DD>, <U10AD>); (<U10DE>, <U10AE>); (<U10DF>, <U10AF>); /
 1467 (<U10E0>, <U10B0>); (<U10E1>, <U10B1>); (<U10E2>, <U10B2>); (<U10E3>, <U10B3>); /
 1468 (<U10E4>, <U10B4>); (<U10E5>, <U10B5>); (<U10E6>, <U10B6>); (<U10E7>, <U10B7>); /
 1469 (<U10E8>, <U10B8>); (<U10E9>, <U10B9>); (<U10EA>, <U10BA>); (<U10EB>, <U10BB>); /
 1470 (<U10EC>, <U10BC>); (<U10ED>, <U10BD>); (<U10EE>, <U10BE>); (<U10EF>, <U10BF>); /
 1471 (<U10F0>, <U10C0>); (<U10F1>, <U10C1>); (<U10F2>, <U10C2>); (<U10F3>, <U10C3>); /
 1472 (<U10F4>, <U10C4>); (<U10F5>, <U10C5>); /

1473 (<U1E01>, <U1E00>); (<U1E03>, <U1E02>); (<U1E05>, <U1E04>); /
 1474 (<U1E07>, <U1E06>); (<U1E09>, <U1E08>); (<U1E0B>, <U1E0A>); (<U1E0D>, <U1E0C>); /
 1475 (<U1E0F>, <U1E0E>); (<U1E11>, <U1E10>); (<U1E13>, <U1E12>); (<U1E15>, <U1E14>); /
 1476 (<U1E17>, <U1E16>); (<U1E19>, <U1E18>); (<U1E1B>, <U1E1A>); (<U1E1D>, <U1E1C>); /
 1477 (<U1E1F>, <U1E1E>); (<U1E21>, <U1E20>); (<U1E23>, <U1E22>); (<U1E25>, <U1E24>); /
 1478 (<U1E27>, <U1E26>); (<U1E29>, <U1E28>); (<U1E2B>, <U1E2A>); (<U1E2D>, <U1E2C>); /
 1479 (<U1E2F>, <U1E2E>); (<U1E31>, <U1E30>); (<U1E33>, <U1E32>); (<U1E35>, <U1E34>); /
 1480 (<U1E37>, <U1E36>); (<U1E39>, <U1E38>); (<U1E3B>, <U1E3A>); (<U1E3D>, <U1E3C>); /
 1481 (<U1E3F>, <U1E3E>); (<U1E41>, <U1E40>); (<U1E43>, <U1E42>); (<U1E45>, <U1E44>); /
 1482 (<U1E47>, <U1E46>); (<U1E49>, <U1E48>); (<U1E4B>, <U1E4A>); (<U1E4D>, <U1E4C>); /
 1483 (<U1E4F>, <U1E4E>); (<U1E51>, <U1E50>); (<U1E53>, <U1E52>); (<U1E55>, <U1E54>); /
 1484 (<U1E57>, <U1E56>); (<U1E59>, <U1E58>); (<U1E5B>, <U1E5A>); (<U1E5D>, <U1E5C>); /
 1485 (<U1E5F>, <U1E5E>); (<U1E61>, <U1E60>); (<U1E63>, <U1E62>); (<U1E65>, <U1E64>); /
 1486 (<U1E67>, <U1E66>); (<U1E69>, <U1E68>); (<U1E6B>, <U1E6A>); (<U1E6D>, <U1E6C>); /
 1487 (<U1E6F>, <U1E6E>); (<U1E71>, <U1E70>); (<U1E73>, <U1E72>); (<U1E75>, <U1E74>); /
 1488 (<U1E77>, <U1E76>); (<U1E79>, <U1E78>); (<U1E7B>, <U1E7A>); (<U1E7D>, <U1E7C>); /
 1489 (<U1E7F>, <U1E7E>); (<U1E81>, <U1E80>); (<U1E83>, <U1E82>); (<U1E85>, <U1E84>); /
 1490 (<U1E87>, <U1E86>); (<U1E89>, <U1E88>); (<U1E8B>, <U1E8A>); (<U1E8D>, <U1E8C>); /
 1491 (<U1E8F>, <U1E8E>); (<U1E91>, <U1E90>); (<U1E93>, <U1E92>); (<U1E95>, <U1E94>); /
 1492 (<U1E9B>, <U1E9A>); /
 1493 (<U1EA1>, <U1EA0>); (<U1EA3>, <U1EA2>); (<U1EA5>, <U1EA4>); (<U1EA7>, <U1EA6>); /
 1494 (<U1EA9>, <U1EA8>); (<U1EAB>, <U1EAA>); (<U1EAD>, <U1EAC>); (<U1EAF>, <U1EAE>); /
 1495 (<U1EB1>, <U1EB0>); (<U1EB3>, <U1EB2>); (<U1EB5>, <U1EB4>); (<U1EB7>, <U1EB6>); /
 1496 (<U1EB9>, <U1EB8>); (<U1EBB>, <U1EBA>); (<U1EBD>, <U1EBC>); (<U1EBF>, <U1EBE>); /
 1497 (<U1EC1>, <U1EC0>); (<U1EC3>, <U1EC2>); (<U1EC5>, <U1EC4>); (<U1EC7>, <U1EC6>); /
 1498 (<U1EC9>, <U1EC8>); (<U1ECB>, <U1ECA>); (<U1ECD>, <U1ECC>); (<U1ECF>, <U1ECE>); /
 1499 (<U1ED1>, <U1ED0>); (<U1ED3>, <U1ED2>); (<U1ED5>, <U1ED4>); (<U1ED7>, <U1ED6>); /
 1500 (<U1ED9>, <U1ED8>); (<U1EDB>, <U1EDA>); (<U1EDD>, <U1EDC>); (<U1EDF>, <U1EDE>); /
 1501 (<U1EE1>, <U1EE0>); (<U1EE3>, <U1EE2>); (<U1EE5>, <U1EE4>); (<U1EE7>, <U1EE6>); /
 1502 (<U1EE9>, <U1EE8>); (<U1EEB>, <U1EEA>); (<U1EED>, <U1EEC>); (<U1EEF>, <U1EEE>); /
 1503 (<U1EF1>, <U1EF0>); (<U1EF3>, <U1EF2>); (<U1EF5>, <U1EF4>); (<U1EF7>, <U1EF6>); /
 1504 (<U1EF9>, <U1EF8>); (<U1EF00>, <U1EF08>); (<U1EF01>, <U1EF09>); (<U1EF02>, <U1EF0A>); /
 1505 (<U1F03>, <U1F0B>); (<U1F04>, <U1F0C>); (<U1F05>, <U1F0D>); (<U1F06>, <U1F0E>); /
 1506 (<U1F07>, <U1F0F>); (<U1F10>, <U1F18>); (<U1F11>, <U1F19>); (<U1F12>, <U1F1A>); /
 1507 (<U1F13>, <U1F1B>); (<U1F14>, <U1F1C>); (<U1F15>, <U1F1D>); (<U1F20>, <U1F28>); /
 1508 (<U1F21>, <U1F29>); (<U1F22>, <U1F2A>); (<U1F23>, <U1F2B>); (<U1F24>, <U1F2C>); /
 1509 (<U1F25>, <U1F2D>); (<U1F26>, <U1F2E>); (<U1F27>, <U1F2F>); (<U1F30>, <U1F38>); /
 1510 (<U1F31>, <U1F39>); (<U1F32>, <U1F3A>); (<U1F33>, <U1F3B>); (<U1F34>, <U1F3C>); /
 1511 (<U1F35>, <U1F3D>); (<U1F36>, <U1F3E>); (<U1F37>, <U1F3F>); (<U1F40>, <U1F48>); /
 1512 (<U1F41>, <U1F49>); (<U1F42>, <U1F4A>); (<U1F43>, <U1F4B>); (<U1F44>, <U1F4C>); /
 1513 (<U1F45>, <U1F4D>); (<U1F51>, <U1F59>); (<U1F53>, <U1F5B>); (<U1F55>, <U1F5D>); /
 1514 (<U1F57>, <U1F5F>); (<U1F60>, <U1F68>); (<U1F61>, <U1F69>); (<U1F62>, <U1F6A>); /
 1515 (<U1F63>, <U1F6B>); (<U1F64>, <U1F6C>); (<U1F65>, <U1F6D>); (<U1F66>, <U1F6E>); /
 1516 (<U1F67>, <U1F6F>); (<U1F70>, <U1F7A>); (<U1F71>, <U1F7B>); (<U1F72>, <U1F7C>); /
 1517 (<U1F73>, <U1F7D>); (<U1F74>, <U1F7E>); (<U1F75>, <U1F7F>); (<U1F76>, <U1F78>); /
 1518 (<U1F77>, <U1F79>); (<U1F78>, <U1F7A>); (<U1F79>, <U1F7B>); (<U1F7A>, <U1F7C>); /
 1519 (<U1F7B>, <U1F7D>); (<U1F7C>, <U1F7E>); (<U1F7D>, <U1F7F>); (<U1F7E>, <U1F78>); /
 1520 (<U1F81>, <U1F89>); (<U1F82>, <U1F8A>); (<U1F83>, <U1F8B>); (<U1F84>, <U1F8C>); /
 1521 (<U1F85>, <U1F8D>); (<U1F86>, <U1F8E>); (<U1F87>, <U1F8F>); (<U1F90>, <U1F98>); /
 1522 (<U1F91>, <U1F99>); (<U1F92>, <U1F9A>); (<U1F93>, <U1F9B>); (<U1F94>, <U1F9C>); /
 1523 (<U1F95>, <U1F9D>); (<U1F96>, <U1F9E>); (<U1F97>, <U1F9F>); (<U1FA0>, <U1FA8>); /
 1524 (<U1FA1>, <U1FA9>); (<U1FA2>, <U1FAA>); (<U1FA3>, <U1FAB>); (<U1FA4>, <U1FAC>); /
 1525 (<U1FA5>, <U1FAD>); (<U1FA6>, <U1FAE>); (<U1FA7>, <U1FAF>); (<U1FB0>, <U1FB8>); /
 1526 (<U1FB1>, <U1FB9>); (<U1FB3>, <U1FBC>); (<U1FC3>, <U1FCC>); (<U1FD0>, <U1FD8>); /
 1527 (<U1FD1>, <U1FD9>); (<U1FE0>, <U1FE8>); (<U1FE1>, <U1FE9>); (<U1FE5>, <U1FEC>); /
 1528 (<U1FF3>, <U1FFC>)
 1529 tolower /
 1530 (<U0041>, <U0061>); (<U0042>, <U0062>); (<U0043>, <U0063>); (<U0044>, <U0064>); /
 1531 (<U0045>, <U0065>); (<U0046>, <U0066>); (<U0047>, <U0067>); (<U0048>, <U0068>); /
 1532 (<U0049>, <U0069>); (<U004A>, <U006A>); (<U004B>, <U006B>); (<U004C>, <U006C>); /
 1533 (<U004D>, <U006D>); (<U004E>, <U006E>); (<U004F>, <U006F>); (<U0050>, <U0070>); /
 1534 (<U0051>, <U0071>); (<U0052>, <U0072>); (<U0053>, <U0073>); (<U0054>, <U0074>); /
 1535 (<U0055>, <U0075>); (<U0056>, <U0076>); (<U0057>, <U0077>); (<U0058>, <U0078>); /
 1536 (<U0059>, <U0079>); (<U005A>, <U007A>); (<U00C0>, <U00E0>); (<U00C1>, <U00E1>); /
 1537 (<U00C2>, <U00E2>); (<U00C3>, <U00E3>); (<U00C4>, <U00E4>); (<U00C5>, <U00E5>); /
 1538 (<U00C6>, <U00E6>); (<U00C7>, <U00E7>); (<U00C8>, <U00E8>); (<U00C9>, <U00E9>); /
 1539 (<U00CA>, <U00EA>); (<U00CB>, <U00EB>); (<U00CC>, <U00EC>); (<U00CD>, <U00ED>); /
 1540 (<U00CE>, <U00EE>); (<U00CF>, <U00EF>); (<U00D0>, <U00F0>); (<U00D1>, <U00F1>); /
 1541 (<U00D2>, <U00F2>); (<U00D3>, <U00F3>); (<U00D4>, <U00F4>); (<U00D5>, <U00F5>); /
 1542 (<U00D6>, <U00F6>); (<U00D8>, <U00F8>); (<U00D9>, <U00F9>); (<U00DA>, <U00FA>); /
 1543 (<U00DB>, <U00FB>); (<U00DC>, <U00FC>); (<U00DD>, <U00FD>); (<U00DE>, <U00FE>); /
 1544 (<U0178>, <U00FF>); (<U0100>, <U0101>); (<U0102>, <U0103>); (<U0104>, <U0105>); /
 1545 (<U0106>, <U0107>); (<U0108>, <U0109>); (<U010A>, <U010B>); (<U010C>, <U010D>); /
 1546 (<U010E>, <U010F>); (<U0110>, <U0111>); (<U0112>, <U0113>); (<U0114>, <U0115>); /
 1547 (<U0116>, <U0117>); (<U0118>, <U0119>); (<U011A>, <U011B>); (<U011C>, <U011D>); /
 1548 (<U011E>, <U011F>); (<U0120>, <U0121>); (<U0122>, <U0123>); (<U0124>, <U0125>); /
 1549 (<U0126>, <U0127>); (<U0128>, <U0129>); (<U012A>, <U012B>); (<U012C>, <U012D>); /
 1550 (<U012E>, <U012F>); (<U0132>, <U0133>); (<U0134>, <U0135>); (<U0136>, <U0137>); /

1551 (<U0139>, <U013A>); (<U013B>, <U013C>); (<U013D>, <U013E>); (<U013F>, <U0140>); /
 1552 (<U0141>, <U0142>); (<U0143>, <U0144>); (<U0145>, <U0146>); (<U0147>, <U0148>); /
 1553 (<U014A>, <U014B>); (<U014C>, <U014D>); (<U014E>, <U014F>); (<U0150>, <U0151>); /
 1554 (<U0152>, <U0153>); (<U0154>, <U0155>); (<U0156>, <U0157>); (<U0158>, <U0159>); /
 1555 (<U015A>, <U015B>); (<U015C>, <U015D>); (<U015E>, <U015F>); (<U0160>, <U0161>); /
 1556 (<U0162>, <U0163>); (<U0164>, <U0165>); (<U0166>, <U0167>); (<U0168>, <U0169>); /
 1557 (<U016A>, <U016B>); (<U016C>, <U016D>); (<U016E>, <U016F>); (<U0170>, <U0171>); /
 1558 (<U0172>, <U0173>); (<U0174>, <U0175>); (<U0176>, <U0177>); (<U0179>, <U017A>); /
 1559 (<U017B>, <U017C>); (<U017D>, <U017E>); (<U0182>, <U0183>); (<U0184>, <U0185>); /
 1560 (<U0187>, <U0188>); (<U0256>, <U0189>); (<U018B>, <U018C>); (<U018E>, <U01DD>); /
 1561 (<U0191>, <U0192>); (<U0198>, <U0199>); (<U01A0>, <U01A1>); (<U01A2>, <U01A3>); /
 1562 (<U01A4>, <U01A5>); (<U01A7>, <U01A8>); (<U01AC>, <U01AD>); (<U01AF>, <U01B0>); /
 1563 (<U01B3>, <U01B4>); (<U01B5>, <U01B6>); (<U01B8>, <U01B9>); (<U01BC>, <U01BD>); /
 1564 (<U01C4>, <U01C6>); (<U01C5>, <U01C6>); (<U01C7>, <U01C9>); /
 1565 (<U01C8>, <U01C9>); (<U01CA>, <U01CC>); (<U01CB>, <U01CC>); /
 1566 (<U01CD>, <U01CE>); (<U01CF>, <U01D0>); (<U01D1>, <U01D2>); /
 1567 (<U01D3>, <U01D4>); (<U01D5>, <U01D6>); (<U01D7>, <U01D8>); (<U01D9>, <U01DA>); /
 1568 (<U01DB>, <U01DC>); (<U01DE>, <U01DF>); (<U01E0>, <U01E1>); (<U01E2>, <U01E3>); /
 1569 (<U01E4>, <U01E5>); (<U01E6>, <U01E7>); (<U01E8>, <U01E9>); (<U01EA>, <U01EB>); /
 1570 (<U01EC>, <U01ED>); (<U01EE>, <U01EF>); (<U01F1>, <U01F3>); /
 1571 (<U01F2>, <U01F3>); (<U01F4>, <U01F5>); (<U01FA>, <U01FB>); (<U01FC>, <U01FD>); /
 1572 (<U01FE>, <U01FF>); (<U0200>, <U0201>); (<U0202>, <U0203>); (<U0204>, <U0205>); /
 1573 (<U0206>, <U0207>); (<U0208>, <U0209>); (<U020A>, <U020B>); (<U020C>, <U020D>); /
 1574 (<U020E>, <U020F>); (<U0210>, <U0211>); (<U0212>, <U0213>); (<U0214>, <U0215>); /
 1575 (<U0216>, <U0217>); (<U0181>, <U0253>); (<U0186>, <U0254>); (<U018A>, <U0257>); /
 1576 (<U018E>, <U0258>); (<U018F>, <U0259>); (<U0190>, <U025B>); (<U0193>, <U0260>); /
 1577 (<U0194>, <U0263>); (<U0197>, <U0268>); (<U0196>, <U0269>); (<U019C>, <U026F>); /
 1578 (<U019D>, <U0272>); (<U01A9>, <U0283>); (<U01AE>, <U0288>); (<U01B1>, <U028A>); /
 1579 (<U01B2>, <U028B>); (<U01B7>, <U0292>); (<U0386>, <U03AC>); (<U0388>, <U03AD>); /
 1580 (<U0389>, <U03AE>); (<U038A>, <U03AF>); (<U0391>, <U03B1>); (<U0392>, <U03B2>); /
 1581 (<U0393>, <U03B3>); (<U0394>, <U03B4>); (<U0395>, <U03B5>); (<U0396>, <U03B6>); /
 1582 (<U0397>, <U03B7>); (<U0398>, <U03B8>); (<U0399>, <U03B9>); (<U039A>, <U03BA>); /
 1583 (<U039B>, <U03BB>); (<U039C>, <U03BC>); (<U039D>, <U03BD>); (<U039E>, <U03BE>); /
 1584 (<U039F>, <U03BF>); (<U03A0>, <U03C0>); (<U03A1>, <U03C1>); (<U03A3>, <U03C3>); /
 1585 (<U03A4>, <U03C4>); (<U03A5>, <U03C5>); (<U03A6>, <U03C6>); (<U03A7>, <U03C7>); /
 1586 (<U03A8>, <U03C8>); (<U03A9>, <U03C9>); (<U03AA>, <U03CA>); (<U03AB>, <U03CB>); /
 1587 (<U038C>, <U03CC>); (<U038E>, <U03CD>); (<U038F>, <U03CE>); (<U03E2>, <U03E3>); /
 1588 (<U03E4>, <U03E5>); (<U03E6>, <U03E7>); (<U03E8>, <U03E9>); (<U03EA>, <U03EB>); /
 1589 (<U03EC>, <U03ED>); (<U03EE>, <U03EF>); (<U0410>, <U0430>); /
 1590 (<U0411>, <U0431>); (<U0412>, <U0432>); (<U0413>, <U0433>); (<U0414>, <U0434>); /
 1591 (<U0415>, <U0435>); (<U0416>, <U0436>); (<U0417>, <U0437>); (<U0418>, <U0438>); /
 1592 (<U0419>, <U0439>); (<U041A>, <U043A>); (<U041B>, <U043B>); (<U041C>, <U043C>); /
 1593 (<U041D>, <U043D>); (<U041E>, <U043E>); (<U041F>, <U043F>); (<U0420>, <U0440>); /
 1594 (<U0421>, <U0441>); (<U0422>, <U0442>); (<U0423>, <U0443>); (<U0424>, <U0444>); /
 1595 (<U0425>, <U0445>); (<U0426>, <U0446>); (<U0427>, <U0447>); (<U0428>, <U0448>); /
 1596 (<U0429>, <U0449>); (<U042A>, <U044A>); (<U042B>, <U044B>); (<U042C>, <U044C>); /
 1597 (<U042D>, <U044D>); (<U042E>, <U044E>); (<U042F>, <U044F>); (<U0401>, <U0451>); /
 1598 (<U0402>, <U0452>); (<U0403>, <U0453>); (<U0404>, <U0454>); (<U0405>, <U0455>); /
 1599 (<U0406>, <U0456>); (<U0407>, <U0457>); (<U0408>, <U0458>); (<U0409>, <U0459>); /
 1600 (<U040A>, <U045A>); (<U040B>, <U045B>); (<U040C>, <U045C>); (<U040E>, <U045E>); /
 1601 (<U040F>, <U045F>); (<U0460>, <U0461>); (<U0462>, <U0463>); (<U0464>, <U0465>); /
 1602 (<U0466>, <U0467>); (<U0468>, <U0469>); (<U046A>, <U046B>); (<U046C>, <U046D>); /
 1603 (<U046E>, <U046F>); (<U0470>, <U0471>); (<U0472>, <U0473>); (<U0474>, <U0475>); /
 1604 (<U0476>, <U0477>); (<U0478>, <U0479>); (<U047A>, <U047B>); (<U047C>, <U047D>); /
 1605 (<U047E>, <U047F>); (<U0480>, <U0481>); (<U0490>, <U0491>); (<U0492>, <U0493>); /
 1606 (<U0494>, <U0495>); (<U0496>, <U0497>); (<U0498>, <U0499>); (<U049A>, <U049B>); /
 1607 (<U049C>, <U049D>); (<U049E>, <U049F>); (<U04A0>, <U04A1>); (<U04A2>, <U04A3>); /
 1608 (<U04A4>, <U04A5>); (<U04A6>, <U04A7>); (<U04A8>, <U04A9>); (<U04AA>, <U04AB>); /
 1609 (<U04AC>, <U04AD>); (<U04AE>, <U04AF>); (<U04B0>, <U04B1>); (<U04B2>, <U04B3>); /
 1610 (<U04B4>, <U04B5>); (<U04B6>, <U04B7>); (<U04B8>, <U04B9>); (<U04BA>, <U04BB>); /
 1611 (<U04BC>, <U04BD>); (<U04BE>, <U04BF>); (<U04C1>, <U04C2>); (<U04C3>, <U04C4>); /
 1612 (<U04C7>, <U04C8>); (<U04CB>, <U04CC>); (<U04D0>, <U04D1>); (<U04D2>, <U04D3>); /
 1613 (<U04D4>, <U04D5>); (<U04D6>, <U04D7>); (<U04D8>, <U04D9>); (<U04DA>, <U04DB>); /
 1614 (<U04DC>, <U04DD>); (<U04DE>, <U04DF>); (<U04E0>, <U04E1>); (<U04E2>, <U04E3>); /
 1615 (<U04E4>, <U04E5>); (<U04E6>, <U04E7>); (<U04E8>, <U04E9>); (<U04EA>, <U04EB>); /
 1616 (<U04EE>, <U04EF>); (<U04F0>, <U04F1>); (<U04F2>, <U04F3>); (<U04F4>, <U04F5>); /
 1617 (<U04F8>, <U04F9>); (<U0531>, <U0561>); (<U0532>, <U0562>); (<U0533>, <U0563>); /
 1618 (<U0534>, <U0564>); (<U0535>, <U0565>); (<U0536>, <U0566>); (<U0537>, <U0567>); /
 1619 (<U0538>, <U0568>); (<U0539>, <U0569>); (<U053A>, <U056A>); (<U053B>, <U056B>); /
 1620 (<U053C>, <U056C>); (<U053D>, <U056D>); (<U053E>, <U056E>); (<U053F>, <U056F>); /
 1621 (<U0540>, <U0570>); (<U0541>, <U0571>); (<U0542>, <U0572>); (<U0543>, <U0573>); /
 1622 (<U0544>, <U0574>); (<U0545>, <U0575>); (<U0546>, <U0576>); (<U0547>, <U0577>); /
 1623 (<U0548>, <U0578>); (<U0549>, <U0579>); (<U054A>, <U057A>); (<U054B>, <U057B>); /
 1624 (<U054C>, <U057C>); (<U054D>, <U057D>); (<U054E>, <U057E>); (<U054F>, <U057F>); /
 1625 (<U0550>, <U0580>); (<U0551>, <U0581>); (<U0552>, <U0582>); (<U0553>, <U0583>); /
 1626 (<U0554>, <U0584>); (<U0555>, <U0585>); (<U0556>, <U0586>); /
 1627 (<U10A0>, <U10D0>); (<U10A1>, <U10D1>); (<U10A2>, <U10D2>); (<U10A3>, <U10D3>); /
 1628 (<U10A4>, <U10D4>); (<U10A5>, <U10D5>); (<U10A6>, <U10D6>); (<U10A7>, <U10D7>); /

```

1629 (<U10A8>, <U10D8>); (<U10A9>, <U10D9>); (<U10AA>, <U10DA>); (<U10AB>, <U10DB>); /
1630 (<U10AC>, <U10DC>); (<U10AD>, <U10DD>); (<U10AE>, <U10DE>); (<U10AF>, <U10DF>); /
1631 (<U10B0>, <U10E0>); (<U10B1>, <U10E1>); (<U10B2>, <U10E2>); (<U10B3>, <U10E3>); /
1632 (<U10B4>, <U10E4>); (<U10B5>, <U10E5>); (<U10B6>, <U10E6>); (<U10B7>, <U10E7>); /
1633 (<U10B8>, <U10E8>); (<U10B9>, <U10E9>); (<U10BA>, <U10EA>); (<U10BB>, <U10EB>); /
1634 (<U10BC>, <U10EC>); (<U10BD>, <U10ED>); (<U10BE>, <U10EE>); (<U10BF>, <U10EF>); /
1635 (<U10C0>, <U10F0>); (<U10C1>, <U10F1>); (<U10C2>, <U10F2>); (<U10C3>, <U10F3>); /
1636 (<U10C4>, <U10F4>); (<U10C5>, <U10F5>); /
1637 (<U1E00>, <U1E01>); /
1638 (<U1E02>, <U1E03>); (<U1E04>, <U1E05>); (<U1E06>, <U1E07>); (<U1E08>, <U1E09>); /
1639 (<U1E0A>, <U1E0B>); (<U1E0C>, <U1E0D>); (<U1E0E>, <U1E0F>); (<U1E10>, <U1E11>); /
1640 (<U1E12>, <U1E13>); (<U1E14>, <U1E15>); (<U1E16>, <U1E17>); (<U1E18>, <U1E19>); /
1641 (<U1E1A>, <U1E1B>); (<U1E1C>, <U1E1D>); (<U1E1E>, <U1E1F>); (<U1E20>, <U1E21>); /
1642 (<U1E22>, <U1E23>); (<U1E24>, <U1E25>); (<U1E26>, <U1E27>); (<U1E28>, <U1E29>); /
1643 (<U1E2A>, <U1E2B>); (<U1E2C>, <U1E2D>); (<U1E2E>, <U1E2F>); (<U1E30>, <U1E31>); /
1644 (<U1E32>, <U1E33>); (<U1E34>, <U1E35>); (<U1E36>, <U1E37>); (<U1E38>, <U1E39>); /
1645 (<U1E3A>, <U1E3B>); (<U1E3C>, <U1E3D>); (<U1E3E>, <U1E3F>); (<U1E40>, <U1E41>); /
1646 (<U1E42>, <U1E43>); (<U1E44>, <U1E45>); (<U1E46>, <U1E47>); (<U1E48>, <U1E49>); /
1647 (<U1E4A>, <U1E4B>); (<U1E4C>, <U1E4D>); (<U1E4E>, <U1E4F>); (<U1E50>, <U1E51>); /
1648 (<U1E52>, <U1E53>); (<U1E54>, <U1E55>); (<U1E56>, <U1E57>); (<U1E58>, <U1E59>); /
1649 (<U1E5A>, <U1E5B>); (<U1E5C>, <U1E5D>); (<U1E5E>, <U1E5F>); (<U1E60>, <U1E61>); /
1650 (<U1E62>, <U1E63>); (<U1E64>, <U1E65>); (<U1E66>, <U1E67>); (<U1E68>, <U1E69>); /
1651 (<U1E6A>, <U1E6B>); (<U1E6C>, <U1E6D>); (<U1E6E>, <U1E6F>); (<U1E70>, <U1E71>); /
1652 (<U1E72>, <U1E73>); (<U1E74>, <U1E75>); (<U1E76>, <U1E77>); (<U1E78>, <U1E79>); /
1653 (<U1E7A>, <U1E7B>); (<U1E7C>, <U1E7D>); (<U1E7E>, <U1E7F>); (<U1E80>, <U1E81>); /
1654 (<U1E82>, <U1E83>); (<U1E84>, <U1E85>); (<U1E86>, <U1E87>); (<U1E88>, <U1E89>); /
1655 (<U1E8A>, <U1E8B>); (<U1E8C>, <U1E8D>); (<U1E8E>, <U1E8F>); (<U1E90>, <U1E91>); /
1656 (<U1E92>, <U1E93>); (<U1E94>, <U1E95>); (<U1EA0>, <U1EA1>); (<U1EA2>, <U1EA3>); /
1657 (<U1EA4>, <U1EA5>); (<U1EA6>, <U1EA7>); (<U1EA8>, <U1EA9>); (<U1EAA>, <U1EAB>); /
1658 (<U1EAC>, <U1EAD>); (<U1EAE>, <U1EAF>); (<U1EB0>, <U1EB1>); (<U1EB2>, <U1EB3>); /
1659 (<U1EB4>, <U1EB5>); (<U1EB6>, <U1EB7>); (<U1EB8>, <U1EB9>); (<U1EBA>, <U1EBB>); /
1660 (<U1EBC>, <U1EBD>); (<U1EBE>, <U1EBF>); (<U1EC0>, <U1EC1>); (<U1EC2>, <U1EC3>); /
1661 (<U1EC4>, <U1EC5>); (<U1EC6>, <U1EC7>); (<U1EC8>, <U1EC9>); (<U1ECA>, <U1ECB>); /
1662 (<U1ECC>, <U1ECD>); (<U1ECE>, <U1ECF>); (<U1ED0>, <U1ED1>); (<U1ED2>, <U1ED3>); /
1663 (<U1ED4>, <U1ED5>); (<U1ED6>, <U1ED7>); (<U1ED8>, <U1ED9>); (<U1EDA>, <U1EDB>); /
1664 (<U1EDC>, <U1EDD>); (<U1EDE>, <U1EDF>); (<U1EE0>, <U1EE1>); (<U1EE2>, <U1EE3>); /
1665 (<U1EE4>, <U1EE5>); (<U1EE6>, <U1EE7>); (<U1EE8>, <U1EE9>); (<U1EEA>, <U1EEB>); /
1666 (<U1EEC>, <U1EED>); (<U1EEE>, <U1EEF>); (<U1EF0>, <U1EF1>); (<U1EF2>, <U1EF3>); /
1667 (<U1EF4>, <U1EF5>); (<U1EF6>, <U1EF7>); (<U1EF8>, <U1EF9>); (<U1F00>, <U1F00>); /
1668 (<U1F09>, <U1F01>); (<U1F0A>, <U1F02>); (<U1F0B>, <U1F03>); (<U1F0C>, <U1F04>); /
1669 (<U1F0D>, <U1F05>); (<U1F0E>, <U1F06>); (<U1F0F>, <U1F07>); (<U1F18>, <U1F10>); /
1670 (<U1F19>, <U1F11>); (<U1F1A>, <U1F12>); (<U1F1B>, <U1F13>); (<U1F1C>, <U1F14>); /
1671 (<U1F1D>, <U1F15>); (<U1F18>, <U1F20>); (<U1F19>, <U1F21>); (<U1F2A>, <U1F22>); /
1672 (<U1F2B>, <U1F23>); (<U1F2C>, <U1F24>); (<U1F2D>, <U1F25>); (<U1F2E>, <U1F26>); /
1673 (<U1F2F>, <U1F27>); (<U1F38>, <U1F30>); (<U1F39>, <U1F31>); (<U1F3A>, <U1F32>); /
1674 (<U1F3B>, <U1F33>); (<U1F3C>, <U1F34>); (<U1F3D>, <U1F35>); (<U1F3E>, <U1F36>); /
1675 (<U1F3F>, <U1F37>); (<U1F48>, <U1F40>); (<U1F49>, <U1F41>); (<U1F4A>, <U1F42>); /
1676 (<U1F4B>, <U1F43>); (<U1F4C>, <U1F44>); (<U1F4D>, <U1F45>); (<U1F59>, <U1F51>); /
1677 (<U1F5B>, <U1F53>); (<U1F5D>, <U1F55>); (<U1F5F>, <U1F57>); (<U1F68>, <U1F60>); /
1678 (<U1F69>, <U1F61>); (<U1F6A>, <U1F62>); (<U1F6B>, <U1F63>); (<U1F6C>, <U1F64>); /
1679 (<U1F6D>, <U1F65>); (<U1F6E>, <U1F66>); (<U1F6F>, <U1F67>); (<U1F8A>, <U1F70>); /
1680 (<U1FBB>, <U1F71>); (<U1FC8>, <U1F72>); (<U1FC9>, <U1F73>); (<U1FCA>, <U1F74>); /
1681 (<U1FCB>, <U1F75>); (<U1FDA>, <U1F76>); (<U1FDB>, <U1F77>); (<U1FF8>, <U1F78>); /
1682 (<U1FF9>, <U1F79>); (<U1FEA>, <U1F7A>); (<U1FEB>, <U1F7B>); (<U1FFA>, <U1F7C>); /
1683 (<U1FFB>, <U1F7D>); (<U1F88>, <U1F80>); (<U1F89>, <U1F81>); (<U1F8A>, <U1F82>); /
1684 (<U1F8B>, <U1F83>); (<U1F8C>, <U1F84>); (<U1F8D>, <U1F85>); (<U1F8E>, <U1F86>); /
1685 (<U1F8F>, <U1F87>); (<U1F98>, <U1F90>); (<U1F99>, <U1F91>); (<U1F9A>, <U1F92>); /
1686 (<U1F9B>, <U1F93>); (<U1F9C>, <U1F94>); (<U1F9D>, <U1F95>); (<U1F9E>, <U1F96>); /
1687 (<U1F9F>, <U1F97>); (<U1FA8>, <U1FA0>); (<U1FA9>, <U1FA1>); (<U1FAA>, <U1FA2>); /
1688 (<U1FAB>, <U1FA3>); (<U1FAC>, <U1FA4>); (<U1FAD>, <U1FA5>); (<U1FAE>, <U1FA6>); /
1689 (<U1FAF>, <U1FA7>); (<U1FB8>, <U1FB0>); (<U1FB9>, <U1FB1>); (<U1FBC>, <U1FB3>); /
1690 (<U1FCC>, <U1FC3>); (<U1FD8>, <U1FD0>); (<U1FD9>, <U1FD1>); (<U1FE8>, <U1FE0>); /
1691 (<U1FE9>, <U1FE1>); (<U1FEC>, <U1FE5>); (<U1FFC>, <U1FF3>); /
1692 %
1693 % The "combining" class reflects ISO/IEC 10646-1 annex B.1
1694 % That is, all combining characters (level 2+3).
1695 class "combining" /
1696 <U0300>..<U036F>; <U20D0>..<U20FF>; <UFE20>..<UFE2F>; /
1697 <U0483>..<U0486>; <U0591>..<U05A1>; <U05A3>..<U05B9>; /
1698 <U05BB>..<U05BD>; <U05BF>; <U05C1>; <U05C2>; <U05C4>; <U064B>..<U0652>; <U0670>; /
1699 <U06D7>..<U06E4>; <U06E7>; <U06E8>; <U06EA>..<U06ED>; <U0901>..<U0903>; <U093C>; /
1700 <U093E>..<U094D>; <U0951>..<U0954>; <U0962>; <U0963>; <U0981>..<U0983>; <U09BC>; /
1701 <U09BE>..<U09C4>; <U09C7>; <U09C8>; <U09CB>..<U09CD>; <U09D7>; <U09E2>; <U09E3>; /
1702 <U0A02>; <U0A3C>; <U0A3E>..<U0A42>; <U0A47>; <U0A48>; <U0A4B>..<U0A4D>; /
1703 <U0A70>; <U0A71>; <U0A83>; <U06EA>; <U06BE>..<U0AC5>; <U0AC7>..<U0AC9>; /
1704 <U0ACB>..<U0ACD>; <U0B01>..<U0B03>; <U0B3C>; <U0B3E>..<U0B43>; <U0B47>; <U0B48>; /
1705 <U0B4B>..<U0B4D>; <U0B56>; <U0B57>; <U0B82>; <U0B83>; <U0BBE>..<U0BC2>; /
1706 <U0BC6>..<U0BC8>; <U0BCA>..<U0BCD>; <U0BD7>; <U0C01>..<U0C03>; <U0C3E>..<U0C44>; /

```

```

1707     <U0C46>..<U0C48>;<U0C4A>..<U0C4D>;<U0C55>;<U0C56>;<U0C82>;<U0C83>;/
1708     <U0CBE>..<U0CC4>;<U0CC6>..<U0CC8>;<U0CCA>..<U0CCD>;<U0CD5>;<U0CD6>;/
1709     <U0D02>;<U0D03>;<U0D3E>..<U0D43>;<U0D46>..<U0D48>;<U0D4A>..<U0D4D>;<U0D57>;/
1710     <U0E31>;<U0E34>..<U0E3A>;<U0E47>..<U0E4E>;<U0EB1>;<U0EB4>..<U0EB9>;/
1711     <U0EBB>;<U0EBC>;<U0EC8>..<U0ECD>;<U0F18>;<U0F19>;<U0F35>;<U0F37>;<U0F39>;/
1712     <U0F3E>;<U0F3F>;<U0F71>..<U0F84>;<U0F86>..<U0F89>;<U0F8B>;<U0F90>..<U0F95>;/
1713     <U0F97>;<U0F99>..<U0FAD>;<U0FB1>..<U0FB7>;<U0FB9>;<U302A>..<U302F>;/
1714     <U3099>;<U309A>;<UFB1E>
1715
1716     %
1717     % The "combining_level3" class reflects ISO/IEC 10646-1 annex B.2
1718     % That is, combining characters of level 3.
1719     class      "combining_level3"; /
1720     <U0300>..<U036F>;<U20D0>..<U20FF>;<U1100>..<U11FF>;<UFE20>..<UFE2F>;/
1721     <U0483>..<U0486>;<U0591>..<U05A1>;<U05A3>..<U05AE>;<U05C4>;/
1722     <U05AF>;<U093C>;<U0953>;<U0954>;<U09BC>;<U09D7>;<U0A3C>;/
1723     <U0A70>;<U0A71>;<U0ABC>;<U0B3C>;<U0B56>;<U0B57>;<U0BD7>;<U0C55>;<U0C56>;/
1724     <U0CD5>;<U0CD6>;<U0D57>;<U0F39>;<U302A>..<U302F>;<U3099>;<U309A>
1725
1726     %
1727     END LC_CTYPE

```

4.4 LC_COLLATE

A collation sequence definition defines the relative order between collating elements (characters and multicharacter collating elements) in the FDCC-set. This order is expressed in terms of collation values; i.e., by assigning each element one or more collation values (also known as collation weights). This does not imply that applications assign such values, but that ordering of strings using the resultant collation definition in the FDCC-set behaves as if such assignment is done and used in the collation process. The collation sequence definition is used by regular expressions, pattern matching. When no weights are specified the collation sequence definition also is used for sorting, else the weighting defines the sorting. The following capabilities are provided:

- (1) Multicharacter collating elements. Specification of multicharacter collating elements (i.e., sequences of two or more characters to be collated as an entity).
- (2) User-defined ordering of collating elements. Each collating element is assigned a collation value defining its order in the character (or basic) collation sequence. This ordering is used by regular expressions and pattern matching and, unless collation weights are explicitly specified, also as the collation weight to be used in sorting.
- (3) Multiple weights and equivalence classes. Collating elements can be assigned one or more (up to the limit (COLL_WEIGHTS_MAX)) collating weights for use in sorting. The first weight is hereafter referred to as the primary weight.
- (4) One-to Many mapping. A single character is mapped into a string of collating elements.
- (5) Many-to-Many substitution. A string of one or more characters is substituted by another string (or an empty string, i.e., the character or characters are ignored for collation purposes).
- (6) Equivalence class definition. Two or more collating elements have the same collation value (primary weight).
- (7) Ordering by weights. When two strings are compared to determine their relative order, the two strings are first broken up into a series of collating elements, and each successive pair of elements are compared according to the relative primary weights for the elements. If equal, and more than one weight has been assigned, then the pairs of collating elements are re-compared according to the relative subsequent weights, until either a pair of collating elements compare unequal or the weights are exhausted.
- (8) Easy reordering of characters. ISO/IEC 14651 has a template for collation specification that with just a few modifications can be culturally correct for a

1766		specific culture. Here the "reorder-after" keyword gives a convenient way to
1767		modify a FDCC-set template.
1768	(9)	Easy reordering of sections. The template in ISO/IEC 14651 gives an ordering of
1769		the sections that may not be culturally acceptable in certain cultures. The keyword
1770		"reorder-section-after" gives a convenient way to modify the order of sections in a
1771		FDCC-set template.
1772		
1773		The following keywords are recognized in a collation sequence definition. Some of them
1774		are described in detail in the following subclauses. The keywords are mandatory unless
1775		otherwise noted.
1776		
1777	copy	Specify the name of an existing FDCC-set to be used
1778		as the source for the definition of this category. If
1779		this keyword is specified, only the "reorder-after",
1780		"reorder-end", "reorder-section-after" and "reorder-
1781		section-end" keywords may also be specified. The
1782		FDCC-set is copied in source form.
1783	coll_weight_max	Define as a decimal number the number of collation
1784		levels that an interpreting system needs to support
1785		for this FDCC-set, this value is elsewhere referred to
1786		as the COLL_WEIGHT_MAX limit (e.g. in the
1787		"order_start" statement). An interpreting system
1788		caters for up to 7 collating levels.
1789	section-symbol	Define a section symbol representing a set of
1790		collation order statements. The section is defined
1791		with the "order_start" keyword until the next
1792		"order_start" or "order_end" keyword. This keyword
1793		is optional.
1794	collating-element	Define a collating-element symbol representing a
1795		multicharacter collating element. This keyword is
1796		optional.
1797	collating-symbol	Define one or more collating symbols for use in
1798		collation order statements. This keyword is optional.
1799	symbol-equivalence	Define a collating-symbol to be equivalent to another
1800		defined collating-symbol.
1801	order_start	Define collation rules. This statement is followed by
1802		one or more collation order statements, assigning
1803		character collation values and collation weights to
1804		collating elements.
1805	order_end	Specify the end of the collation-order statements.
1806	reorder-after	Redefine collating rules. Specify after which
1807		collating element the redefinition of collation order
1808		takes order. This statement is followed by one or
1809		more collation order statements, reassigning character
1810		collation values and collation weights to collating
1811		elements.
1812	reorder-end	Specify the end of the "reorder-after" collating order
1813		statements.
1814	reorder-section-after	Redefine the order of sections. This statement is
1815		followed by one or more section symbols,
1816		reassigning character collation values and collation
1817		weights to collating elements.

1818 **reorder-section-end** Specify the end of the "reorder-section" section order
 1819 statements.
 1820

1821 **4.4.1 Collation statements**

1822
 1823 The "order_start" and "reorder-after" keywords are followed by collating statements. The
 1824 syntax for the collating statements is
 1825

```
1826 "%s %s;%s;...;%s\n",<collating-identifier>,<weight>,<weight>,...
```

1827
 1828 Each <collating-identifier> consists of either a character (in any of the forms defined in
 1829 4.1.1), a <collating-element>, a <collating-symbol>, an ellipsis, or the special symbol
 1830 "UNDEFINED". The weights for each of the collation elements determines the character
 1831 collation sequence - such that each collation statement does not need to be in collation
 1832 order, and weights could be rearranged via for example the "reorder-after" keyword. No
 1833 character has any specific predetermined placement in the collation sequence. The order in
 1834 which collating elements are specified determines the character collation sequence, such
 1835 that each collating element compares less than the elements following it.
 1836

1837 A <collating-element> is used to specify multicharacter collating elements, and indicates
 1838 that the character sequence specified via the <collating-element> is to be collated as a unit
 1839 and in the relative order specified by its place in the list of collating statements.
 1840

1841 A <collating-symbol> is used to define a position in the relative order for use in weights.
 1842

1843 The absolute ellipsis symbol ("...") specifies that a sequence of characters collate according
 1844 to their encoded character values. It is interpreted as indicating that all characters with a
 1845 coded character set value higher than the value of the character in the preceding line, and
 1846 lower than the coded character set value for the character in the following line, in the
 1847 current coded character set, are placed in the character collation order between the
 1848 previous and the following character in ascending order according to their coded character
 1849 set values. An initial ellipsis is interpreted as if the preceding line specified the <NUL>
 1850 character, and a trailing ellipsis as if the following line specified the highest coded
 1851 character set value in the current coded character set. An ellipsis is treated as invalid if the
 1852 preceding or following lines do not specify characters in the current coded character set.
 1853 The use of the ellipsis symbol ties the definition to a specific coded character set and may
 1854 preclude the definition from being portable between applications, and is depreciated.
 1855 Symbolic ellipses may be used as the ellipses symbol, but generating symbolic character
 1856 names, and thus have a better chance of portability between applications.
 1857

1858 The symbolic ellipses (".." or "....") specifies a sequence of collating statements. It is
 1859 interpreted as indicating that all characters with symbolic names higher than the symbolic
 1860 name of the character in the preceding line, and lower in the sequence of symbolic names
 1861 for the character in the following line, is placed in the character collation order between
 1862 the previous and the following character in ascending order.
 1863

1864 The symbol "UNDEFINED" is interpreted as including all coded character set values not
 1865 specified explicitly or via the ellipsis or one of the symbolic ellipses symbols. Such
 1866 characters are inserted in the character collation order at the point indicated by the symbol,
 1867 and in ascending order according to their coded character set values. If no "UNDEFINED"
 1868 symbol is specified, and the current coded character set contains characters not specified
 1869 in this clause, the utility issues a warning message and place such characters at the end of

1870 the character collation order.

1871
 1872 The optional operands for each collation-element are used to define the primary,
 1873 secondary, or subsequent weights for the collating element. The first operand specifies the
 1874 relative primary weight, the second the relative secondary weight, and so on. Two or more
 1875 collation-elements can be assigned the same weight; they belong to the same equivalence
 1876 class if they have the same primary weight. Collation behaves as if, for each weight level,
 1877 "IGNORE"d elements are removed. Then each successive pair of elements is compared
 1878 according to the relative weights for the elements. If the two strings compare equal, the
 1879 process is repeated for the next weight level, up to the limit "COLL_WEIGHTS_MAX" of
 1880 the associated FDCC-set.

1881
 1882 Weights are expressed as characters (in any of the forms specified here), <collating-
 1883 symbol>s, <collating-element>s, an ellipsis, or the special symbol "IGNORE". A single
 1884 character, a <collating-symbol>, or a <collating-element> represent the relative order in
 1885 the character collating sequence of the character or symbol, rather than the character or
 1886 characters themselves.

1887
 1888 One-to-many mapping is indicated by specifying two or more concatenated characters or
 1889 symbolic names. Thus, if the character <ss> is given the string <s><s> as a weight,
 1890 comparisons are performed as if all occurrences of the character <ss> are replaced by
 1891 <s><s>. If it is desirable to define <ss> and <s><s> as an equivalence class, then a
 1892 collating-element must be defined for the string "ss", as in the example below.

1893
 1894 All characters specified via an ellipsis are by default assigned unique weights, equal to the
 1895 relative order of characters. Characters specified via an explicit or implicit "UNDEFINED"
 1896 special symbol are by default assigned the same primary weight (i.e., belong to the same
 1897 equivalence class). An ellipsis symbol as a weight is interpreted to mean that each
 1898 character in the sequence has unique weights, equal to the relative order of their character
 1899 in the character collation sequence. Secondary and subsequent weights have unique values.
 1900 The use of the ellipsis as a weight is treated as an error if the collating element is neither
 1901 an ellipsis nor the special symbol "UNDEFINED".

1902
 1903 The special keyword "IGNORE" as a weight indicates that when strings are compared
 1904 using the weights at the level where "IGNORE" is specified, the collating element is
 1905 ignored; i.e., as if the string did not contain the collating element. In regular expressions
 1906 and pattern matching, all characters that are "IGNORE"d in their primary weight form an
 1907 equivalence class.

1908
 1909 A <comment_character> occurring where the delimiter ";" may occur, terminates the
 1910 collating statement.

1911
 1912 An empty operand is interpreted as the collating-element itself.

1913
 1914 For example, the collation statement

1915 <a> <a>;<a>

1916
 1917
 1918 is equal to

1919 <a>

1920
 1921
 1922 An ellipsis (absolute or symbolic) can be used as an operand if the collating-element was

1923 an ellipsis, and is interpreted as the value of each character defined by the ellipsis.

```

1924
1925     Example:
1926
1927     collating-element <ch> from "<c><h>"
1928     collating-element <Ch> from "<C><h>"
1929     order_start      forward;backward
1930     UNDEFINED       IGNORE;IGNORE
1931     <LOW>
1932     <space>         <LOW>;<space>
1933     ...             <LOW>;
1934     <a>              <a>;<a>
1935     <a'>             <a>;<a'>
1936     <A>              <a>;<A>
1937     <A'>            <a>;<A'>
1938     <ch>             <ch>;<ch>
1939     <Ch>             <ch>;<Ch>
1940     <s>              <s>;<s>
1941     <ss>             "<s><s>" ; "<ss><ss>"
1942     order_end

```

1943 This example is interpreted as follows:

- 1944 (1) The UNDEFINED means that all characters not specified in this definition (explicitly or via the
- 1945 ellipsis) is ignored.
- 1946 (2) <LOW> defines the first collating weight, and thus the lowest weight in this example.
- 1947 (3) All characters between <space> and <a> have the same primary equivalence class <LOW> and
- 1948 individual secondary weights based on their ordinal encoded values. (The use of absolute ellipses is
- 1949 deprecated, but used here to illustrate generic use of ellipses. Symbolic ellipses should be used
- 1950 instead).
- 1951 (4) All characters based on the upper or lowercase character "a" belong to the same primary equivalence
- 1952 class.
- 1953 (5) The multicharacter collating element <c><h> is represented by the collating symbol <ch> and belongs
- 1954 to the same primary equivalence class as the multicharacter collating element <C><h>.
- 1955 (6) The <ss> collating element has two weights on the primary level, and it is in the same primary
- 1956 equivalence class as two consecutive <s>-es; on the secondary level the collating element has two
- 1957 weights of the equivalence class <ss>.
- 1958
- 1959

1960 4.4.2 "copy" keyword

1961 This keyword specifies the name of an existing FDCC-set to be used as the source for the

1962 definition of this category. The syntax is

```
1963     "copy %s\n", <FDCC-set-name>
```

1964 The <FDCC-set-name> consists of one or more characters (in any of the forms defined in

1965 4.1.1). If this keyword is specified, only the "reorder-after", "reorder-end", "reorder-

1966 section-after" and "reorder-section-end" keywords may also be specified. The FDCC-set is

1967 copied in source form.

1972 4.4.3 "coll_weight_max" keyword

1973 This keyword defines as a decimal number the number of collation levels that an

1974 interpreting system needs to support. An interpreting system caters for up to 7 collating

1975 levels. The syntax is

```
1976     "coll_weight_max %d\n", <value>
```

1981 4.4.4 "section-symbol" keyword

1983 This keyword is used to define symbols for use in section related statements; such as the
 1984 "order_start", and "reorder-section-after" keywords and section-reordering statements. The
 1985 syntax is

1986
 1987 "section-symbol %s\n", <section-symbol>
 1988

1989 The <section-symbol> is a symbolic name, enclosed between angle brackets (< and >),
 1990 and does not duplicate any symbolic name in the current charmap (if any), or any other
 1991 symbolic name defined in this collation definition. A <section-symbol> defined via this
 1992 keyword is only defined within the LC_COLLATE category.

1993
 1994 Example:
 1995 section-symbol <LATIN>
 1996 section-symbol <ARABIC>

1997 1998 **4.4.5 "collating-element" keyword**

1999 In addition to the collating elements in the character set, the collating-element keyword is
 2000 used to define multicharacter collating elements. The syntax is

2001
 2002 "collating-element %s from %s\n",<collating-symbol>,<string>
 2003
 2004

2005 The <collating-symbol> operand is a symbolic name, enclosed between angle brackets (<
 2006 and >), and does not duplicate any symbolic name in the current charmap or repertoire
 2007 file (if any), or any other symbolic name defined in this collation definition. The string
 2008 operand is a string of two or more characters that collates as an entity. A <collating-
 2009 element> defined via this keyword is only defined within the LC_COLLATE category.

2010
 2011 Example with ISO/IEC 10646-1:
 2012 collating-element <ch> from "<c><h>"
 2013 collating-element <e-acute> from "<e><combining-acute>"
 2014 collating-element <aa> from "<a><a>"
 2015

2016 Note: The problem of comparing a fully composed character of ISO/IEC 10646 with a
 2017 decomposed representation of the same text is sometimes handled by the two strings
 2018 comparing equal up to level 3 (the case level) of ISO/IEC 14651, but distinguishing the
 2019 two at the 4th level.
 2020

2021 **4.4.6 "collating-symbol" keyword**

2022 This keyword is used to define symbols for use in collation sequence statements; e.g.,
 2023 between the order_start and the order_end keywords. The syntax is

2024
 2025 "collating-symbol %s;%s;...%s\n", <collating-symbol>, <collating-symbol> ...
 2026
 2027

2028 The <collating-symbol> is a symbolic name, enclosed between angle brackets (< and >),
 2029 and does not duplicate any symbolic name in the current charmap (if any), or any other
 2030 symbolic name defined in this collation definition. A <collating-symbol> defined via this
 2031 keyword is only defined within the LC_COLLATE category. More than one <collating-
 2032 symbol> may be defined with one "collating-symbol" keyword, and symbolic ellipses may
 2033 be used.

2034
 2035 Example:
 2036 collating-symbol <CAPITAL>

2037 collating-symbol <HIGH>

2038

2039 4.4.7 "symbol-equivalence" keyword

2040

2041 This keyword is used to define symbols for use in collation sequence statements; and
2042 assign the same weight as another defined symbol. The syntax is

2043

2044 "symbol-equivalence %s %s\n", <collating-symbol-1>, <collating-symbol-2>

2045

2046 The <collating-symbol-1> and <collating-symbol-2> are symbolic names, enclosed
2047 between angle brackets (< and >). <collating-symbol-1> does not duplicate any symbolic
2048 name in the current charmap (if any), or any other symbolic name defined in this collation
2049 definition. <collating-symbol-2> is defined elsewhere in the LC_COLLATE category as a
2050 collating-symbol. The use of <collating-symbol-2> is equivalent to using the <collating-
2051 symbol-1> in the LC_COLLATE category. A <collating-symbol-1> defined via this
2052 keyword is only defined within the LC_COLLATE category.

2053

2054 Example

2055 collating-symbol <CAP>

2056 symbol-equivalence <CAPITAL> <CAP>

2057

2058 4.4.8 "order_start" keyword

2059

2060 The "order_start" keyword precedes collation order entries and also defines the number of
2061 weights for this collation sequence definition, the collation section name and other
2062 collation rules.

2063

2064 The syntax of the "order_start" keyword has two forms:

2065

2066 "order_start %s;%s;...;%s\n", <sort-rule>, <sort-rule> ...

2067 and

2068 "order_start %s;%s;...;%s\n", <section-symbol>, <sort-rules>, <sort-rules> ...

2069

2070 The operands to the order_start keyword are optional. If present, the operands define rules
2071 to be applied when strings are compared. The first operand may be a <section-symbol>
2072 surrounded by "<" and ">" and the set of collating statements following the "order_start"
2073 keyword until the "order_end" keyword are identified with this <section-symbol> or
2074 another "order_start" keyword is encountered. The remaining number of operands define
2075 how many weights each element is assigned; if no operands are present, one forward
2076 operand is assumed. If present, the first operand defines rules to be applied when
2077 comparing strings using the first (primary) weight; the second when comparing strings
2078 using the second weight, and so on. Operands are separated by semicolons (;). Each
2079 operand consists of one or more collation directives, separated by commas (.). If the
2080 number of operands exceeds the (COLL_WEIGHTS_MAX) limit, a utility parsing the
2081 FDCC-set description issues a warning message. The following directives are supported:

2082

2083 **forward** Specifies that the direction of scanning a part of a string at a given point in a
2084 string is done towards the logical end of the whole string for this weight level.

2085 **backward** Specifies that the direction of scanning a part of a string at a given point in a
2086 string is done towards the logical beginning of the whole string for this weight
2087 level.

2088 **position** Specifies that comparison operations for the weight level will consider the
2089 relative position of non-"IGNORE"d elements in the strings. The string

2090 containing a non-"IGNORE"d element after the fewest IGNOREd collating
 2091 elements from the start of the compare collates first. If both strings contain a
 2092 non-"IGNORE"d character in the same relative position, the collating values
 2093 assigned to the elements determine the ordering. In case of equality,
 2094 subsequent non-IGNOREd characters are considered in the same manner.
 2095

2096 The directives "forward" and "backward" are mutually exclusive at a given level. The
 2097 directives "backward" and "position" are mutually exclusive at a given level.

2098 Examples:

```
2099 order_start forward;backward
2100 order_start <CYRILLIC>;forward;forward
```

2101
 2102
 2103 If no operands are specified, a single forward operand is assumed.
 2104

2105 4.4.9 "order_end" keyword

2106
 2107
 2108 The collating order entries are terminated with an "order_end" keyword.
 2109

2110 4.4.10 "reorder-after" keyword

2111
 2112 The "reorder-after" keyword is used to specify a modification to a copied collation
 2113 specification of an existing FDCC-set. There can be more than one "reorder-after"
 2114 statement in a collating specification. The syntax is:

```
2115 "reorder-after %s\n",<collating-symbol>
```

2116
 2117
 2118 The <collating-symbol> operand is a symbolic name, enclosed between angle brackets,
 2119 and is present in the source FDCC-set copied via the "copy" keyword.

2120 The "reorder-after" statement is followed by one or more collation statements as described
 2121 in the "Collating Order" clause (4.4.5), with the exception that the ellipsis symbol (...) is
 2122 not used.

2123
 2124 Each collation statement reassigns character collation values and collation weights to
 2125 collating elements existing in the copied collation specification, by removing the collating
 2126 statement from the copied specification, and inserting the collating element in the collating
 2127 sequence with the new collation weights after the preceding collating element of the
 2128 "reorder-after" specification, the first collating element in the collation sequence being the
 2129 <collating-symbol> specified in the "reorder-after" statement.
 2130

2131 A "reorder-after" specification is terminated by another "reorder-after" specification or the
 2132 "reorder-end" statement.
 2133

2134 4.4.10.1 Example of "reorder-after"

```
2135 reorder-after <y8>
2136 <U:> <Y>;<U:>;<CAPITAL>
2137 <u:> <Y>;<U:>;<SMALL>
2138 reorder-after <z8>
2139 <AE> <AE>;<NONE>;<CAPITAL>
2140 <ae> <AE>;<NONE>;<SMALL>
2141 <A:> <AE>;<DIAERESIS>;<CAPITAL>
2142 <a:> <AE>;<DIAERESIS>;<SMALL>
2143 <O/> <O/>;<NONE>;<CAPITAL>
2144 <o/> <O/>;<NONE>;<SMALL>
```

2146 <AA> <AA> ; <NONE> ; <CAPITAL>
 2147 <aa> <AA> ; <NONE> ; <SMALL>
 2148 reorder-end

2149
 2150 The example is interpreted as follows (using the "i18nrep" repertoire map):

- 2151
 2152 1. The collating element <U:> is removed from the copied collating sequence and inserted after <y8> in the
 2153 collating sequence with the new weights. The collating element <u:> is removed from the copied collating
 2154 sequence and inserted in the resulting collation sequence after <U:> with the new weights. <y8> is used to
 2155 indicate the last entry of the <y> letters.
 2156
 2157 2. The second "reorder-after" statement terminates the first list of reordering collation identifier entries, and
 2158 initiates a second list, rearranging the order and weights for the <AE>, <ae>, <A:>, <a:>, <O/>, and <o/>
 2159 collating elements after the <z8> collating symbol in the copied specification. <z8> is used to indicate the
 2160 last entry of the <z> letters.
 2161
 2162 3. The "reorder-end" statement terminates the second list of reordering entries.
 2163
 2164 4. Thus for the original sequence

2165 ... (U u Ü ü) V v W w X x Y y Z z

2166 this example reordering gives

2167 ... U u V v W w X x (Y y Ü ü) Z z (Æ æ Ä ä) Ø ø Å å

2168
 2169
 2170 where the parenthesis indicate ordering with the same weight on the first level for multiple upper/lowercase
 2171 pairs.
 2172
 2173
 2174

2175 4.4.11 "reorder-end" keyword

2176
 2177 The "reorder-end" keyword specifies the end of a list of collating statements, initiated by
 2178 the "reorder-after" keyword.
 2179

2180 4.4.12 "reorder-section-after" keyword

2181
 2182 The "reorder-section-after" keyword is used to specify a modification to a copied collation
 2183 specification of an existing FDCC-set. The "reorder-section-after" statement is followed by
 2184 one or more statements consisting of section reordering statements.
 2185

2186 4.4.12.1 section reordering statements

2187
 2188 The section reordering statements rearranges the set of collating entries and changes
 2189 sorting rules for the set of collating entries identified by a section symbol in a preceding
 2190 "order_start" statement. Each section reorder statement has the syntax:
 2191

2192 "%s %s;...%s\n", <section-symbol>, <sort-rule>, <sort-rule> ...

2193
 2194 The <section-symbol> identifies the set of collating entries, and is defined via a "section-
 2195 symbol" keyword.
 2196

2197 The <sort-rule>s are as described for the "order_start" keyword. Specified <sort-rule>s
 2198 replace the specification for the ordering of the section given on the "order_start"
 2199 statement identified by the <section-symbol>. The <sort-rule>s are optional, and <sort-
 2200 rule>s not to be changed may be given by empty specifications.
 2201

2202 Note: The <sort-rule> capability is an extension over ISO/IEC 14651 functionality.

2203 The order of the section reordering statements rearranges the assignment of collation
 2204 entries for the sets of collation entries identified by the <section-symbols> to the order
 2205 that the <section-symbols> occur after the "reorder-section-after" statement.
 2206

2207 The section reordering statements are terminated by a "reorder-section-end" statement.
 2208

2209 4.4.12.2 Example of section reordering

```
2210 copy "i18n"
2211 reorder-section-after <DIGITS>
2212 <ARABIC>
2213 <LATIN> forward;backward;forward;forward,position
2214 reorder-section-end
2215
```

2216 This example is interpreted as follows: The LC_COLLATE category of the "i18n" FDCC-set is copied. Then a
 2217 reordering of all collating statements for the sections <ARABIC> and <LATIN> is done, leaving the rest of the
 2218 sections as they were in the "i18n" FDCC-set. The <ARABIC> section is placed immediately after the <DIGITS>
 2219 section, and the <LATIN> section immediately following the <ARABIC> section. The ordering rules are kept as
 2220 they were in the "i18n" FDCC-set, while the <LATIN> section gets new ordering rules as indicated. The
 2221 "reorder-section-end" keyword terminates the section reordering statements.
 2222

2223 4.4.13 "reorder-section-end" keyword

2224 The "reorder-section-end" keyword specifies the end of a list of section symbols, initiated
 2225 by the "reorder-section-after" keyword.
 2226

2227 4.4.14 "i18n" LC_COLLATE category

2228 The "i18n" LC_COLLATE category is defined as the following, which includes the
 2229 tailorable template in ISO/IEC 14651.
 2230

```
2231 LC_COLLATE
2232 % This is the ISO/IEC TR 14652 i18n fdcc-set definition for
2233 % the LC_COLLATE category.
2234 %
2235 % equivalences
2236 symbol-equivalence <NONE> <BLANK>
2237 symbol-equivalence <CAPITAL> <CAP>
2238 symbol-equivalence <SMALL> <MIN>
2239 symbol-equivalence <CAPITAL-SMALL> <COMPATCAP>
2240 symbol-equivalence <SMALL-CAPITAL> <COMPAT>
2241 symbol-equivalence <MACRON> <MACRO>
2242 symbol-equivalence <STROKE> <OBLIK>
2243 symbol-equivalence <ACUTE> <AIGUT>
2244 symbol-equivalence <CIRCUMFLEX> <CIRCF>
2245 symbol-equivalence <RING> <CRCLE>
2246 symbol-equivalence <DIAERESIS> <TREMA>
2247 symbol-equivalence <DOT> <POINT>
2248 symbol-equivalence <CEDILLA> <CEDIL>
2249 symbol-equivalence <OGONEK> <OGONK>
2250 symbol-equivalence <HOOK> <CROOK>
2251 symbol-equivalence <HORN> <HORNU>
2252 symbol-equivalence <DOT-BELOW> <POINS>
2253
2254 order_start forward;forward;forward;forward,position
2255
2256 % Copy the template from ISO/IEC 14651
2257 copy "ISO14651_2000_TABLE1.txt"
2258
2259 order_end
2260
2261 END LC_COLLATE
2262
```


2268
 2269
 2270
 2271
 2272
 2273
 2274
 2275
 2276
 2277
 2278
 2279
 2280
 2281
 2282
 2283
 2284
 2285
 2286
 2287
 2288
 2289
 2290
 2291
 2292
 2293
 2294
 2295
 2296
 2297
 2298
 2299
 2300
 2301
 2302
 2303
 2304
 2305
 2306
 2307
 2308
 2309
 2310
 2311
 2312
 2313
 2314
 2315
 2316
 2317
 2318
 2319

4.5 LC_MONETARY

The LC_MONETARY category defines the rules and symbols that are used to format monetary numeric information. The operands are strings. For some keywords, the strings can contain only integers. More than one set of monetary values may be provided, and for each set a period of validity and conversion rate may be given. Keywords that are not provided, string values set to the empty string "", or integer keywords set to -1, are used to indicate that the value is unspecified, and then no default is implied. The following keywords are defined:

copy	Specify the name of an existing FDCC-set to be used as the source for the definition of this category. If this keyword is specified, no other keyword is specified.
valid_from	One or more strings separated by semicolons, representing a Gregorian date in the form "YYYYMMDD" according to ISO 8601, specifying the beginning date (inclusive from the beginning of day local time) of the validity of a currency. The position of the string in the list corresponds to the position of operands in other keywords in the LC_MONETARY category. The currencies should be ordered in terms of validity dates, and for each validity period with the currency that the amounts are stored in first. If not specified, it is taken to be an implementation-defined beginning of time. This keyword is optional.
valid_to	One or more strings separated by semicolons, representing a Gregorian date in the form "YYYYMMDD" according to ISO 8601, specifying the end date (inclusive to the end of day local time) of the validity of a currency. If not specified, it is taken to be an implementation-defined end of time. This keyword is optional.
conversion_rate	one or more pairs of integers separated by a <semicolon> specifying the fixed conversion rate between the current currency (determined by the parameter number) and the first currency that is valid, determined by a date provided by the application. If the currency is not the first valid currency for the period in question, the first integer is for multiplying the first valid currency, and the second for dividing this result to get the amount in the current currency. The currency to be the current currency is selected by the application from the date applicable and the currency number (first, second, third etc valid currency at that date); and whether domestic or international formatting is used is also determined by the application. Each pair of integers are separated by a <slash>. The default value is "1/100". This keyword is optional. Note: The two integers are used instead of a floating point value, to be able to cater for legal requirements on Euro conversion where a multiplication and division is prescribed, instead of just one floating point multiplication.
currency_symbol	One or more strings separated by semicolons that are used as the local currency symbol.
mon_decimal_point	The operand is a string containing the symbol that is used as

2320		the decimal delimiter in monetary formatted quantities. In
2321		contexts where other standards limit the "mon_decimal_point"
2322		to a single byte, the result of specifying a
2323		multibyte operand is unspecified. The keyword is specified,
2324		unless the "copy" keyword is used.
2325	mon_thousands_sep	The operand is a string containing the symbol that is used as
2326		a separator for groups of digits to the left of the decimal
2327		delimiter in formatted monetary quantities. In contexts where
2328		other standards limit the "mon_thousands_sep" to a single
2329		byte, the result of specifying a multibyte operand is
2330		unspecified. The keyword is specified, unless the "copy"
2331		keyword is used.
2332	mon_grouping	Define the size of each group of digits in formatted
2333		monetary quantities. The operand is a sequence of integers
2334		separated by semicolons. Each integer specifies the number
2335		of digits in each group, with the initial integer defining the
2336		size of the group immediately preceding the decimal
2337		delimiter, and the following integers defining the preceding
2338		groups. If the last integer is not -1, then the size of the
2339		previous group (if any) is repeatedly used for the remainder
2340		of the digits. If the last integer is -1, then no further
2341		grouping is performed. The keyword is specified, unless the
2342		"copy" keyword is used.
2343	positive_sign	A string that is used to indicate a nonnegative-valued
2344		formatted monetary quantity. The keyword is specified,
2345		unless the "copy" keyword is used.
2346	negative_sign	A string that is used to indicate a negative-valued formatted
2347		monetary quantity. The keyword is specified, unless the
2348		"copy" keyword is used.
2349	frac_digits	One or more integers separated by semicolons, representing
2350		the number of fractional digits (those to the right of the
2351		decimal delimiter) to be written in a formatted monetary
2352		quantity using "currency_symbol". The keyword is specified,
2353		unless the "copy" keyword is used.
2354	p_cs_precedes	One or more integers separated by semicolons, set to 1 if the
2355		"currency_symbol" precedes the value for a nonnegative
2356		formatted monetary quantity, and set to 0 if the symbol
2357		succeeds the value. The keyword is specified, unless the
2358		"copy" keyword is used.
2359	p_sep_by_space	One or more integers separated by semicolons, set to 0 if no
2360		space separates the "currency_symbol" from the value for a
2361		nonnegative formatted monetary quantity, set to 1 if a space
2362		separates the symbol from the value, and set to 2 if a space
2363		separates the symbol and the sign string, if adjacent. The
2364		keyword is specified, unless the "copy" keyword is used.
2365	n_cs_precedes	One or more integers separated by semicolons, set to 1 if the
2366		"currency_symbol" precedes the value for a negative
2367		formatted monetary quantity, and set to 0 if the symbol
2368		succeeds the value. The keyword is specified, unless the
2369		"copy" keyword is used.
2370	n_sep_by_space	One or more integers separated by semicolons, set to 0 if no
2371		space separates the "currency_symbol" from the value for a

2372		negative formatted monetary quantity, set to 1 if a space
2373		separates the symbol from the value, and set to 2 if a space
2374		separates the symbol and the sign string, if adjacent. The
2375		keyword is specified, unless the "copy" keyword is used.
2376	p_sign_posn	One or more integers separated by semicolons, set to a value
2377		indicating the positioning of the "positive_sign" for a
2378		nonnegative formatted monetary quantity using the
2379		"currency_symbol". The following integer values are defined:
2380		
2381		0 Parentheses enclose the quantity and the
2382		"currency_symbol".
2383		1 The sign string precedes the quantity and the
2384		"currency_symbol".
2385		2 The sign string succeeds the quantity and the
2386		"currency_symbol".
2387		3 The sign string immediately precedes the
2388		"currency_symbol".
2389		4 The sign string immediately succeeds the
2390		"currency_symbol".
2391		The keyword is specified, unless the "copy" keyword is used.
2392		
2393	n_sign_posn	One or more integers separated by semicolons, set to a value
2394		indicating the positioning of the "negative_sign" for a
2395		negative formatted monetary quantity using the
2396		"currency_symbol". The following integer values are defined:
2397		
2398		0 Parentheses enclose the quantity and the
2399		"currency_symbol".
2400		1 The sign string precedes the quantity and the
2401		"currency_symbol".
2402		2 The sign string succeeds the quantity and the
2403		"currency_symbol".
2404		3 The sign string immediately precedes the
2405		"currency_symbol".
2406		4 The sign string immediately succeeds the
2407		"currency_symbol".
2408		The keyword is specified, unless the "copy" keyword is used.
2409		
2410	int_curr_symbol	One or more strings separated by semicolons that are used as
2411		the international currency symbols. Each operand is a four
2412		character string, with the first three characters containing the
2413		alphabetic international currency symbol in accordance with
2414		those specified in ISO 4217, <i>Codes for the representation of</i>
2415		<i>currencies and funds</i> . The fourth character is the character
2416		used to separate the international currency symbol from the
2417		monetary quantity. The keyword is specified, unless the
2418		"copy" keyword is used.
2419	int_frac_digits	One or more integers separated by semicolons, representing
2420		the number of fractional digits (those to the right of the
2421		decimal delimiter) to be written in a formatted monetary
2422		quantity using "int_curr_symbol". The keyword is specified,
2423		unless the "copy" keyword is used.

2424	int_p_cs_precedes	One or more integers separated by semicolons; set to 1 if the
2425		"int_curr_symbol" precedes the value for a nonnegative
2426		formatted monetary quantity, and set to 0 if the symbol
2427		succeeds the value. If not specified, the value of
2428		"p_cs_precedes" is taken.
2429	int_p_sep_by_space	One or more integers separated by semicolons; set to 0 if no
2430		space separates the "int_curr_symbol" from the value for a
2431		nonnegative formatted monetary quantity, set to 1 if a space
2432		separates the symbol from the value, and set to 2 if a space
2433		separates the symbol and the sign string, if adjacent. If not
2434		specified, the value of "p_sep_by_space" is taken.
2435	int_n_cs_precedes	One or more integers separated by semicolons; set to 1 if the
2436		"int_curr_symbol" precedes the value for a negative
2437		formatted monetary quantity, and set to 0 if the symbol
2438		succeeds the value. If not specified, the value of
2439		"n_cs_precedes" is taken.
2440	int_n_sep_by_space	One or more integers separated by semicolons; set to 0 if no
2441		space separates the "int_curr_symbol" from the value for a
2442		negative formatted monetary quantity, set to 1 if a space
2443		separates the symbol from the value, and set to 2 if a space
2444		separates the symbol and the sign string, if adjacent. If not
2445		specified, the value of "n_sep_by_space" is taken.
2446	int_p_sign_posn	One or more integers separated by semicolons, set to a value
2447		indicating the positioning of the "positive_sign" for a
2448		nonnegative formatted monetary quantity using the
2449		"int_curr_symbol". The following integer values are defined:
2450		
2451		0 Parentheses enclose the quantity and the
2452		"int_curr_symbol".
2453		1 The sign string precedes the quantity and the
2454		"int_curr_symbol".
2455		2 The sign string succeeds the quantity and the
2456		"int_curr_symbol".
2457		3 The sign string immediately precedes the
2458		"int_curr_symbol".
2459		4 The sign string immediately succeeds the
2460		"int_curr_symbol".
2461		If no "int_p_sign_posn" is present the value of the
2462		"p_sign_posn" is taken.
2463		
2464	int_n_sign_posn	One or more integers separated by semicolons, set to a value
2465		indicating the positioning of the "negative_sign" for a
2466		negative formatted monetary quantity using the
2467		"int_curr_symbol". The following integer values are defined:
2468		
2469		0 Parentheses enclose the quantity and the
2470		"int_curr_symbol".
2471		1 The sign string precedes the quantity and the
2472		"int_curr_symbol".
2473		2 The sign string succeeds the quantity and the
2474		"int_curr_symbol".
2475		3 The sign string immediately precedes the

2476 "int_curr_symbol".
 2477 4 The sign string immediately succeeds the
 2478 "int_curr_symbol".
 2479 If no "int_n_sign_posn" is present the value of the
 2480 "n_sign_posn" is taken.
 2481

2482 The "i18n" FDCC-set is defined as follows for the LC_MONETARY category.
 2483

```

2484 LC_MONETARY
2485 % This is the 14652 i18n fdcc-set definition for
2486 % the LC_MONETARY category.
2487 %
2488 int_curr_symbol      ""
2489 currency_symbol     ""
2490 mon_decimal_point    "<U002C>"
2491 mon_thousands_sep   ""
2492 mon_grouping        -1
2493 positive_sign       ""
2494 negative_sign       "<U002E>"
2495 int_frac_digits     -1
2496 frac_digits         -1
2497 p_cs_precedes       -1
2498 p_sep_by_space      -1
2499 n_cs_precedes       -1
2500 n_sep_by_space      -1
2501 p_sign_posn        -1
2502 n_sign_posn        -1
2503 %
2504 END LC_MONETARY
  
```

2507 4.6 LC_NUMERIC

2508
 2509 The LC_NUMERIC category defines the rules and symbols that are used to format
 2510 nonmonetary numeric information. The operands are strings. For some keywords, the
 2511 strings only can contain integers. Keywords that are not provided, string values set to the
 2512 empty string (""), or integer keywords set to -1, are used to indicate that the value is
 2513 unspecified. The following keywords are defined:
 2514

- 2515 **copy** Specify the name of an existing FDCC-set to be used as the
 2516 source for the definition of this category. If this keyword is
 2517 specified, no other keyword is specified.
- 2518 **decimal_point** The operand is a string containing the symbol that is used as the
 2519 decimal delimiter in numeric, nonmonetary formatted quantities.
 2520 This keyword cannot be omitted and cannot be set to the empty
 2521 string. In contexts where other standards limit the decimal point
 2522 to a single byte, the result of specifying a multibyte operand is
 2523 unspecified.
- 2524 **thousands_sep** The operand is a string containing the symbol that is used as a
 2525 separator for groups of digits to the left of the decimal delimiter
 2526 in numeric, nonmonetary formatted monetary quantities. In
 2527 contexts where other standards limit the "thousands_sep" to a
 2528 single byte, the result of specifying a multibyte operand is
 2529 unspecified.
- 2530 **grouping** Define the size of each group of digits in formatted non-
 2531 monetary quantities. The operand is a sequence of integers
 2532 separated by semicolons. Each integer specifies the number of
 2533 digits in each group, with the initial integer defining the size of

2534 the group immediately preceding the decimal delimiter, and the
 2535 following integers defining the preceding groups. If the last
 2536 integer is not -1, then the size of the previous group (if any) is
 2537 repeatedly used for the remainder of the digits. If the last integer
 2538 is -1, then no further grouping is performed.
 2539

2540 The "i18n" FDCC-set is for the LC_NUMERIC category:

```

2541
2542 LC_NUMERIC
2543 % This is the 14652 i18n fdcc-set definition for
2544 % the LC_NUMERIC category.
2545 %
2546 decimal_point    "<U002C>"
2547 thousands_sep    " "
2548 grouping         -1
2549 %
2550 END LC_NUMERIC
2551
  
```

2552 4.7 LC_TIME

2553 The LC_TIME category defines the rules and symbols that are used to format date and
 2554 time information. The following keywords are defined:

2557	copy	Specify the name of an existing FDCC-set to be used as the source for the definition of this category. If this keyword is specified, no other keyword is specified.
2561	abday	Define the abbreviated weekday names for calendar systems with weeks of constant length, to be referenced by the %a field descriptor. The length of the week and a gregorian date for the first weekday is defined by the "week" keyword. The operand consists of semicolon-separated strings. The first string is the abbreviated name of the day corresponding to the first day of the week (default Sunday), the second the abbreviated name of the day corresponding to the second day of the week (default Monday), and so on.
2569	day	Define the full weekday names for calendar systems with weeks of constant length, to be referenced by the %A field descriptor. The length of the week and a gregorian date for the first weekday is defined by the "week" keyword. The operand consists of semicolon-separated strings. The first string is the full name of the day corresponding to the first day of the week (default Sunday), the second the full name of the day corresponding to the second day of the week (default Monday), and so on.
2577	week	Is used to define the number of days in a week, and which weekday is the first weekday (the first weekday has the value 1), and which week is to be considered the first in a year. The first operand is an integer specifying the number of days in the week. The second operand is an integer specifying the Gregorian date in the format YYYYMMDD, and it specifies a day that is a first weekday (all other first weekdays may then be calculated by adding or subtracting a whole multipulum of the number of days in the week as specified with the first operand). The third operand is an integer specifying the weekday number to be contained in the first week of the year. The third operand may also be understood as the number of days required in a week for it to be considered the first week of the year. If the

2589		keyword is not specified the values are taken as 7, 19971130 (a
2590		Sunday), and 7 (Saturday), respectively. ISO 8601 conforming
2591		applications should use the values 7, 19971201 (a Monday), and 4
2592		(Thursday), respectively. This keyword is optional.
2593	abmon	Define the abbreviated month names, to be referenced by the %b
2594		field descriptor. The operand consists of twelve or thirteen
2595		semicolon-separated strings. The first string is the abbreviated name
2596		of the first month of the year (January), the second the abbreviated
2597		name of the second month, and so on.
2598	mon	Define the full month names, to be referenced by the %B field
2599		descriptor. The operand consists of twelve or thirteen semicolon-
2600		separated strings. The first string is the full name of the first month
2601		of the year (January), the second the full name of the second month,
2602		and so on.
2603	d_t_fmt	Define the appropriate date and time representation, to be referenced
2604		by the %c field descriptor. The operand consists of a string, and can
2605		contain any combination of characters and field descriptors. In ad-
2606		dition, the string can contain escape sequences defined in Table 3.
2607	d_fmt	Define the appropriate date representation, to be referenced by the
2608		%x field descriptor. The operand consists of a string, and can contain
2609		any combination of characters and field descriptors. In addition, the
2610		string can contain escape sequences defined in Table 3.
2611	t_fmt	Define the appropriate time representation, to be referenced by the
2612		%X field descriptor. The operand consists of a string, and can
2613		contain any combination of characters and field descriptors. In
2614		addition, the string can contain escape sequences defined in Table 3.
2615	am_pm	Define the appropriate representation of the ante meridiem and post
2616		meridiem strings, to be referenced by the %p field descriptor. The
2617		operand consists of two strings, separated by a semicolon. The first
2618		string represents the antemeridiem designation, the last string the
2619		postmeridiem designation. The keyword is optional. If unspecified,
2620		the %p field descriptor refers to the empty string.
2621	t_fmt_ampm	Define the appropriate time representation in the 12-hour clock
2622		format with "am_pm", to be referenced by the %r field descriptor.
2623		The operand consists of a string and can contain any combination of
2624		characters and field descriptors. If the string is empty, the 12-hour
2625		format is not supported in the FDCC-set.

The following keywords are all optional

2627		
2628		
2629	era	Is used to define alternate Eras, corresponding to the %E field
2630		descriptor modifier. The format of the operand is unspecified, but
2631		supports the definition of the %EC and %Ey field descriptors, and
2632		may also define the "era_year" format (%EY).
2633	era_year	Is used to define the format of the year in alternate Era format,
2634		corresponding to the %EY field descriptor.
2635	era_d_t_fmt	Is used to define the format of the date and time in alternate Era
2636		notation, corresponding to the %Ec field descriptor.
2637	era_d_fmt	Is used to define the format of the date in alternate Era notation,
2638		corresponding to the %Ex field descriptor.
2639	era_t_fmt	Is used to define the format of the time in alternate Era notation,
2640		corresponding to the %EX field descriptor.

2641	alt_digits	Is used to define alternate symbols for digits, corresponding to the
2642		%O field descriptor modifier. The operand consists of semicolon-
2643		separated strings. The first string is the alternate symbol correspon-
2644		ding with zero, the second string the symbol corresponding with one,
2645		and so on. Up to 100 alternate symbol strings can be specified. The
2646		%O modifier indicates that the string corresponding to the value spe-
2647		cified via the field descriptor is used instead of the value.
2648	first_weekday	Is used to define the first day to be displayed, for example in a
2649		calendar display utility. The operand is an integer specifying the day
2650		number (1 = first) according to the information specified with the
2651		"day" keyword. The keyword may be omitted, and then the value 1 is
2652		taken, corresponding to Sunday for a week beginning Sunday, or to
2653		Monday for a week beginning Monday.
2654	first_workday	Is used to define the first workday as an integer according to the day
2655		numbering specified with the "week" keyword.
2656	cal_direction	Is used to define the direction of the display of dates, for example in
2657		a calendar display utility. The operand is an integer, and the
2658		following values are defined:
2659		1 left-right from top
2660		2 top-down from left
2661		3 right-left from top
2662		The keyword may be omitted, and then the value 1 is taken.
2663	timezone	Is used to define one or more timezones, each defined by a string,
2664		and the strings separated by a <semicolon>. In the following the
2665		characters <, >, [and] are used as metacharacters. Only characters
2666		with a visible glyph from the portable character set may be used,
2667		except in the <std> and <dst> fields. The syntax of a string is:
2668		
2669		<std><offset><dst>[<offset>][,<rule>[,<rule>...]];
2670		
2671		where
2672		
2673		<std> and <dst> Indicates no less than three, nor more than 10
2674		characters that are the designation for the
2675		standard <std>, or Daylight Savings Time or
2676		summer time <dst> zone. Only <std> is
2677		required; if <dst> is missing, then Daylight
2678		Savings Time or summer time does not apply
2679		in this category. Upper- and lowercase letters
2680		are explicitly allowed. Any characters except a
2681		leading colon <:> or digits, the comma <,>, the
2682		minus <->, the plus <+>, and the null character
2683		are permitted to appear in these fields, but their
2684		meaning is unspecified.
2685		<offset> Indicates the value one must add to the local
2686		time to arrive at the Coordinated Universal
2687		Time. The <offset> has the form:
2688		
2689		hh[:mm[:ss]]
2690		
2691		The minutes (mm) and seconds (ss) are
2692		optional. The hour (hh) is required and may be

2693		a single digit. The <offset> following <std> is
2694		required. If no <offset> follows <dst>, summer
2695		time is assumed to be one hour ahead of
2696		standard time. One or more digits may be used;
2697		the value is always interpreted as a decimal
2698		number. The hour is between zero and 24, and
2699		the minutes (and seconds) - if present - is
2700		between zero and 59. If preceded by a "-", the
2701		time zone is east of the Prime Meridian;
2702		otherwise it is west of (which may be indicated
2703		by an optional preceding "+").
2704	<rule>	A specification for Daylight Savings Time
2705		changes that indicates when to change to and
2706		back from summer time. The <rule> has the
2707		form:
2708		<date>[/<time>/<year>],<date>[/<time>
2709		>/<year>]
2710		where the first <date> describes when the
2711		change from standard time to summer time
2712		occurs, and the second <date> describes when
2713		the change back happens. Each <time> field
2714		describes when, in current local time, the
2715		change to the other time is made. The first
2716		<year> field defines the beginning of the
2717		validity of this rule, and the second <year>
2718		field defines the end of the validity of the rule.
2719		A number of rules may be given.
2720		
2721		The format of <date> is one of the following:
2722		
2723	J<n>	The Julian day <n> (1 <= n
2724		<= 365) Leap years are not
2725		counted. That is, in all years -
2726		including leap years -
2727		February 28 is day 59 and
2728		March 1 is day 60. It is
2729		impossible to explicitly refer
2730		to the occasional February 29.
2731	<n>	The zero-based Julian day (0
2732		<= n <= 365). Leap years are
2733		counted and it is possible to
2734		refer to February 29.
2735	M<m>.<n>.<d>	
2736		the <d>th day (0 <= d <= 7)
2737		of week <n> of month <m> (1
2738		<= n <= 5, 1 <= m <= 12,
2739		where week 5 means "the last
2740		<d> day in month <m>"
2741		which may occur in either the
2742		fourth or fifth week). Week 1
2743		is the first week in which the
2744		<d>th day occurs. Day zero

2745 and day seven is Sunday.

2746
2747 The <time> has the same format as <offset>
2748 except that no leading sign ("- or "+) is
2749 allowed. The default, if <time> is not given, is
2750 "02:00:00".

2751
2752 The <year> has the format YYYY.

2753
2754 NOTE: This way of specifying the timezone is compatible with the
2755 format for the environment variable TZ described in Section 8.1.1 of
2756 POSIX.1.

2757 2758 **4.7.1 Date Field Descriptors**

2759
2760 The LC_TIME category defines the interpretation of a number of field descriptors. The
2761 field descriptors are also available in the definitions with the following LC_TIME
2762 keywords: "d_t_fmt", "d_fmt", "t_fmt", "t_fmt_ampm", "era", "era_d_t_fmt", "era_d_fmt",
2763 and "era_t_fmt". A field descriptor may not be used with the LC_TIME keywords defining
2764 it.

2765
2766 **Table 3: Escape sequences for the date field**

2767		
2768	%a	FDCC-set's abbreviated weekday name.
2769	%A	FDCC-set's full weekday name.
2770	%b	FDCC-set's abbreviated month name.
2771	%B	FDCC-set's full month name.
2772	%c	FDCC-set's appropriate date and time representation.
2773	%C	Century (a year divided by 100 and truncated to integer) as decimal
2774		number (00-99).
2775	%d	Day of the month as a decimal number (01-31).
2776	%D	Date in the format mm/dd/yy.
2777	%e	Day of the month as a decimal number (1-31 in at two-digit field with
2778		leading <space> fill).
2779	%F	The date in the format YYYY-MM-DD (ISO 8601 format).
2780	%g	Week-based year within century, as a decimal number (00-99).
2781	%G	Week-based year with century, as a decimal number (for example 1997).
2782	%h	A synonym for %b.
2783	%H	Hour (24-hour clock), as a decimal number (00-23).
2784	%I	Hour (12-hour clock), as a decimal number (01-12).
2785	%j	Day of the year, as a decimal number (001-366).
2786	%m	Month, as a decimal number (01-13).
2787	%M	Minute, as a decimal number (00-59).
2788	%n	A <newline> character.
2789	%p	FDCC-set's equivalent of either AM or PM.
2790	%r	12-hour clock time (01-12), using the AM/PM notation.
2791	%R	24-hour clock time, in the format "%H:%M".
2792	%S	Seconds, as a decimal number (00-61).
2793	%t	A <tab> character.
2794	%T	24-hour clock time, in the format HH:MM:SS.
2795	%u	Weekday, as a decimal number (1(Monday)-7).
2796	%U	Week number of the year (Sunday as the first day of the week) as a

2797		decimal number (00-53). All days in a new year preceding the first
2798		Sunday are considered to be in week 0.
2799	%v	Week number of the year, as a decimal number with two digits including
2800		a possible leading zero, according to "week" keyword.
2801	%V	Week of the year (Monday as the first day of the week), as a decimal
2802		number (01-53). The method for determining the week number is as
2803		specified by ISO 8601.
2804	%w	Weekday, as a decimal number (0(Sunday)-6).
2805	%W	Week number of the year (Monday as the first day of the week), as a
2806		decimal number (00-53). All days in a new year preceding the first
2807		Monday are considered to be in week 0.
2808	%x	FDCC-set's appropriate date representation.
2809	%X	FDCC-set's appropriate time representation.
2810	%y	Year within century (00-99).
2811	%Y	Year with century, as a decimal number.
2812	%z	The offset from UTC in the ISO 8601 format "-0430" (meaning 4 hours
2813		30 minutes behind UTC, west of Greenwich), or by no characters if no
2814		time zone is determinable.
2815	%Z	Time-zone name, or no characters if no time zone is determinable.
2816	%%	A <percent-sign> character.

2817
2818 NOTE: %g, %G and %V give values according to the ISO 8601 week-based year. In
2819 this system, weeks begin on a Monday and week 1 of the year is the week that includes
2820 4th January, which is also the week that includes the first Thursday of the year, and is
2821 also the first week that contains at least four days in the year. If the first Monday of the
2822 year is the 2nd, 3rd or 4th, the preceding days are part of the last week of the
2823 preceding year; thus, for Saturday 2nd January 1999, %G is replaced by 1998 and %V
2824 is replaced by 53. If the 29th, 30th or 31st January is a Monday, it and any following
2825 days are part of week 1 of the following year. Thus, for Tuesday 30th December 1997,
2826 %G is replaced by 1998 and %V is replaced by 1.

2827 2828 **4.7.2 Modified Field Descriptors**

2829
2830 Some field descriptors can be modified by the E and O modifier characters to indicate a
2831 different format or specification as specified in the LC_TIME FDCC-set description. If the
2832 corresponding keyword (see "era", "era_year", "era_d_t_fmt", "era_d_fmt", "era_t_fmt"
2833 and "alt_digits") is not specified for the current FDCC-set, the unmodified field descriptor
2834 value is used.

2835		
2836	%Ec	FDCC-set's alternate date and time representation.
2837	%EC	The name of the base year (period) in the FDCC-set's alternate represen-
2838		tation.
2839	%Ex	FDCC-set's alternate date representation.
2840	%EX	FDCC-set's alternate time representation.
2841	%Ey	Offset from %EC (year only) in the FDCC-set's alternate representation.
2842	%EY	Full alternate year representation.
2843	%Od	Day of month using the FDCC-set's alternate numeric symbols.
2844	%Oe	Day of month using the FDCC-set's alternate numeric symbols.
2845	%Of	Weekday as a decimal number according to alt_day (1 is first day).
2846	%OH	Hour (24-hour clock) using the FDCC-set's alternate numeric symbols.
2847	%OI	Hour (12-hour clock) using the FDCC-set's alternate numeric symbols.
2848	%Om	Month using the FDCC-set's alternate numeric symbols.

2849	%OM	Minutes using the FDCC-set's alternate numeric symbols.
2850	%OS	Seconds using the FDCC-set's alternate numeric symbols.
2851	%Ou	Weekday as a number in the alternate representation of the FDCC-set
2852		(Monday=1).
2853	%OU	Week number of the year (Sunday as the first day of the week) using the
2854		FDCC-set's alternate numeric symbols.
2855	%OV	Week number of the year (Monday as the first day of the week, ISO 8601 rules)
2856		using the alternate numeric symbols of the FDCC-set.
2857	%Ow	Weekday as number in the FDCC-set's alternate representation
2858		(Sunday=0).
2859	%OW	Week number of the year (Monday as the first day of the week) using the
2860		FDCC-set's alternate numeric symbols.
2861	%Oy	Year (offset from %C) in alternate representation.

4.7.3 "i18n" LC_TIME category

The "i18n" LC_TIME category is (following ISO 8601):

```

2867 LC_TIME
2868 % This is the ISO/IEC TR 14652 "i18n" definition for
2869 % the LC_TIME category.
2870 %
2871 % Weekday and week numbering according to ISO 8601
2872 abday "<U0031>";"<U0032>";"<U0033>";"<U0034>";/
2873 "<U0035>";"<U0036>";"<U0037>"
2874 day "<U0031>";"<U0032>";"<U0033>";"<U0034>";/
2875 "<U0035>";"<U0036>";"<U0037>"
2876 week 7;19971201;4
2877 abmon "<U0030><U0031>";"<U0030><U0032>";"<U0030><U0033>";/
2878 "<U0030><U0034>";"<U0030><U0035>";"<U0030><U0036>";/
2879 "<U0030><U0037>";"<U0030><U0038>";"<U0030><U0039>";/
2880 "<U0031><U0030>";"<U0031><U0031>";"<U0031><U0032>"
2881 mon "<U0030><U0031>";"<U0030><U0032>";"<U0030><U0033>";/
2882 "<U0030><U0034>";"<U0030><U0035>";"<U0030><U0036>";/
2883 "<U0030><U0037>";"<U0030><U0038>";"<U0030><U0039>";/
2884 "<U0031><U0030>";"<U0031><U0031>";"<U0031><U0032>"
2885 am_pm "";"
2886 % Date formats following ISO 8601
2887 % Appropriate date and time representation (%c)
2888 % "%F %T"
2889 d_t_fmt "<U0025><U0046><U0020><U0025><U0054>"
2890 %
2891 % Appropriate date representation (%x) "%F"
2892 d_fmt "<U0025><U0046>"
2893 %
2894 % Appropriate time representation (%X) "%T"
2895 t_fmt "<U0025><U0054>"
2896 t_fmt_ampm ""
2897 %
2898 END LC_TIME
2899

```

4.8 LC_MESSAGES

The LC_MESSAGES category defines the format and values for affirmative and negative responses. The operands are strings or extended regular expressions to specify which response strings that should be considered matches; see ISO/IEC 9945-2:1993 clause 2.8.4 for a definition of extended regular expressions. The following keywords are defined:

2900	copy	Specify the name of an existing FDCC-set to be used as the source for the
2901		definition of this category. If this keyword is specified, no other keyword
2902		is specified.
2903	yesexpr	The operand consists of an extended regular expression that describes the
2904		acceptable affirmative response to a question expecting an affirmative or
2905		
2906		
2907		
2908		
2909		
2910		
2911		

2912 negative response.
 2913 **noexpr** The operand consists of an extended regular expression that describes the
 2914 acceptable negative response to a question expecting an affirmative or
 2915 negative response.
 2916

2917 The "i18n" LC_MESSAGES category is:
 2918

```

2919 LC_MESSAGES
2920 % This is the ISO/IEC 14652 "i18n" definition for
2921 % the LC_MESSAGES category.
2922 %
2923 yesexpr "<U005B><U002B><U0031><U005D>"
2924 noexpr  "<U005B><U002D><U0030><U005D>"
2925 END LC_MESSAGES
  
```

2927 Note: This uses regular expression syntax with brackets ([]) to for example
 2928 specify the both <+> and <1> is allowed as an affirmative answer.
 2929

2930 4.9 LC_XLITERATE

2931 The LC_XLITERATE category defines formats to transliterate strings, by transforming
 2932 substrings in the source to substrings in the target string. The capabilities are limited to
 2933 simple transliteration based on substring substitution, while more advanced transliteration
 2934 schemes, for example based on pattern matching, is either cumbersome to specify, or not
 2935 addressed. The transliteration may for example be from the Cyrillic script to the Latin
 2936 script.
 2937

2938 Transliteration of an incoming character string to a character string in a FDCC-set can be
 2939 specified with the following transliteration keywords and transliteration statements.
 2940

2941 **copy** Specify the name of an existing FDCC-set to be used as the
 2942 source for the definition of this category. If this keyword is
 2943 specified, no other keyword is specified.
 2944 **include** The name of the FDCC-set in text form to transliterate from,
 2945 and the repertoiremap for the FDCC-set to be used for the
 2946 definition of the transliteration statements. Other
 2947 transliteration statements may follow to replace specification
 2948 of the copied FDCC-set. This keyword is optional.
 2949 **default_missing** defines a string of one or more characters to be used if no
 2950 transliteration statement can be applied to a input
 2951 <transliteration-source>. This keyword is optional.
 2952 **translit_ignore** defines a set of characters, separated by semicolons, that are
 2953 to be ignored in the incoming character string. The characters
 2954 may use the notations defined in 4.3 for lists of characters.
 2955 This keyword is optional.
 2956 **redefine** This keyword introduces a list of transliteration statements
 2957 where each of the <transliteration_source> strings have been
 2958 defined previously in the specification, and the new
 2959 transliteration statements then replaces the old transliteration
 2960 statements for the <transliteration_source> strings specified.
 2961 This keyword is optional.
 2962

2963 4.9.1 Transliteration statements

2964 The syntax for a transliteration statement is:
 2965
 2966

2967 "%s %s;%s;...;%s\n",<transliteration_source>,<transliteration_string>,...
 2968

2969 Each <transliteration_source> consists of one or more characters (in any of the forms
 2970 defined in 4.1.1). The <transliteration_source> that is the longest in terms of number of
 2971 characters that match the input string is the one selected for transliteration.
 2972

2973 If a transliteration statement contains more than one <transliteration_string>, the order that
 2974 each <transliteration_string> occurs in the transliteration statement defines the precedence
 2975 order for choosing a particular <transliteration_string> to substitute for the
 2976 <transliteration_source>. When a process makes use of a transliteration statement to
 2977 transliterate text, and that transliteration statement contains more than one
 2978 <transliteration_string>, that process chooses the first <transliteration_string>, in the
 2979 defined precedence order, that satisfies the requirements of the transliteration.
 2980

2981 Note: the exact definition of the concept of satisfying the requirements of the
 2982 transliteration is outside the context of this Technical Report. If, for example, a
 2983 transliteration involves a change in the coded character set of a string, a
 2984 <transliteration_string> must be chosen, all of whose elements are members of that
 2985 coded character set. In order to determine this, it would be expected that a
 2986 repertoire describing which characters are to be present in the resulting transformed
 2987 string be available to the transliteration API. Also, a transliteration may involve
 2988 requirements such as that string length not change under transliteration. Such
 2989 requirements may also affect the choice among alternative <transliteration_string>
 2990 values.
 2991

2992 If more than one transliteration statement is given for a given <transliteration_source> this
 2993 is an error, and duplicate transliteration statements are ignored. Tailoring of transliteration
 2994 statements may be done via the "redefine" keyword.
 2995

2996 **4.9.2 "include" keyword**

2997
 2998 The "include" keyword specifies a set of transliteration statements in text form to be
 2999 included in the applied transliteration.

3000
 3001 The syntax of the "include" statement is:

3002
 3003 "include %s;%s\n", <FDCC-set>, <repertoiremap>
 3004

3005 <FDCC-set> is a string identifying the FDCC-set to be included from.
 3006

3007 <repertoiremap> is a string identifying the repertoire used in the FDCC-set being
 3008 included, and is used to map character specifications from the specified FDCC-set into the
 3009 current FDCC-set.
 3010

3011 **4.9.3 Example of use of transliteration**

```
3012 LC_XLITERATE
3013 include "de_DE"; "de_repmap"
3014 default_missing <?>
3015 translit_ignore <U3200>..

```

3020
 3021 The "LC_XLITERATE" statement introduces the transliteration category.
 3022

3023 The "include" keyword specifies that the FDCC-set "de_DE" is copied and that the repertoiremap "de_repmap" is
 3024 used to define the symbolic character names in the FDCC-set "de_DE".
 3025

3026 The "default_missing" keyword introduces the character sequence "<?>" as the string to transform into for input
 3027 characters that cannot be transformed into other strings, because no transliteration statement is applicable to the
 3028 character.
 3029

3030 The "translit_ignore" keyword specifies that a set of Ideographic characters, Hangul, East Asian symbols and the
 3031 private use area etc. (the range <U3200>..<>UFAFF>) is ignored for the transliteration.
 3032

3033 The next 3 lines are transliteration statements.
 3034

3035 The first transliteration statement defines a number of transliterations for the LATIN LETTER AE, including into
 3036 LATIN LETTER A WITH DIAERESIS, GREEK LETTER EPSILON, the two Latin letters A and E, and finally
 3037 the LATIN LETTER E.
 3038

3039 The second transliteration statement defines transliteration of the LATIN LETTER S into GREEK LETTER
 3040 SIGMA, and CYRILLIC LETTER ES.
 3041

3042 The third transliteration statement transliterates the two Latin letters K and O into the Japanese Hiragana character
 3043 KO.
 3044

3045 The transliteration category is terminated via the "END LC_XLITERATE" statement in the above example.
 3046

3047 There is no "i18n" entry for the LC_XLITERATE category
 3048

3049 4.10 LC_NAME

3050
 3051 The LC_NAME category defines formats to be used in addressing a person, e.g. in a
 3052 postal address or in a letter. The following keywords are defined:
 3053

3054 **copy** Specify the name of an existing FDCC-set to be used as the source for the
 3055 definition of this category. If this keyword is specified, no other keyword is
 3056 specified.

3057 **name_fmt** Define the appropriate representation of a person's name and title. The
 3058 operand consists of a string, and can contain any combination of characters
 3059 and field descriptors. In addition, the string can contain escape sequences
 3060 defined below.

3061 **name_gen** The operand is a string defining a salutation valid for all persons.

3062 **name_miss** The operand is a string defining a salutation valid for unmarried females.

3063 **name_mr** The operand is a string defining a salutation valid for males.

3064 **name_mrs** The operand is a string defining a salutation valid for married females.

3065 **name_ms** The operand is a string defining a salutation valid for all females.
 3066

3067 NOTE: There are a number of variations for addressing a person among the
 3068 cultures. Middle names are not used in many countries and even the family name
 3069 is not used in some countries. In other countries there is extensive use of one or
 3070 more middle names and corresponding initials. The specification below should be
 3071 regarded as a starting point for this problem.
 3072

3073 The LC_NAME category defines the interpretation of a number of escape sequences. The
 3074 escape sequences are also available in the definitions with the following LC_NAME
 3075 keywords: "name_fmt".
 3076

3077 Escape sequences for the "name_fmt" keyword:
 3078

3079	%f	Family names.
3080	%F	Family names in uppercase.
3081	%g	First given name.
3082	%G	First given initial.
3083	%l	First given name with latin letters.
3084	%o	Other shorter name, eg. "Bill".
3085	%m	Middle names.
3086	%M	Middle initials.
3087	%p	Profession.
3088	%s	Salutation, such as "Doctor"
3089	%S	Abbreviated salutation, such as "Mr." or "Dr."
3090	%d	Salutation, using the FDCC-sets conventions, with 1 for the name_gen, 2 for name_mr, 3 for name_mrs, 4 for name_miss, 5 for name_ms.
3091		
3092	%t	If the preceding escape sequence resulted in an empty string, then the empty string, else a <space>.
3093		
3094		

Each escape sequence may have an <R> after the <%> to specify that the information is taken from a Romanized version string of the entity.

The "i18n" LC_NAME category is:

```

3100 LC_NAME
3101 % This is the ISO/IEC TR 14652 "i18n" definition for
3102 % the LC_NAME category.
3103 %
3104 name_fmt      "<U0025><U0070><U0025><U0074><U0025><U0067><U0025><U0074>/
3105 <U0025><U006D><U0025><U0074><U0025><U0066>"
3106 END LC_NAME
3107

```

4.11 LC_ADDRESS

The LC_ADDRESS category defines formats to be used in specifying a location like a person's living or office, for use in a postal address or in a letter, and other items related to geography. All keywords are optional. The following keywords are recognized:

3113		
3114	copy	Specify the name of an existing FDCC-set to be used as the source for the definition of this category. If this keyword is specified, no other keyword is specified.
3115		
3116		
3117	postal_fmt	Define the appropriate representation of a postal address such as street and city. The proper formatting of a person's name and title is done with the "name_fmt" keyword of the LC_NAME category. The operand consists of a string, and can contain any combination of characters and field descriptors. In addition, the string can contain escape sequences defined below.
3118		
3119		
3120		
3121		
3122		
3123	country_name	The operand is a string with the name of the country in the language of the FDCC-set.
3124		
3125	country_post	The operand is a string with the abbreviation of the country, used for postal addresses, for example by CEPT-MAILCODE.
3126		
3127	lang_name	The operand is a string with the name of the language in the language of the FDCC-set.
3128		
3129	lang_ab2	The operand is a string with the two-letter abbreviation of the language, according to ISO 639.
3130		
3131	lang_ab3_term	The operand is a string with the three-letter abbreviation of the language for terminology use, according to ISO 639-2.
3132		
3133	lang_ab3_lib	The operand is a string with the three-letter abbreviation of the

3134 language for library use, according to ISO 639-2. If not specified, the
3135 value of the "lang_ab3_term" keyword is taken.
3136

3137 Note: The "lang_ab3_term" and "lang_ab3_lib" keywords will in most cases contain the
3138 same value, but they may differ, e.g the valuse for the German language is "deu" and
3139 "ger" respectively.
3140

3141 The LC_ADDRESS category defines the interpretation of a number of escape sequences.
3142 The escape sequences are also available in the definitions with the following
3143 LC_ADDRESS keywords: "postal_fmt".
3144

3145 Escape sequences for the "postal_fmt" keyword:
3146

3147	%a	C/O address.
3148	%f	Firm name.
3149	%d	Department name.
3150	%b	Building name.
3151	%s	Street or block (eg. Japanese) name.
3152	%h	House number or designation.
3153	%N	If any graphical characters have been specified then an end of line is
3154		made.
3155	%t	If the preceding escape sequence resulted in an empty string, then the
3156		empty string, else a <space>.
3157	%r	Room number, door designation.
3158	%e	Floor number.
3159	%C	Country designation, from the <country_post> keyword.
3160	%l	Local township
3161	%z	Zip number, postal code.
3162	%T	Town, city.
3163	%S	State, province, or prefecture.
3164	%c	Country.

3165
3166 Each escape sequence may have an <R> after the <%> to specify that the information is
3167 taken from a Romanized version string of the entity.
3168

3169 NOTE: There are a number of variations for specifying a location among the cultures.
3170 Some of the information, like the middle names, or even the family name, is not used
3171 in some cultures. The specification here should be regarded as a starting point for this
3172 problem.
3173

3174 The "i18n" LC_ADDRESS category is:
3175

```
3176 LC_ADDRESS
3177 % This is the ISO/IEC TR 14652 "i18n" definition for
3178 % the LC_ADDRESS category.
3179 %
3180 postal_fmt      "<U0025><U0061><U0025><U004E><U0025><U0066><U0025><U004E>/
3181 <U0025><U0064><U0025><U004E><U0025><U0062><U0025><U004E><U0025><U0073>/
3182 <U0020><U0025><U0068><U0020><U0025><U0065><U0020><U0025><U0072><U0025>/
3183 <U004E><U0025><U0043><U002D><U0025><U007A><U0020><U0025><U0054><U0025>/
3184 <U004E><U0025><U0063><U0025><U004E>"
3185 END LC_ADDRESS
3186
```

3187

3188 4.12 LC_TELEPHONE

3189

3190 The LC_TELEPHONE category defines formats to be used with telephone services. All
3191 keywords are optional. The following keywords are defined:

3192		
3193	copy	Specify the name of an existing FDCC-set to be used as the source
3194		for the definition of this category. If this keyword is specified, no
3195		other keyword is specified.
3196	tel_int_fmt	Define the appropriate representation of a telephone number for
3197		international use. The operand consists of a string, and can contain
3198		any combination of characters and field descriptors. In addition, the
3199		string can contain escape sequences defined below.
3200	tel_dom_fmt	Define the appropriate representation of a telephone number for
3201		domestic use. The operand consists of a string, and can contain any
3202		combination of characters and field descriptors. In addition, the string
3203		can contain escape sequences defined below.
3204	int_select	The operand is a string with the digits used to call international
3205		telephone numbers.
3206	int_prefix	The operand is a string with the prefix used from other countries to
3207		call the area.
3208		

3209 The LC_TELEPHONE category defines the interpretation of a number of escape
3210 sequences. The escape sequences are also available in the definitions with the following
3211 LC_TELEPHONE keywords: "tel_int_fmt" and "tel_dom_fmt".

3212		
3213	%a	area code without prefix (prefix is often <0>).
3214	%A	area code including prefix (prefix is often <0>).
3215	%l	local number.
3216	%c	country code
3217	%C	alternative carrier service code used for dialling abroad
3218		

3219 The "i18n" LC_TELEPHONE category is:

```
3220
3221 LC_TELEPHONE
3222 % This is the ISO/IEC TR 14652 "i18n" definition for
3223 % the LC_TELEPHONE category.
3224 %
3225 %
3226 tel_int_fmt      "<U002B><U0025><U0063><U0020><U002B><U0061><U0020><U002B>/
3227 <U006C>"
3228 END LC_TELEPHONE
```

3229 5. CHARMAP

3230 A character set description may exist for each coded character set supported by an
3231 application. This text is referred elsewhere in this Technical Report as a charmap.

3232 A conforming charmap to be used with a FDCC-set supports the portable character set
3233 specified in Table 1.

3234 Conforming charmaps specify certain character and character set attributes, as defined in
3235 5.1.

3236 5.1 Character Set Description Text

3237 The character set description text (charmap) describes the mapping between symbolic
3238 character names and actual encoding of a coded character set. It is used to bind the
3239

3245 symbolic character names in a FDCC-set to an actual encoding, so an application can
 3246 process data in this encoding.

3247
 3248 The following declarations can precede the character definitions. Each consist of the
 3249 symbol shown in the following list, starting in column 1, including the surrounding
 3250 brackets, followed by one of more "blank"s, followed by the value to be assigned to the
 3251 symbol. If any of the declarations are included, they are specified in the order shown in
 3252 the following list:

3253		
3254	<code_set_name>	The name of the coded character set for which the character set description text is defined. The characters of the name are taken from the set of characters with visible glyphs defined in Table 1.
3255		
3256		
3257		
3258	<mb_cur_max>	The maximum number of bytes in a multibyte character. This defaults to 1.
3259		
3260		
3261	<mb_cur_min>	An unsigned positive integer value that defines the minimum number of bytes in a character for the encoded character set. The value is less or equal to "mb_cur_max". If not specified, the minimum number is equal to "mb_cur_max".
3262		
3263		
3264		
3265		
3266	<escape_char>	The escape character used to indicate that the characters following is interpreted in a special way, as defined later in this subclause. This defaults to backslash (\). The character slash (/) is used in all the following text and examples, unless otherwise noted.
3267		
3268		
3269		
3270		
3271		
3272	<comment_char>	The character that when placed in column 1 of a charmap line, is used to indicate that the line is ignored. The default character is the number sign (#). The character percent-sign (%) is used in all the following text and examples, unless otherwise noted.
3273		
3274		
3275		
3276		
3277	<repertoiremap>	The name of the repertoiremap used to define the symbolic character names in the charmap. The characters of the name are taken from the set of characters with visible glyphs defined in Table 1.
3278		
3279		
3280		
3281		
3282	<escseq2022>	defines the escape sequences for ISO 2022 shifting for the coded character set defined by the charmap. The semicolon-separated operands are all strings with characters taken from the set of characters with visible glyphs defined in table 1. The first operand defines the g-set or c-set to be defined, and the following values are defined: c0, c1, g0, g1, g2, g3. The second operand defines what range of characters in the charmap is affected, and the values defined are: c0, c1, g0, g1. The third operand is the escape sequence that is defined.
3283		
3284		
3285		
3286		
3287		
3288		
3289		
3290		
3291		
3292	<addset>	the name of the charmap to be added to the current coded character set, and to be selected by the escape sequences defined by <escseq> of the added charmap.
3293		
3294		
3295		
3296	<include>	include the encoding of another charmap in the current charmap.

3297 The semicolon-separated operands are all strings with characters
 3298 taken from the set of characters with visible glyphs defined in
 3299 table 1. The first operand defines the g-set or c-set to be defined
 3300 in the current charmap, and the following values are defined: c0,
 3301 c1, g0, g1, g2, g3. The second operand defines a range of
 3302 characters in the referenced charmap, and the values defined are:
 3303 c0, c1, g0, g1. The third operand is the name of the charmap to
 3304 be included. The coded character sets are defined initially for the
 3305 encoding, and therefore do not need escape sequences for
 3306 identification. If two g0 sets are defined, the second is switched
 3307 to using the SHIFT OUT control character, while the first is
 3308 shifted to using the SHIFT IN control character.
 3309

3310 The character set mapping definitions are all the lines immediately following an identifier
 3311 line containing the string "CHARMAP" starting in column 1, and preceding a trailer line
 3312 containing the string "END CHARMAP" starting in column 1. Empty lines and lines
 3313 containing a <comment_char> in the first column are ignored. Each noncomment line of
 3314 the character set mapping definition (i.e., between the "CHARMAP" and "END
 3315 CHARMAP" lines of the text) is in one of the following syntaxes.
 3316

3317
 3318 "%s %s %s\n", <symbolic-name>,<encoding>,<comments>

3319
 3320 "%s...%s %s %s\n", <symbolic-name>,<symbolic-name>,<encoding>,<comments>

3321
 3322 "%s....%s %s %s\n", <symbolic-name>,<symbolic-name>,<encoding>,<comments>

3323
 3324 "%s..%s %s %s\n", <symbolic-name>,<symbolic-name>,<encoding>,<comments>

3325
 3326 In the first syntax, the line of the character set mapping definition starts with the symbolic
 3327 name, immediately preceded by a <less-than> character and immediately followed by a
 3328 <greater-than> character. Symbolic names only contain characters from the set shown
 3329 with a visible glyph in Table 1.
 3330

3331 The same symbolic name may occur several times, with different values. The first value is
 3332 the one used when generating an encoding, while the other values are accepted in
 3333 decoding. Symbolic names may be included to identify values that can overlap with each
 3334 other or with the values of the symbolic names shown in Table 1. It is possible to specify
 3335 symbolic names for which no encoding exists in the encoded character set, by not
 3336 specifying a value.
 3337

3338 In the second and third syntax (symbolic decimal ellipsis), the line in the character set
 3339 mapping defines a range of one or more symbolic names. The difference between the
 3340 second and the third syntax is the number of dots in the ellipsis: the second has 3 dots, the
 3341 third has 4 dots. In these forms the symbolic names consist of zero or more nonnumeric
 3342 characters from the set shown with visible glyphs in Table 1, followed by an integer
 3343 formed by one or more decimal digits. The characters preceding the integer are identical
 3344 in the two symbolic names, and the integer formed by the digits in the second symbolic
 3345 name are identical to or greater than the integer formed by the digits in the first name.
 3346 This is interpreted as a series of symbolic names formed from the common part and each
 3347 of the integers in decimal format between the first and the second integer, inclusive, and
 3348 with a length of the symbolic names generated that is equal to the length of the first (and

3349 also the second) symbolic name. As an example, <j0101>...<j0104> is interpreted as the
 3350 symbolic names <j0101>, <j0102>, <j0103>, and <j0104>, in that order.
 3351

3352 Note: The rationale to allow both a 3-dot and a 4-dot symbol for symbolic decimal
 3353 ellipses is that in the POSIX standard the decimal symbolic ellipses was defined by a 3-
 3354 dot symbol for charmaps, while the 3-dot symbol was an absolute ellipses for POSIX
 3355 locales, and this Technical Report specifies a 4-dot symbol for the decimal symbolic
 3356 ellipses. The 3-dot symbolic decimal ellipses in charmaps is deprecated.
 3357

3358 In the fourth syntax (symbolic hexadecimal ellipsis, with two dots), the line in the
 3359 character set mapping defines a range of one or more symbolic names. In this form the
 3360 symbolic names consist of zero or more nonnumeric characters from the set shown with
 3361 visible glyphs in Table 1, followed by an integer formed by one or more hexadecimal
 3362 digits, using uppercase letters only for the range "A" to "F". The characters preceding the
 3363 hexadecimal integer are identical in the two symbolic names, and the integer formed by
 3364 the hexadecimal digits in the second symbolic name is identical to or greater than the
 3365 integer formed by the hexadecimal digits in the first name. This is interpreted as a series
 3366 of symbolic names formed from the common part and each of the integers in hexadecimal
 3367 format using uppercase letters only between the first and the second integer, inclusive, and
 3368 with a length of the symbolic names generated that is equal to the length of the first (and
 3369 also the second) symbolic name. As an example, <U010E>..<U0111> is interpreted as the
 3370 symbolic names <U010E>, <U010F>, <U0110>, and <U0111>, in that order.
 3371

3372 The encoding part is expressed as one (for single-byte values) or more concatenated
 3373 decimal, octal or hexadecimal constants (hexadecimal constants is recommended). Decimal
 3374 constants are represented by two or three decimal digits, preceded by the escape character
 3375 and the lowercase letter "d"; for example /d05, /d97, or /d143. Hexadecimal constants are
 3376 represented by two hexadecimal digits, preceded by the escape character and the lowercase
 3377 letter "x"; for example /x05, /x61, or /x8f. Octal constants are represented by two or three
 3378 octal digits, preceded by the escape character; for example /05, /141, or /217. In a
 3379 charmap, each constant should represent an 8 bit byte for portability reasons. Applications
 3380 supporting other byte sizes may allow constants to represent values larger than those that
 3381 can be represented in 8 bit bytes, and to allow additional digits in constants. When
 3382 constants are concatenated for multibyte character values, they may be of different types,
 3383 and interpreted in byte order from the first to the last with the least significant byte of the
 3384 multibyte character specified by the last byte. The manner in which these constants are
 3385 represented in the character stored in the system is application defined. Omitting bytes
 3386 from a multibyte character produces undefined results.
 3387

3388 In lines defining ranges of symbolic names, the encoded value is the value for the first
 3389 symbolic name in the range (the symbolic name preceding the ellipsis). Subsequent
 3390 symbolic names defined by the range have encoding values in increasing order. For
 3391 example the line

3392
 3393 <j0101>...<j0104> /d129/d254
 3394

3395 is interpreted as

3396
 3397 <j0101> /d129/d254
 3398 <j0102> /d129/d255
 3399 <j0103> /d130/d000
 3400 <j0104> /d130/d001

3401 The comments parameter is optional.

3402

3403

3404

Example of using ISO 2022 techniques:

3405

3406

The following example defines two coded character sets, a 7-bit and a 14-bit. They are then merged into one encoding. It is an example on how encodings used in Eastern Asia could be specified.

3407

3408

3409

The 7-bit charmap

3410

3411

```
<escape_char> /
<comment_char> %
% The 7bit charmap defines both control and graphic characters
<code_set_name> "eastern7bit"
<escseq2022> "c0";"c0", "/x21/x40"
<escseq2022> "g0";"g0", "/x28/x48"
<escseq2022> "g1";"g0", "/x29/x48"
<escseq2022> "g2";"g0", "/x2A/x48"
<escseq2022> "g3";"g0", "/x2B/x48"
```

3412

3413

3414

3415

3416

3417

3418

3419

3420

```
CHARMAP
<tab> /x08
<newline> /x0D
<a> /x61
% more character encodings to be defined here
END CHARMAP
```

3421

3422

3423

3424

3425

3426

3427

3428

The 14-bit charmap

3429

3430

3431

```
<escape_char> /
<comment_char> %
<code_set_name> "eastern14bit"
<mb_cur_max> 2
<esqseq> "g0";"g0";" /x24/x40"
<esqseq> "g1";"g0";" /x24/x29/x40"
<esqseq> "g2";"g0";" /x24/x2A/x40"
<esqseq> "g3";"g0";" /x24/x2B/x40"
CHARMAP
<U0165> /d036/d055 % the character codes are only examples
<U0244> /d036/d056
% more character encodings to be defined here
END CHARMAP
```

3432

3433

3434

3435

3436

3437

3438

3439

3440

3441

3442

3443

3444

3445

The merged encoding

3446

3447

3448

3449

3450

3451

3452

3453

3454

3455

3456

3457

3458

3459

3460

3461

```
<escape_char> /
<comment_char> %
<code_set_name> "shift-eastern"
<mb_cur_max> 2
<mb_cur_min> 1
<include> "c0";"c0";"eastern7bit"
<include> "g0";"g0";"eastern7bit"
<include> "g1";"g0";"eastern14bit"
% This defines the g0 values of "eastern14bit" (without the 8th
% bit set) to be the g1 in this encoding (with the 8th bit set).
%
% So the bytes without the 8th bit set is from the "shift7bit"
% coded character set, while bytes with the 8th bit set are from
% the 14-bit set.
```

3462

Another merged encoding using the same charmaps:

3463

3464

3465

3466

3467

3468

3469

```
<escape_char> /
<comment_char> %
<code_set_name> "EUC-eastern"
<mb_cur_max> 2
<mb_cur_min> 1
```

```

3470 <include>          "c0";"c0";"eastern7bit"
3471 <include>          "g0";"g0";"eastern7bit"
3472 <include>          "g0";"g0";"eastern14bit"
3473 % As there are two "g0" sets defined, the first referenced is the
3474 % initial g0 set, while the second can be shifted to via the SHIFT OUT
3475 % control character. The first can then be shifted to by the SHIFT IN
3476 % control character.
3477
3478

```

3479 WIDTH section

3480
3481 After the "END CHARMAP" statement the following declarations may follow. Each
3482 consists of the keyword shown in the following list, starting in column 1, followed by the
3483 value(s) to be associated to the keyword, as defined below.

3484
3485 **WIDTH** An unassigned positive integer value defining the column width for the
3486 characters in the coded character set. Coded character values are defined using symbolic
3487 character names followed by a column width value. Defining a character with more than
3488 one **WIDTH** produces undefined results. The **END WIDTH** keyword is used to terminate
3489 the **WIDTH** definitions.

3490
3491 **WIDTH_DEFAULT** An unsigned positive integer value defining the column width for
3492 any character not listed by one of the **WIDTH** keywords. If no **WIDTH_DEFAULT**
3493 keyword is included in the charmap, the default character width is 1.

3494
3495 Example:

3496
3497 After the "END CHARMAP" statement, a syntax for width definition would be:

```

3498
3499 WIDTH
3500 <A> 1
3501 <B> 1
3502 <j0101>...<j0195> 2
3503 <U3200>..<UFAFF> 2
3504 END WIDTH
3505 WIDTH_DEFAULT 1
3506

```

3507 In this example, the code point values represented by <A> and are assigned a width
3508 of 1. The code point values <j0101>...<j0195> (decimal ellipses) and <U3200>..<UFAFF>
3509 are assigned a width of 2. The last line defines the **DEFAULT_WIDTH** to 1.

3511 6 REPERTOIREMAP

3512
3513 FDCC-set and Charmap sources may be specified in a coded character set independent
3514 way, using symbolic character names. The relation between the symbolic character names
3515 and characters may be specified via a Repertoiremap, which defines the repertoire of
3516 characters defined for a FDCC-set, and the symbolic character names and corresponding
3517 abstract character (by a reference to ISO/IEC 10646).

3518
3519 The repertoire mapping is defined by specifying the symbolic character name and the
3520 ISO/IEC 10646 code position in hexadecimal form (with a preceding 'U') and optionally
3521 the long ISO/IEC 10646 character name in the following syntax:

```

3522 "%s %s %s\n",<symbolic-name>,<10646-short-identifier>,<comments>
3523

```

3524 The symbolic character name and the ISO/IEC 10646 short identifier are each surrounded
 3525 by angle brackets <>, and the fields are separated by one or more spaces or tabs on a line.
 3526 If a right angle bracket or an escape character is used within a symbolic name, it is
 3527 preceded by the escape character. Characters not in ISO/IEC 10646 may be referenced by
 3528 the symbolic character names <P00000000>..<<PF8FFFFFFF>.

3530 The escape character can be redefined from the default reverse solidus (\) with the first
 3531 line of the Repertoiremap containing the string "escape_char" followed by one or more
 3532 spaces or tabs and then the escape character.

3534 Several symbolic character names can refer to the same abstract character, and are then
 3535 used as synonyms in FDCC-sets and charmaps. The set of <U0000>..<<UFFFF> and
 3536 <U00000000>..<<U7FFFFFFF> symbolic names (no lowercase letters) are predefined and
 3537 refers to the corresponding code points of ISO/IEC 10646 with the same short identifier.

3539 The "i18nrep" repertoiremap is defined to accommodate prior art, such as defined in
 3540 Annex G of the ISO/IEC 9945-2:1993 standard, and used by ISO and IEC member bodies
 3541 in their national POSIX locale specifications, and as used in POSIX locales distributed by
 3542 the ISO/IEC POSIX working group and The Open Group. Many POSIX charmaps
 3543 registered with ISO/IEC 15897 use these symbolic names. It also reflects use on the
 3544 Internet, and many of the Internet registered charsets are specified using these symbolic
 3545 names. The "i18nrep" repertoiremap thus facilitates reuse of both POSIX locale data and
 3546 POSIX charmaps with data from this Technical Report. The sequence <a8>..<<z8> are used
 3547 as hooks for tailoring to denote the last accented Latin letter of each of the ISO/IEC 646
 3548 letters <a>..<<z>, so that tailorings that need to have specifications after the last letter of
 3549 such a family, for example to introduce a new letter of an alphabet, can do so with a
 3550 reference that is stable over different versions of the "i18n" FDCC-set. The contents of the
 3551 "i18nrep" repertoiremap is as follows:

```

escape_char /
<NUL>          <U0000>  NULL (NUL)
<SOH>          <U0001>  START OF HEADING (SOH)
<STX>          <U0002>  START OF TEXT (STX)
<ETX>          <U0003>  END OF TEXT (ETX)
<EOT>          <U0004>  END OF TRANSMISSION (EOT)
<ENQ>          <U0005>  ENQUIRY (ENQ)
<ACK>          <U0006>  ACKNOWLEDGE (ACK)
<alert>        <U0007>  BELL (BEL)
<BEL>          <U0007>  BELL (BEL)
<backspace>    <U0008>  BACKSPACE (BS)
<tab>          <U0009>  CHARACTER TABULATION (HT)
<newline>      <U000A>  LINE FEED (LF)
<vertical-tab> <U000B>  LINE TABULATION (VT)
<form-feed>    <U000C>  FORM FEED (FF)
<carriage-return> <U000D>  CARRIAGE RETURN (CR)
<DLE>          <U0010>  DATALINK ESCAPE (DLE)
<DC1>          <U0011>  DEVICE CONTROL ONE (DC1)
<DC2>          <U0012>  DEVICE CONTROL TWO (DC2)
<DC3>          <U0013>  DEVICE CONTROL THREE (DC3)
<DC4>          <U0014>  DEVICE CONTROL FOUR (DC4)
<NAK>          <U0015>  NEGATIVE ACKNOWLEDGE (NAK)
<SYN>          <U0016>  SYNCHRONOUS IDLE (SYN)
<ETB>          <U0017>  END OF TRANSMISSION BLOCK (ETB)
<CAN>          <U0018>  CANCEL (CAN)
<SUB>          <U001A>  SUBSTITUTE (SUB)
<ESC>          <U001B>  ESCAPE (ESC)
<IS4>          <U001C>  FILE SEPARATOR (IS4)
<IS3>          <U001D>  GROUP SEPARATOR (IS3)
<intro>        <U001D>  GROUP SEPARATOR (IS3)
<IS2>          <U001E>  RECORD SEPARATOR (IS2)
<IS1>          <U001F>  UNIT SEPARATOR (IS1)
<DEL>          <U007F>  DELETE (DEL)
<space>        <U0020>  SPACE
<exclamation-mark> <U0021>  EXCLAMATION MARK
<quotation-mark> <U0022>  QUOTATION MARK
<number-sign>  <U0023>  NUMBER SIGN
<dollar-sign>  <U0024>  DOLLAR SIGN
<percent-sign> <U0025>  PERCENT SIGN
<ampersand>    <U0026>  AMPERSAND

```


<apostrophe>	<U0027>	APOSTROPHE
<left-parenthesis>	<U0028>	LEFT PARENTHESIS
<right-parenthesis>	<U0029>	RIGHT PARENTHESIS
<asterisk>	<U002A>	ASTERISK
<plus-sign>	<U002B>	PLUS SIGN
<comma>	<U002C>	COMMA
<hyphen>	<U002D>	HYPHEN-MINUS
<hyphen-minus>	<U002D>	HYPHEN-MINUS
<period>	<U002E>	FULL STOP
<full-stop>	<U002E>	FULL STOP
<slash>	<U002F>	SOLIDUS
<solidus>	<U002F>	SOLIDUS
<zero>	<U0030>	DIGIT ZERO
<one>	<U0031>	DIGIT ONE
<two>	<U0032>	DIGIT TWO
<three>	<U0033>	DIGIT THREE
<four>	<U0034>	DIGIT FOUR
<five>	<U0035>	DIGIT FIVE
<six>	<U0036>	DIGIT SIX
<seven>	<U0037>	DIGIT SEVEN
<eight>	<U0038>	DIGIT EIGHT
<nine>	<U0039>	DIGIT NINE
<colon>	<U003A>	COLON
<semicolon>	<U003B>	SEMICOLON
<less-than-sign>	<U003C>	LESS-THAN SIGN
<equals-sign>	<U003D>	EQUALS SIGN
<greater-than-sign>	<U003E>	GREATER-THAN SIGN
<question-mark>	<U003F>	QUESTION MARK
<commercial-at>	<U0040>	COMMERCIAL AT
<left-square-bracket>	<U005B>	LEFT SQUARE BRACKET
<backslash>	<U005C>	REVERSE SOLIDUS
<reverse-solidus>	<U005C>	REVERSE SOLIDUS
<right-square-bracket>	<U005D>	RIGHT SQUARE BRACKET
<circumflex>	<U005E>	CIRCUMFLEX ACCENT
<circumflex-accent>	<U005E>	CIRCUMFLEX ACCENT
<underscore>	<U005F>	LOW LINE
<low-line>	<U005F>	LOW LINE
<grave-accent>	<U0060>	GRAVE ACCENT
<left-brace>	<U007B>	LEFT CURLY BRACKET
<left-curly-bracket>	<U007B>	LEFT CURLY BRACKET
<vertical-line>	<U007C>	VERTICAL LINE
<right-brace>	<U007D>	RIGHT CURLY BRACKET
<right-curly-bracket>	<U007D>	RIGHT CURLY BRACKET
<tilde>	<U007E>	TILDE
<a8>	<U0252>	Weight indicating the position of the last a
<b8>	<U0182>	Weight indicating the position of the last b
<c8>	<U0255>	Weight indicating the position of the last c
<d8>	<U018D>	Weight indicating the position of the last d
<e8>	<U0264>	Weight indicating the position of the last e
<f8>	<U0191>	Weight indicating the position of the last f
<g8>	<U01A2>	Weight indicating the position of the last g
<h8>	<U02BD>	Weight indicating the position of the last h
<i8>	<U0196>	Weight indicating the position of the last i
<j8>	<U0284>	Weight indicating the position of the last j
<k8>	<U029E>	Weight indicating the position of the last k
<l8>	<U028E>	Weight indicating the position of the last l
<m8>	<U0271>	Weight indicating the position of the last m
<n8>	<U014A>	Weight indicating the position of the last n
<o8>	<U0277>	Weight indicating the position of the last o
<p8>	<U0278>	Weight indicating the position of the last p
<q8>	<U0138>	Weight indicating the position of the last q
<r8>	<U02B6>	Weight indicating the position of the last r
<s8>	<U0286>	Weight indicating the position of the last s
<t8>	<U0287>	Weight indicating the position of the last t
<u8>	<U01B1>	Weight indicating the position of the last u
<v8>	<U028C>	Weight indicating the position of the last v
<w8>	<U028D>	Weight indicating the position of the last w
<x8>	<U216B>	Weight indicating the position of the last x
<y8>	<U01B3>	Weight indicating the position of the last y
<z8>	<U0293>	Weight indicating the position of the last z
<NU>	<U0000>	NULL (NUL)
<SH>	<U0001>	START OF HEADING (SOH)
<SX>	<U0002>	START OF TEXT (STX)
<EX>	<U0003>	END OF TEXT (ETX)
<ET>	<U0004>	END OF TRANSMISSION (EOT)
<EQ>	<U0005>	ENQUIRY (ENQ)
<AK>	<U0006>	ACKNOWLEDGE (ACK)
<BL>	<U0007>	BELL (BEL)
<BS>	<U0008>	BACKSPACE (BS)
<HT>	<U0009>	CHARACTER TABULATION (HT)
<LF>	<U000A>	LINE FEED (LF)
<VT>	<U000B>	LINE TABULATION (VT)
<FF>	<U000C>	FORM FEED (FF)
<CR>	<U000D>	CARRIAGE RETURN (CR)
<SO>	<U000E>	SHIFT OUT (SO)
<SI>	<U000F>	SHIFT IN (SI)

761	<DL>	<U0010>	DATALINK ESCAPE (DLE)
762	<D1>	<U0011>	DEVICE CONTROL ONE (DC1)
763	<D2>	<U0012>	DEVICE CONTROL TWO (DC2)
764	<D3>	<U0013>	DEVICE CONTROL THREE (DC3)
765	<D4>	<U0014>	DEVICE CONTROL FOUR (DC4)
766	<NK>	<U0015>	NEGATIVE ACKNOWLEDGE (NAK)
767	<SY>	<U0016>	SYNCHRONOUS IDLE (SYN)
768	<EB>	<U0017>	END OF TRANSMISSION BLOCK (ETB)
769	<CN>	<U0018>	CANCEL (CAN)
770		<U0019>	END OF MEDIUM (EM)
771	<SB>	<U001A>	SUBSTITUTE (SUB)
772	<EC>	<U001B>	ESCAPE (ESC)
773	<FS>	<U001C>	FILE SEPARATOR (IS4)
774	<GS>	<U001D>	GROUP SEPARATOR (IS3)
775	<RS>	<U001E>	RECORD SEPARATOR (IS2)
776	<US>	<U001F>	UNIT SEPARATOR (IS1)
777	<DT>	<U007F>	DELETE (DEL)
778	<PA>	<U0080>	PADDING CHARACTER (PAD)
779	<HO>	<U0081>	HIGH OCTET PRESET (HOP)
780	<BH>	<U0082>	BREAK PERMITTED HERE (BPH)
781	<NH>	<U0083>	NO BREAK HERE (NBH)
782	<IN>	<U0084>	INDEX (IND)
783	<NL>	<U0085>	NEXT LINE (NEL)
784	<SA>	<U0086>	START OF SELECTED AREA (SSA)
785	<ES>	<U0087>	END OF SELECTED AREA (ESA)
786	<HS>	<U0088>	CHARACTER TABULATION SET (HTS)
787	<HJ>	<U0089>	CHARACTER TABULATION WITH JUSTIFICATION (HTJ)
788	<VS>	<U008A>	LINE TABULATION SET (VTS)
789	<PD>	<U008B>	PARTIAL LINE FORWARD (PLD)
790	<PU>	<U008C>	PARTIAL LINE BACKWARD (PLU)
791	<RI>	<U008D>	REVERSE LINE FEED (RI)
792	<S2>	<U008E>	SINGLE-SHIFT TWO (SS2)
793	<S3>	<U008F>	SINGLE-SHIFT THREE (SS3)
794	<DC>	<U0090>	DEVICE CONTROL STRING (DCS)
795	<P1>	<U0091>	PRIVATE USE ONE (PU1)
796	<P2>	<U0092>	PRIVATE USE TWO (PU2)
797	<TS>	<U0093>	SET TRANSMIT STATE (STS)
798	<CC>	<U0094>	CANCEL CHARACTER (CCH)
799	<MW>	<U0095>	MESSAGE WAITING (MW)
800	<SG>	<U0096>	START OF GUARDED AREA (SPA)
801	<EG>	<U0097>	END OF GUARDED AREA (EPA)
802	<SS>	<U0098>	START OF STRING (SOS)
803	<GC>	<U0099>	SINGLE GRAPHIC CHARACTER INTRODUCER (SGCI)
804	<SC>	<U009A>	SINGLE CHARACTER INTRODUCER (SCI)
805	<CI>	<U009B>	CONTROL SEQUENCE INTRODUCER (CSI)
806	<ST>	<U009C>	STRING TERMINATOR (ST)
807	<OC>	<U009D>	OPERATING SYSTEM COMMAND (OSC)
808	<PM>	<U009E>	PRIVACY MESSAGE (PM)
809	<AC>	<U009F>	APPLICATION PROGRAM COMMAND (APC)
810	<SP>	<U0020>	SPACE
811	<!>	<U0021>	EXCLAMATION MARK
812	<">	<U0022>	QUOTATION MARK
813	<#>	<U0023>	NUMBER SIGN
814	<\$>	<U0024>	DOLLAR SIGN
815	<%>	<U0025>	PERCENT SIGN
816	<&>	<U0026>	AMPERSAND
817	<'>	<U0027>	APOSTROPHE
818	<(>	<U0028>	LEFT PARENTHESIS
819	<)>	<U0029>	RIGHT PARENTHESIS
820	<*>	<U002A>	ASTERISK
821	<+>	<U002B>	PLUS SIGN
822	<, >	<U002C>	COMMA
823	<->	<U002D>	HYPHEN-MINUS
824	<.>	<U002E>	FULL STOP
825	< / >	<U002F>	SOLIDUS
826	<0>	<U0030>	DIGIT ZERO
827	<1>	<U0031>	DIGIT ONE
828	<2>	<U0032>	DIGIT TWO
829	<3>	<U0033>	DIGIT THREE
830	<4>	<U0034>	DIGIT FOUR
831	<5>	<U0035>	DIGIT FIVE
832	<6>	<U0036>	DIGIT SIX
833	<7>	<U0037>	DIGIT SEVEN
834	<8>	<U0038>	DIGIT EIGHT
835	<9>	<U0039>	DIGIT NINE
836	<:>	<U003A>	COLON
837	< ; >	<U003B>	SEMICOLON
838	<<>	<U003C>	LESS-THAN SIGN
839	<=>	<U003D>	EQUALS SIGN
840	< / >>	<U003E>	GREATER-THAN SIGN
841	<?>	<U003F>	QUESTION MARK
842	<@>	<U0040>	COMMERCIAL AT
843	<A>	<U0041>	LATIN CAPITAL LETTER A
844		<U0042>	LATIN CAPITAL LETTER B
845	<C>	<U0043>	LATIN CAPITAL LETTER C
846	<D>	<U0044>	LATIN CAPITAL LETTER D
847	<E>	<U0045>	LATIN CAPITAL LETTER E
848	<F>	<U0046>	LATIN CAPITAL LETTER F

3769	<G>	<U0047>	LATIN CAPITAL LETTER G
3770	<H>	<U0048>	LATIN CAPITAL LETTER H
3771	<I>	<U0049>	LATIN CAPITAL LETTER I
3772	<J>	<U004A>	LATIN CAPITAL LETTER J
3773	<K>	<U004B>	LATIN CAPITAL LETTER K
3774	<L>	<U004C>	LATIN CAPITAL LETTER L
3775	<M>	<U004D>	LATIN CAPITAL LETTER M
3776	<N>	<U004E>	LATIN CAPITAL LETTER N
3777	<O>	<U004F>	LATIN CAPITAL LETTER O
3778	<P>	<U0050>	LATIN CAPITAL LETTER P
3779	<Q>	<U0051>	LATIN CAPITAL LETTER Q
3780	<R>	<U0052>	LATIN CAPITAL LETTER R
3781	<S>	<U0053>	LATIN CAPITAL LETTER S
3782	<T>	<U0054>	LATIN CAPITAL LETTER T
3783	<U>	<U0055>	LATIN CAPITAL LETTER U
3784	<V>	<U0056>	LATIN CAPITAL LETTER V
3785	<W>	<U0057>	LATIN CAPITAL LETTER W
3786	<X>	<U0058>	LATIN CAPITAL LETTER X
3787	<Y>	<U0059>	LATIN CAPITAL LETTER Y
3788	<Z>	<U005A>	LATIN CAPITAL LETTER Z
3789	<<>	<U005B>	LEFT SQUARE BRACKET
3790	<////>	<U005C>	REVERSE SOLIDUS
3791	< />>	<U005D>	RIGHT SQUARE BRACKET
3792	<' />>	<U005E>	CIRCUMFLEX ACCENT
3793	<_>	<U005F>	LOW LINE
3794	<'!>	<U0060>	GRAVE ACCENT
3795	<a>	<U0061>	LATIN SMALL LETTER A
3796		<U0062>	LATIN SMALL LETTER B
3797	<c>	<U0063>	LATIN SMALL LETTER C
3798	<d>	<U0064>	LATIN SMALL LETTER D
3799	<e>	<U0065>	LATIN SMALL LETTER E
3800	<f>	<U0066>	LATIN SMALL LETTER F
3801	<g>	<U0067>	LATIN SMALL LETTER G
3802	<h>	<U0068>	LATIN SMALL LETTER H
3803	<i>	<U0069>	LATIN SMALL LETTER I
3804	<j>	<U006A>	LATIN SMALL LETTER J
3805	<k>	<U006B>	LATIN SMALL LETTER K
3806	<l>	<U006C>	LATIN SMALL LETTER L
3807	<m>	<U006D>	LATIN SMALL LETTER M
3808	<n>	<U006E>	LATIN SMALL LETTER N
3809	<o>	<U006F>	LATIN SMALL LETTER O
3810	<p>	<U0070>	LATIN SMALL LETTER P
3811	<q>	<U0071>	LATIN SMALL LETTER Q
3812	<r>	<U0072>	LATIN SMALL LETTER R
3813	<s>	<U0073>	LATIN SMALL LETTER S
3814	<t>	<U0074>	LATIN SMALL LETTER T
3815	<u>	<U0075>	LATIN SMALL LETTER U
3816	<v>	<U0076>	LATIN SMALL LETTER V
3817	<w>	<U0077>	LATIN SMALL LETTER W
3818	<x>	<U0078>	LATIN SMALL LETTER X
3819	<y>	<U0079>	LATIN SMALL LETTER Y
3820	<z>	<U007A>	LATIN SMALL LETTER Z
3821	<(!>	<U007B>	LEFT CURLY BRACKET
3822	<!!>	<U007C>	VERTICAL LINE
3823	<!)>	<U007D>	RIGHT CURLY BRACKET
3824	<'?>	<U007E>	TILDE
3825	<NS>	<U00A0>	NO-BREAK SPACE
3826	<!I>	<U00A1>	INVERTED EXCLAMATION MARK
3827	<Ct>	<U00A2>	CENT SIGN
3828	<Pd>	<U00A3>	POUND SIGN
3829	<Cu>	<U00A4>	CURRENCY SIGN
3830	<Ye>	<U00A5>	YEN SIGN
3831	<BB>	<U00A6>	BROKEN BAR
3832	<SE>	<U00A7>	SECTION SIGN
3833	<' :>	<U00A8>	DIAERESIS
3834	<Co>	<U00A9>	COPYRIGHT SIGN
3835	<-a>	<U00AA>	FEMININE ORDINAL INDICATOR
3836	<<<>	<U00AB>	LEFT-POINTING DOUBLE ANGLE QUOTATION MARK
3837	<NO>	<U00AC>	NOT SIGN
3838	<-->	<U00AD>	SOFT HYPHEN
3839	<Rg>	<U00AE>	REGISTERED SIGN
3840	<'m>	<U00AF>	MACRON
3841	<DG>	<U00B0>	DEGREE SIGN
3842	<+>	<U00B1>	PLUS-MINUS SIGN
3843	<2S>	<U00B2>	SUPERSCRIP TWO
3844	<3S>	<U00B3>	SUPERSCRIP THREE
3845	<' ' >	<U00B4>	ACUTE ACCENT
3846	<My>	<U00B5>	MICRO SIGN
3847	<PI>	<U00B6>	PILCROW SIGN
3848	<.M>	<U00B7>	MIDDLE DOT
3849	<' , >	<U00B8>	CEDILLA
3850	<1S>	<U00B9>	SUPERSCRIP ONE
3851	<-o>	<U00BA>	MASCULINE ORDINAL INDICATOR
3852	</>/>>	<U00BB>	RIGHT-POINTING DOUBLE ANGLE QUOTATION MARK
3853	<14>	<U00BC>	VULGAR FRACTION ONE QUARTER
3854	<12>	<U00BD>	VULGAR FRACTION ONE HALF
3855	<34>	<U00BE>	VULGAR FRACTION THREE QUARTERS
3856	<?I>	<U00BF>	INVERTED QUESTION MARK

<A!>	<U00C0>	LATIN CAPITAL LETTER A WITH GRAVE
<A'>	<U00C1>	LATIN CAPITAL LETTER A WITH ACUTE
<A/>>	<U00C2>	LATIN CAPITAL LETTER A WITH CIRCUMFLEX
<A?>	<U00C3>	LATIN CAPITAL LETTER A WITH TILDE
<A:>	<U00C4>	LATIN CAPITAL LETTER A WITH DIAERESIS
<AA>	<U00C5>	LATIN CAPITAL LETTER A WITH RING ABOVE
<AE>	<U00C6>	LATIN CAPITAL LETTER AE (ash)
<C.>	<U00C7>	LATIN CAPITAL LETTER C WITH CEDILLA
<E!>	<U00C8>	LATIN CAPITAL LETTER E WITH GRAVE
<E'>	<U00C9>	LATIN CAPITAL LETTER E WITH ACUTE
<E/>>	<U00CA>	LATIN CAPITAL LETTER E WITH CIRCUMFLEX
<E:>	<U00CB>	LATIN CAPITAL LETTER E WITH DIAERESIS
<I!>	<U00CC>	LATIN CAPITAL LETTER I WITH GRAVE
<I'>	<U00CD>	LATIN CAPITAL LETTER I WITH ACUTE
<I/>>	<U00CE>	LATIN CAPITAL LETTER I WITH CIRCUMFLEX
<I:>	<U00CF>	LATIN CAPITAL LETTER I WITH DIAERESIS
<D->	<U00D0>	LATIN CAPITAL LETTER ETH (Icelandic)
<N?>	<U00D1>	LATIN CAPITAL LETTER N WITH TILDE
<O!>	<U00D2>	LATIN CAPITAL LETTER O WITH GRAVE
<O'>	<U00D3>	LATIN CAPITAL LETTER O WITH ACUTE
<O/>>	<U00D4>	LATIN CAPITAL LETTER O WITH CIRCUMFLEX
<O?>	<U00D5>	LATIN CAPITAL LETTER O WITH TILDE
<O:>	<U00D6>	LATIN CAPITAL LETTER O WITH DIAERESIS
<*X>	<U00D7>	MULTIPLICATION SIGN
<O//>	<U00D8>	LATIN CAPITAL LETTER O WITH STROKE
<U!>	<U00D9>	LATIN CAPITAL LETTER U WITH GRAVE
<U'>	<U00DA>	LATIN CAPITAL LETTER U WITH ACUTE
<U/>>	<U00DB>	LATIN CAPITAL LETTER U WITH CIRCUMFLEX
<U:>	<U00DC>	LATIN CAPITAL LETTER U WITH DIAERESIS
<Y'>	<U00DD>	LATIN CAPITAL LETTER Y WITH ACUTE
<TH>	<U00DE>	LATIN CAPITAL LETTER THORN (Icelandic)
<ss>	<U00DF>	LATIN SMALL LETTER SHARP S (German)
<a!>	<U00E0>	LATIN SMALL LETTER A WITH GRAVE
<a'>	<U00E1>	LATIN SMALL LETTER A WITH ACUTE
<a/>>	<U00E2>	LATIN SMALL LETTER A WITH CIRCUMFLEX
<a?>	<U00E3>	LATIN SMALL LETTER A WITH TILDE
<a:>	<U00E4>	LATIN SMALL LETTER A WITH DIAERESIS
<aa>	<U00E5>	LATIN SMALL LETTER A WITH RING ABOVE
<ae>	<U00E6>	LATIN SMALL LETTER AE (ash)
<c.>	<U00E7>	LATIN SMALL LETTER C WITH CEDILLA
<e!>	<U00E8>	LATIN SMALL LETTER E WITH GRAVE
<e'>	<U00E9>	LATIN SMALL LETTER E WITH ACUTE
<e/>>	<U00EA>	LATIN SMALL LETTER E WITH CIRCUMFLEX
<e:>	<U00EB>	LATIN SMALL LETTER E WITH DIAERESIS
<i!>	<U00EC>	LATIN SMALL LETTER I WITH GRAVE
<i'>	<U00ED>	LATIN SMALL LETTER I WITH ACUTE
<i/>>	<U00EE>	LATIN SMALL LETTER I WITH CIRCUMFLEX
<i:>	<U00EF>	LATIN SMALL LETTER I WITH DIAERESIS
<d->	<U00F0>	LATIN SMALL LETTER ETH (Icelandic)
<n?>	<U00F1>	LATIN SMALL LETTER N WITH TILDE
<o!>	<U00F2>	LATIN SMALL LETTER O WITH GRAVE
<o'>	<U00F3>	LATIN SMALL LETTER O WITH ACUTE
<o/>>	<U00F4>	LATIN SMALL LETTER O WITH CIRCUMFLEX
<o?>	<U00F5>	LATIN SMALL LETTER O WITH TILDE
<o:>	<U00F6>	LATIN SMALL LETTER O WITH DIAERESIS
<-:>	<U00F7>	DIVISION SIGN
<o//>	<U00F8>	LATIN SMALL LETTER O WITH STROKE
<u!>	<U00F9>	LATIN SMALL LETTER U WITH GRAVE
<u'>	<U00FA>	LATIN SMALL LETTER U WITH ACUTE
<u/>>	<U00FB>	LATIN SMALL LETTER U WITH CIRCUMFLEX
<u:>	<U00FC>	LATIN SMALL LETTER U WITH DIAERESIS
<y'>	<U00FD>	LATIN SMALL LETTER Y WITH ACUTE
<th>	<U00FE>	LATIN SMALL LETTER THORN (Icelandic)
<y:>	<U00FF>	LATIN SMALL LETTER Y WITH DIAERESIS
<A->	<U0100>	LATIN CAPITAL LETTER A WITH MACRON
<a->	<U0101>	LATIN SMALL LETTER A WITH MACRON
<A>	<U0102>	LATIN CAPITAL LETTER A WITH BREVE
<a>	<U0103>	LATIN SMALL LETTER A WITH BREVE
<A;~>	<U0104>	LATIN CAPITAL LETTER A WITH OGONEK
<a;~>	<U0105>	LATIN SMALL LETTER A WITH OGONEK
<C'>	<U0106>	LATIN CAPITAL LETTER C WITH ACUTE
<c'>	<U0107>	LATIN SMALL LETTER C WITH ACUTE
<C/>>	<U0108>	LATIN CAPITAL LETTER C WITH CIRCUMFLEX
<c/>>	<U0109>	LATIN SMALL LETTER C WITH CIRCUMFLEX
<C.>	<U010A>	LATIN CAPITAL LETTER C WITH DOT ABOVE
<c.>	<U010B>	LATIN SMALL LETTER C WITH DOT ABOVE
<C<>	<U010C>	LATIN CAPITAL LETTER C WITH CARON
<c<>	<U010D>	LATIN SMALL LETTER C WITH CARON
<D<>	<U010E>	LATIN CAPITAL LETTER D WITH CARON
<d<>	<U010F>	LATIN SMALL LETTER D WITH CARON
<D//>	<U0110>	LATIN CAPITAL LETTER D WITH STROKE
<d//>	<U0111>	LATIN SMALL LETTER D WITH STROKE
<E->	<U0112>	LATIN CAPITAL LETTER E WITH MACRON
<e->	<U0113>	LATIN SMALL LETTER E WITH MACRON
<E>	<U0114>	LATIN CAPITAL LETTER E WITH BREVE
<e>	<U0115>	LATIN SMALL LETTER E WITH BREVE
<E.>	<U0116>	LATIN CAPITAL LETTER E WITH DOT ABOVE
<e.>	<U0117>	LATIN SMALL LETTER E WITH DOT ABOVE

<E; >	<U0118>	LATIN CAPITAL LETTER E WITH OGONEK
<e; >	<U0119>	LATIN SMALL LETTER E WITH OGONEK
<E< >	<U011A>	LATIN CAPITAL LETTER E WITH CARON
<e< >	<U011B>	LATIN SMALL LETTER E WITH CARON
<G/> >	<U011C>	LATIN CAPITAL LETTER G WITH CIRCUMFLEX
<g/> >	<U011D>	LATIN SMALL LETTER G WITH CIRCUMFLEX
<G(> >	<U011E>	LATIN CAPITAL LETTER G WITH BREVE
<g(> >	<U011F>	LATIN SMALL LETTER G WITH BREVE
<G.> >	<U0120>	LATIN CAPITAL LETTER G WITH DOT ABOVE
<g.> >	<U0121>	LATIN SMALL LETTER G WITH DOT ABOVE
<G.> >	<U0122>	LATIN CAPITAL LETTER G WITH CEDILLA
<g.> >	<U0123>	LATIN SMALL LETTER G WITH CEDILLA
<H/> >	<U0124>	LATIN CAPITAL LETTER H WITH CIRCUMFLEX
<h/> >	<U0125>	LATIN SMALL LETTER H WITH CIRCUMFLEX
<H//> >	<U0126>	LATIN CAPITAL LETTER H WITH STROKE
<h//> >	<U0127>	LATIN SMALL LETTER H WITH STROKE
<I?> >	<U0128>	LATIN CAPITAL LETTER I WITH TILDE
<i?> >	<U0129>	LATIN SMALL LETTER I WITH TILDE
<I-> >	<U012A>	LATIN CAPITAL LETTER I WITH MACRON
<i-> >	<U012B>	LATIN SMALL LETTER I WITH MACRON
<I(> >	<U012C>	LATIN CAPITAL LETTER I WITH BREVE
<i(> >	<U012D>	LATIN SMALL LETTER I WITH BREVE
<I; >	<U012E>	LATIN CAPITAL LETTER I WITH OGONEK
<i; >	<U012F>	LATIN SMALL LETTER I WITH OGONEK
<I.> >	<U0130>	LATIN CAPITAL LETTER I WITH DOT ABOVE
<i.> >	<U0131>	LATIN SMALL LETTER DOTLESS I
<IJ> >	<U0132>	LATIN CAPITAL LIGATURE IJ
<ij> >	<U0133>	LATIN SMALL LIGATURE IJ
<J/> >	<U0134>	LATIN CAPITAL LETTER J WITH CIRCUMFLEX
<j/> >	<U0135>	LATIN SMALL LETTER J WITH CIRCUMFLEX
<K.> >	<U0136>	LATIN CAPITAL LETTER K WITH CEDILLA
<k.> >	<U0137>	LATIN SMALL LETTER K WITH CEDILLA
<kk> >	<U0138>	LATIN SMALL LETTER KRA (Greenlandic)
<L' >	<U0139>	LATIN CAPITAL LETTER L WITH ACUTE
<l' >	<U013A>	LATIN SMALL LETTER L WITH ACUTE
<L.> >	<U013B>	LATIN CAPITAL LETTER L WITH CEDILLA
<l.> >	<U013C>	LATIN SMALL LETTER L WITH CEDILLA
<L<> >	<U013D>	LATIN CAPITAL LETTER L WITH CARON
<l<> >	<U013E>	LATIN SMALL LETTER L WITH CARON
<L.> >	<U013F>	LATIN CAPITAL LETTER L WITH MIDDLE DOT
<l.> >	<U0140>	LATIN SMALL LETTER L WITH MIDDLE DOT
<L//> >	<U0141>	LATIN CAPITAL LETTER L WITH STROKE
<l//> >	<U0142>	LATIN SMALL LETTER L WITH STROKE
<N' >	<U0143>	LATIN CAPITAL LETTER N WITH ACUTE
<n' >	<U0144>	LATIN SMALL LETTER N WITH ACUTE
<N.> >	<U0145>	LATIN CAPITAL LETTER N WITH CEDILLA
<n.> >	<U0146>	LATIN SMALL LETTER N WITH CEDILLA
<N<> >	<U0147>	LATIN CAPITAL LETTER N WITH CARON
<n<> >	<U0148>	LATIN SMALL LETTER N WITH CARON
<'n> >	<U0149>	LATIN SMALL LETTER N PRECEDED BY APOSTROPHE
<NG> >	<U014A>	LATIN CAPITAL LETTER ENG (Sami)
<ng> >	<U014B>	LATIN SMALL LETTER ENG (Sami)
<O-> >	<U014C>	LATIN CAPITAL LETTER O WITH MACRON
<o-> >	<U014D>	LATIN SMALL LETTER O WITH MACRON
<O(> >	<U014E>	LATIN CAPITAL LETTER O WITH BREVE
<o(> >	<U014F>	LATIN SMALL LETTER O WITH BREVE
<O" >	<U0150>	LATIN CAPITAL LETTER O WITH DOUBLE ACUTE
<o" >	<U0151>	LATIN SMALL LETTER O WITH DOUBLE ACUTE
<OE> >	<U0152>	LATIN CAPITAL LIGATURE OE
<oe> >	<U0153>	LATIN SMALL LIGATURE OE
<R' >	<U0154>	LATIN CAPITAL LETTER R WITH ACUTE
<r' >	<U0155>	LATIN SMALL LETTER R WITH ACUTE
<R.> >	<U0156>	LATIN CAPITAL LETTER R WITH CEDILLA
<r.> >	<U0157>	LATIN SMALL LETTER R WITH CEDILLA
<R<> >	<U0158>	LATIN CAPITAL LETTER R WITH CARON
<r<> >	<U0159>	LATIN SMALL LETTER R WITH CARON
<S' >	<U015A>	LATIN CAPITAL LETTER S WITH ACUTE
<s' >	<U015B>	LATIN SMALL LETTER S WITH ACUTE
<S/> >	<U015C>	LATIN CAPITAL LETTER S WITH CIRCUMFLEX
<s/> >	<U015D>	LATIN SMALL LETTER S WITH CIRCUMFLEX
<S.> >	<U015E>	LATIN CAPITAL LETTER S WITH CEDILLA
<s.> >	<U015F>	LATIN SMALL LETTER S WITH CEDILLA
<S<> >	<U0160>	LATIN CAPITAL LETTER S WITH CARON
<s<> >	<U0161>	LATIN SMALL LETTER S WITH CARON
<T.> >	<U0162>	LATIN CAPITAL LETTER T WITH CEDILLA
<t.> >	<U0163>	LATIN SMALL LETTER T WITH CEDILLA
<T<> >	<U0164>	LATIN CAPITAL LETTER T WITH CARON
<t<> >	<U0165>	LATIN SMALL LETTER T WITH CARON
<T//> >	<U0166>	LATIN CAPITAL LETTER T WITH STROKE
<t//> >	<U0167>	LATIN SMALL LETTER T WITH STROKE
<U?> >	<U0168>	LATIN CAPITAL LETTER U WITH TILDE
<u?> >	<U0169>	LATIN SMALL LETTER U WITH TILDE
<U-> >	<U016A>	LATIN CAPITAL LETTER U WITH MACRON
<u-> >	<U016B>	LATIN SMALL LETTER U WITH MACRON
<U(> >	<U016C>	LATIN CAPITAL LETTER U WITH BREVE
<u(> >	<U016D>	LATIN SMALL LETTER U WITH BREVE
<U0> >	<U016E>	LATIN CAPITAL LETTER U WITH RING ABOVE
<u0> >	<U016F>	LATIN SMALL LETTER U WITH RING ABOVE

4	<U" >	<U0170>	LATIN CAPITAL LETTER U WITH DOUBLE ACUTE
4	<u" >	<U0171>	LATIN SMALL LETTER U WITH DOUBLE ACUTE
4	<U; >	<U0172>	LATIN CAPITAL LETTER U WITH OGONEK
4	<u; >	<U0173>	LATIN SMALL LETTER U WITH OGONEK
4	<W/ >>	<U0174>	LATIN CAPITAL LETTER W WITH CIRCUMFLEX
4	<w/ >>	<U0175>	LATIN SMALL LETTER W WITH CIRCUMFLEX
4	<Y/ >>	<U0176>	LATIN CAPITAL LETTER Y WITH CIRCUMFLEX
4	<y/ >>	<U0177>	LATIN SMALL LETTER Y WITH CIRCUMFLEX
4	<Y: >	<U0178>	LATIN CAPITAL LETTER Y WITH DIAERESIS
4	<Z' >	<U0179>	LATIN CAPITAL LETTER Z WITH ACUTE
4	<z' >	<U017A>	LATIN SMALL LETTER Z WITH ACUTE
4	<Z. >	<U017B>	LATIN CAPITAL LETTER Z WITH DOT ABOVE
4	<z. >	<U017C>	LATIN SMALL LETTER Z WITH DOT ABOVE
4	<Z< >	<U017D>	LATIN CAPITAL LETTER Z WITH CARON
4	<z< >	<U017E>	LATIN SMALL LETTER Z WITH CARON
4	<s1 >	<U017F>	LATIN SMALL LETTER LONG S
4	<b/ / >	<U0180>	LATIN SMALL LETTER B WITH STROKE
4	<B2 >	<U0181>	LATIN CAPITAL LETTER B WITH HOOK
4	<C2 >	<U0187>	LATIN CAPITAL LETTER C WITH HOOK
4	<c2 >	<U0188>	LATIN SMALL LETTER C WITH HOOK
4	<F2 >	<U0191>	LATIN CAPITAL LETTER F WITH HOOK
4	<f2 >	<U0192>	LATIN SMALL LETTER F WITH HOOK
4	<K2 >	<U0198>	LATIN CAPITAL LETTER K WITH HOOK
4	<k2 >	<U0199>	LATIN SMALL LETTER K WITH HOOK
4	<O9 >	<U01A0>	LATIN CAPITAL LETTER O WITH HORN
4	<o9 >	<U01A1>	LATIN SMALL LETTER O WITH HORN
4	<OI >	<U01A2>	LATIN CAPITAL LETTER OI
4	<oi >	<U01A3>	LATIN SMALL LETTER OI
4	<yr >	<U01A6>	LATIN LETTER YR
4	<U9 >	<U01AF>	LATIN CAPITAL LETTER U WITH HORN
4	<u9 >	<U01B0>	LATIN SMALL LETTER U WITH HORN
4	<Z/ / >	<U01B5>	LATIN CAPITAL LETTER Z WITH STROKE
4	<z/ / >	<U01B6>	LATIN SMALL LETTER Z WITH STROKE
4	<ED >	<U01B7>	LATIN CAPITAL LETTER EZH
4	<DZ< >	<U01C4>	LATIN CAPITAL LETTER DZ WITH CARON
4	<Dz< >	<U01C5>	LATIN CAPITAL LETTER D WITH SMALL LETTER Z WITH CARON
4	<dz< >	<U01C6>	LATIN SMALL LETTER DZ WITH CARON
4	<LJ3 >	<U01C7>	LATIN CAPITAL LETTER LJ
4	<Lj3 >	<U01C8>	LATIN CAPITAL LETTER L WITH SMALL LETTER J
4	<lj3 >	<U01C9>	LATIN SMALL LETTER LJ
4	<NJ3 >	<U01CA>	LATIN CAPITAL LETTER NJ
4	<Nj3 >	<U01CB>	LATIN CAPITAL LETTER N WITH SMALL LETTER J
4	<nj3 >	<U01CC>	LATIN SMALL LETTER NJ
4	<A< >	<U01CD>	LATIN CAPITAL LETTER A WITH CARON
4	<a< >	<U01CE>	LATIN SMALL LETTER A WITH CARON
4	<I< >	<U01CF>	LATIN CAPITAL LETTER I WITH CARON
4	<i< >	<U01D0>	LATIN SMALL LETTER I WITH CARON
4	<O< >	<U01D1>	LATIN CAPITAL LETTER O WITH CARON
4	<o< >	<U01D2>	LATIN SMALL LETTER O WITH CARON
4	<U< >	<U01D3>	LATIN CAPITAL LETTER U WITH CARON
4	<u< >	<U01D4>	LATIN SMALL LETTER U WITH CARON
4	<U: - >	<U01D5>	LATIN CAPITAL LETTER U WITH DIAERESIS AND MACRON
4	<u: - >	<U01D6>	LATIN SMALL LETTER U WITH DIAERESIS AND MACRON
4	<U: ' >	<U01D7>	LATIN CAPITAL LETTER U WITH DIAERESIS AND ACUTE
4	<u: ' >	<U01D8>	LATIN SMALL LETTER U WITH DIAERESIS AND ACUTE
4	<U: < >	<U01D9>	LATIN CAPITAL LETTER U WITH DIAERESIS AND CARON
4	<u: < >	<U01DA>	LATIN SMALL LETTER U WITH DIAERESIS AND CARON
4	<U: ! >	<U01DB>	LATIN CAPITAL LETTER U WITH DIAERESIS AND GRAVE
4	<u: ! >	<U01DC>	LATIN SMALL LETTER U WITH DIAERESIS AND GRAVE
4	<e1 >	<U01DD>	LATIN SMALL LETTER TURNED E
4	<A1 >	<U01DE>	LATIN CAPITAL LETTER A WITH DIAERESIS AND MACRON
4	<a1 >	<U01DF>	LATIN SMALL LETTER A WITH DIAERESIS AND MACRON
4	<A7 >	<U01E0>	LATIN CAPITAL LETTER A WITH DOT ABOVE AND MACRON
4	<a7 >	<U01E1>	LATIN SMALL LETTER A WITH DOT ABOVE AND MACRON
4	<A3 >	<U01E2>	LATIN CAPITAL LETTER AE WITH MACRON (ash)
4	<a3 >	<U01E3>	LATIN SMALL LETTER AE WITH MACRON (ash)
4	<G/ / >	<U01E4>	LATIN CAPITAL LETTER G WITH STROKE
4	<g/ / >	<U01E5>	LATIN SMALL LETTER G WITH STROKE
4	<G< >	<U01E6>	LATIN CAPITAL LETTER G WITH CARON
4	<g< >	<U01E7>	LATIN SMALL LETTER G WITH CARON
4	<K< >	<U01E8>	LATIN CAPITAL LETTER K WITH CARON
4	<k< >	<U01E9>	LATIN SMALL LETTER K WITH CARON
4	<O; >	<U01EA>	LATIN CAPITAL LETTER O WITH OGONEK
4	<o; >	<U01EB>	LATIN SMALL LETTER O WITH OGONEK
4	<O1 >	<U01EC>	LATIN CAPITAL LETTER O WITH OGONEK AND MACRON
4	<o1 >	<U01ED>	LATIN SMALL LETTER O WITH OGONEK AND MACRON
4	<EZ >	<U01EE>	LATIN CAPITAL LETTER EZH WITH CARON
4	<ez >	<U01EF>	LATIN SMALL LETTER EZH WITH CARON
4	<j< >	<U01F0>	LATIN SMALL LETTER J WITH CARON
4	<DZ3 >	<U01F1>	LATIN CAPITAL LETTER DZ
4	<Dz3 >	<U01F2>	LATIN CAPITAL LETTER D WITH SMALL LETTER Z
4	<dz3 >	<U01F3>	LATIN SMALL LETTER DZ
4	<G' >	<U01F4>	LATIN CAPITAL LETTER G WITH ACUTE
4	<g' >	<U01F5>	LATIN SMALL LETTER G WITH ACUTE
4	<AA' >	<U01FA>	LATIN CAPITAL LETTER A WITH RING ABOVE AND ACUTE
4	<aa' >	<U01FB>	LATIN SMALL LETTER A WITH RING ABOVE AND ACUTE
4	<AE' >	<U01FC>	LATIN CAPITAL LETTER AE WITH ACUTE (ash)
4	<ae' >	<U01FD>	LATIN SMALL LETTER AE WITH ACUTE (ash)

4	<O//>	<U01FE>	LATIN CAPITAL LETTER O WITH STROKE AND ACUTE
4	<o//>	<U01FF>	LATIN SMALL LETTER O WITH STROKE AND ACUTE
4	<A!!>	<U0200>	LATIN CAPITAL LETTER A WITH DOUBLE GRAVE
4	<a!!>	<U0201>	LATIN SMALL LETTER A WITH DOUBLE GRAVE
4	<A>	<U0202>	LATIN CAPITAL LETTER A WITH INVERTED BREVE
4	<a>	<U0203>	LATIN SMALL LETTER A WITH INVERTED BREVE
4	<E!!>	<U0204>	LATIN CAPITAL LETTER E WITH DOUBLE GRAVE
4	<e!!>	<U0205>	LATIN SMALL LETTER E WITH DOUBLE GRAVE
4	<E>	<U0206>	LATIN CAPITAL LETTER E WITH INVERTED BREVE
4	<e>	<U0207>	LATIN SMALL LETTER E WITH INVERTED BREVE
4	<I!!>	<U0208>	LATIN CAPITAL LETTER I WITH DOUBLE GRAVE
4	<i!!>	<U0209>	LATIN SMALL LETTER I WITH DOUBLE GRAVE
4	<I>	<U020A>	LATIN CAPITAL LETTER I WITH INVERTED BREVE
4	<i>	<U020B>	LATIN SMALL LETTER I WITH INVERTED BREVE
4	<O!!>	<U020C>	LATIN CAPITAL LETTER O WITH DOUBLE GRAVE
4	<o!!>	<U020D>	LATIN SMALL LETTER O WITH DOUBLE GRAVE
4	<O>	<U020E>	LATIN CAPITAL LETTER O WITH INVERTED BREVE
4	<o>	<U020F>	LATIN SMALL LETTER O WITH INVERTED BREVE
4	<R!!>	<U0210>	LATIN CAPITAL LETTER R WITH DOUBLE GRAVE
4	<r!!>	<U0211>	LATIN SMALL LETTER R WITH DOUBLE GRAVE
4	<R>	<U0212>	LATIN CAPITAL LETTER R WITH INVERTED BREVE
4	<r>	<U0213>	LATIN SMALL LETTER R WITH INVERTED BREVE
4	<U!!>	<U0214>	LATIN CAPITAL LETTER U WITH DOUBLE GRAVE
4	<u!!>	<U0215>	LATIN SMALL LETTER U WITH DOUBLE GRAVE
4	<U>	<U0216>	LATIN CAPITAL LETTER U WITH INVERTED BREVE
4	<u>	<U0217>	LATIN SMALL LETTER U WITH INVERTED BREVE
4	<r1>	<U027C>	LATIN SMALL LETTER R WITH LONG LEG
4	<ed>	<U0292>	LATIN SMALL LETTER EZH
4	<;S>	<U02BB>	MODIFIER LETTER TURNED COMMA
4	<1/>	<U02C6>	MODIFIER LETTER CIRCUMFLEX ACCENT
4	<'<>	<U02C7>	CARON (Mandarin Chinese third tone)
4	<1->	<U02C9>	MODIFIER LETTER MACRON (Mandarin Chinese first tone)
4	<1!>	<U02CB>	MODIFIER LETTER GRAVE ACCENT (Mandarin Chinese fourth tone)
4	<'>	<U02D8>	BREVE
4	<'>	<U02D9>	DOT ABOVE (Mandarin Chinese light tone)
4	<'0>	<U02DA>	RING ABOVE
4	<'>	<U02DB>	OGONEK
4	<1?>	<U02DC>	SMALL TILDE
4	<' " >	<U02DD>	DOUBLE ACUTE ACCENT
4	<'G>	<U0374>	GREEK NUMERAL SIGN (Dexia keraia)
4	<,G>	<U0375>	GREEK LOWER NUMERAL SIGN (Aristeri keraia)
4	<j3>	<U037A>	GREEK YPOGEGRAMMENI
4	<?%>	<U037E>	GREEK QUESTION MARK (Erotimatiko)
4	<' * >	<U0384>	GREEK TONOS
4	<' % >	<U0385>	GREEK DIALYTIKA TONOS
4	<A%>	<U0386>	GREEK CAPITAL LETTER ALPHA WITH TONOS
4	<. * >	<U0387>	GREEK ANO TELEIA
4	<E%>	<U0388>	GREEK CAPITAL LETTER EPSILON WITH TONOS
4	<Y%>	<U0389>	GREEK CAPITAL LETTER ETA WITH TONOS
4	<I%>	<U038A>	GREEK CAPITAL LETTER IOTA WITH TONOS
4	<O%>	<U038C>	GREEK CAPITAL LETTER OMICRON WITH TONOS
4	<U%>	<U038E>	GREEK CAPITAL LETTER UPSILON WITH TONOS
4	<W%>	<U038F>	GREEK CAPITAL LETTER OMEGA WITH TONOS
4	<i3>	<U0390>	GREEK SMALL LETTER IOTA WITH DIALYTIKA AND TONOS
4	<A* >	<U0391>	GREEK CAPITAL LETTER ALPHA
4	<B* >	<U0392>	GREEK CAPITAL LETTER BETA
4	<G* >	<U0393>	GREEK CAPITAL LETTER GAMMA
4	<D* >	<U0394>	GREEK CAPITAL LETTER DELTA
4	<E* >	<U0395>	GREEK CAPITAL LETTER EPSILON
4	<Z* >	<U0396>	GREEK CAPITAL LETTER ZETA
4	<Y* >	<U0397>	GREEK CAPITAL LETTER ETA
4	<H* >	<U0398>	GREEK CAPITAL LETTER THETA
4	<I* >	<U0399>	GREEK CAPITAL LETTER IOTA
4	<K* >	<U039A>	GREEK CAPITAL LETTER KAPPA
4	<L* >	<U039B>	GREEK CAPITAL LETTER LAMDA
4	<M* >	<U039C>	GREEK CAPITAL LETTER MU
4	<N* >	<U039D>	GREEK CAPITAL LETTER NU
4	<C* >	<U039E>	GREEK CAPITAL LETTER XI
4	<O* >	<U039F>	GREEK CAPITAL LETTER OMICRON
4	<P* >	<U03A0>	GREEK CAPITAL LETTER PI
4	<R* >	<U03A1>	GREEK CAPITAL LETTER RHO
4	<S* >	<U03A3>	GREEK CAPITAL LETTER SIGMA
4	<T* >	<U03A4>	GREEK CAPITAL LETTER TAU
4	<U* >	<U03A5>	GREEK CAPITAL LETTER UPSILON
4	<F* >	<U03A6>	GREEK CAPITAL LETTER PHI
4	<X* >	<U03A7>	GREEK CAPITAL LETTER CHI
4	<Q* >	<U03A8>	GREEK CAPITAL LETTER PSI
4	<W* >	<U03A9>	GREEK CAPITAL LETTER OMEGA
4	<J* >	<U03AA>	GREEK CAPITAL LETTER IOTA WITH DIALYTIKA
4	<V* >	<U03AB>	GREEK CAPITAL LETTER UPSILON WITH DIALYTIKA
4	<a%>	<U03AC>	GREEK SMALL LETTER ALPHA WITH TONOS
4	<e%>	<U03AD>	GREEK SMALL LETTER EPSILON WITH TONOS
4	<y%>	<U03AE>	GREEK SMALL LETTER ETA WITH TONOS
4	<i%>	<U03AF>	GREEK SMALL LETTER IOTA WITH TONOS
4	<u3>	<U03B0>	GREEK SMALL LETTER UPSILON WITH DIALYTIKA AND TONOS
4	<a* >	<U03B1>	GREEK SMALL LETTER ALPHA
4	<b* >	<U03B2>	GREEK SMALL LETTER BETA
4	<g* >	<U03B3>	GREEK SMALL LETTER GAMMA

4209	<d*>	<U03B4>	GREEK SMALL LETTER DELTA
4210	<e*>	<U03B5>	GREEK SMALL LETTER EPSILON
4211	<z*>	<U03B6>	GREEK SMALL LETTER ZETA
4212	<y*>	<U03B7>	GREEK SMALL LETTER ETA
4213	<h*>	<U03B8>	GREEK SMALL LETTER THETA
4214	<i*>	<U03B9>	GREEK SMALL LETTER IOTA
4215	<k*>	<U03BA>	GREEK SMALL LETTER KAPPA
4216	<l*>	<U03BB>	GREEK SMALL LETTER LAMDA
4217	<m*>	<U03BC>	GREEK SMALL LETTER MU
4218	<n*>	<U03BD>	GREEK SMALL LETTER NU
4219	<c*>	<U03BE>	GREEK SMALL LETTER XI
4220	<o*>	<U03BF>	GREEK SMALL LETTER OMICRON
4221	<p*>	<U03C0>	GREEK SMALL LETTER PI
4222	<r*>	<U03C1>	GREEK SMALL LETTER RHO
4223	<s*>	<U03C2>	GREEK SMALL LETTER FINAL SIGMA
4224	<s*>	<U03C3>	GREEK SMALL LETTER SIGMA
4225	<t*>	<U03C4>	GREEK SMALL LETTER TAU
4226	<u*>	<U03C5>	GREEK SMALL LETTER UPSILON
4227	<f*>	<U03C6>	GREEK SMALL LETTER PHI
4228	<x*>	<U03C7>	GREEK SMALL LETTER CHI
4229	<q*>	<U03C8>	GREEK SMALL LETTER PSI
4230	<w*>	<U03C9>	GREEK SMALL LETTER OMEGA
4231	<j*>	<U03CA>	GREEK SMALL LETTER IOTA WITH DIALYTIKA
4232	<v*>	<U03CB>	GREEK SMALL LETTER UPSILON WITH DIALYTIKA
4233	<o%>	<U03CC>	GREEK SMALL LETTER OMICRON WITH TONOS
4234	<u%>	<U03CD>	GREEK SMALL LETTER UPSILON WITH TONOS
4235	<w%>	<U03CE>	GREEK SMALL LETTER OMEGA WITH TONOS
4236	<b3>	<U03D0>	GREEK BETA SYMBOL
4237	<T3>	<U03DA>	GREEK LETTER STIGMA
4238	<M3>	<U03DC>	GREEK LETTER DIGAMMA
4239	<K3>	<U03DE>	GREEK LETTER KOPPA
4240	<P3>	<U03E0>	GREEK LETTER SAMPI
4241	<IO>	<U0401>	CYRILLIC CAPITAL LETTER IO
4242	<D%>	<U0402>	CYRILLIC CAPITAL LETTER DJE (Serbocroatian)
4243	<G%>	<U0403>	CYRILLIC CAPITAL LETTER GJE
4244	<IE>	<U0404>	CYRILLIC CAPITAL LETTER UKRAINIAN IE
4245	<DS>	<U0405>	CYRILLIC CAPITAL LETTER DZE
4246	<II>	<U0406>	CYRILLIC CAPITAL LETTER BYELORUSSIAN-UKRAINIAN I
4247	<YI>	<U0407>	CYRILLIC CAPITAL LETTER YI (Ukrainian)
4248	<J%>	<U0408>	CYRILLIC CAPITAL LETTER JE
4249	<LJ>	<U0409>	CYRILLIC CAPITAL LETTER LJE
4250	<NJ>	<U040A>	CYRILLIC CAPITAL LETTER NJE
4251	<Ts>	<U040B>	CYRILLIC CAPITAL LETTER TSHE (Serbocroatian)
4252	<KJ>	<U040C>	CYRILLIC CAPITAL LETTER KJE
4253	<V%>	<U040E>	CYRILLIC CAPITAL LETTER SHORT U (Byelorussian)
4254	<DZ>	<U040F>	CYRILLIC CAPITAL LETTER DZHE
4255	<A=>	<U0410>	CYRILLIC CAPITAL LETTER A
4256	<B=>	<U0411>	CYRILLIC CAPITAL LETTER BE
4257	<V=>	<U0412>	CYRILLIC CAPITAL LETTER VE
4258	<G=>	<U0413>	CYRILLIC CAPITAL LETTER GHE
4259	<D=>	<U0414>	CYRILLIC CAPITAL LETTER DE
4260	<E=>	<U0415>	CYRILLIC CAPITAL LETTER IE
4261	<Z%>	<U0416>	CYRILLIC CAPITAL LETTER ZHE
4262	<Z=>	<U0417>	CYRILLIC CAPITAL LETTER ZE
4263	<I=>	<U0418>	CYRILLIC CAPITAL LETTER I
4264	<J=>	<U0419>	CYRILLIC CAPITAL LETTER SHORT I
4265	<K=>	<U041A>	CYRILLIC CAPITAL LETTER KA
4266	<L=>	<U041B>	CYRILLIC CAPITAL LETTER EL
4267	<M=>	<U041C>	CYRILLIC CAPITAL LETTER EM
4268	<N=>	<U041D>	CYRILLIC CAPITAL LETTER EN
4269	<O=>	<U041E>	CYRILLIC CAPITAL LETTER O
4270	<P=>	<U041F>	CYRILLIC CAPITAL LETTER PE
4271	<R=>	<U0420>	CYRILLIC CAPITAL LETTER ER
4272	<S=>	<U0421>	CYRILLIC CAPITAL LETTER ES
4273	<T=>	<U0422>	CYRILLIC CAPITAL LETTER TE
4274	<U=>	<U0423>	CYRILLIC CAPITAL LETTER U
4275	<F=>	<U0424>	CYRILLIC CAPITAL LETTER EF
4276	<H=>	<U0425>	CYRILLIC CAPITAL LETTER HA
4277	<C=>	<U0426>	CYRILLIC CAPITAL LETTER TSE
4278	<C%>	<U0427>	CYRILLIC CAPITAL LETTER CHE
4279	<S%>	<U0428>	CYRILLIC CAPITAL LETTER SHA
4280	<Sc>	<U0429>	CYRILLIC CAPITAL LETTER SHCHA
4281	<=">	<U042A>	CYRILLIC CAPITAL LETTER HARD SIGN
4282	<Y=>	<U042B>	CYRILLIC CAPITAL LETTER YERU
4283	<% ">	<U042C>	CYRILLIC CAPITAL LETTER SOFT SIGN
4284	<JE>	<U042D>	CYRILLIC CAPITAL LETTER E
4285	<JU>	<U042E>	CYRILLIC CAPITAL LETTER YU
4286	<JA>	<U042F>	CYRILLIC CAPITAL LETTER YA
4287	<a=>	<U0430>	CYRILLIC SMALL LETTER A
4288	<b=>	<U0431>	CYRILLIC SMALL LETTER BE
4289	<v=>	<U0432>	CYRILLIC SMALL LETTER VE
4290	<g=>	<U0433>	CYRILLIC SMALL LETTER GHE
4291	<d=>	<U0434>	CYRILLIC SMALL LETTER DE
4292	<e=>	<U0435>	CYRILLIC SMALL LETTER IE
4293	<z%>	<U0436>	CYRILLIC SMALL LETTER ZHE
4294	<z=>	<U0437>	CYRILLIC SMALL LETTER ZE
4295	<i=>	<U0438>	CYRILLIC SMALL LETTER I
4296	<j=>	<U0439>	CYRILLIC SMALL LETTER SHORT I

<k=>	<U043A>	CYRILLIC SMALL LETTER KA
<l=>	<U043B>	CYRILLIC SMALL LETTER EL
<m=>	<U043C>	CYRILLIC SMALL LETTER EM
<n=>	<U043D>	CYRILLIC SMALL LETTER EN
<o=>	<U043E>	CYRILLIC SMALL LETTER O
<p=>	<U043F>	CYRILLIC SMALL LETTER PE
<r=>	<U0440>	CYRILLIC SMALL LETTER ER
<s=>	<U0441>	CYRILLIC SMALL LETTER ES
<t=>	<U0442>	CYRILLIC SMALL LETTER TE
<u=>	<U0443>	CYRILLIC SMALL LETTER U
<f=>	<U0444>	CYRILLIC SMALL LETTER EF
<h=>	<U0445>	CYRILLIC SMALL LETTER HA
<c=>	<U0446>	CYRILLIC SMALL LETTER TSE
<c%>	<U0447>	CYRILLIC SMALL LETTER CHE
<s%>	<U0448>	CYRILLIC SMALL LETTER SHA
<sc>	<U0449>	CYRILLIC SMALL LETTER SHCHA
<='>	<U044A>	CYRILLIC SMALL LETTER HARD SIGN
<y=>	<U044B>	CYRILLIC SMALL LETTER YERU
<%'/>	<U044C>	CYRILLIC SMALL LETTER SOFT SIGN
<je>	<U044D>	CYRILLIC SMALL LETTER E
<ju>	<U044E>	CYRILLIC SMALL LETTER YU
<ja>	<U044F>	CYRILLIC SMALL LETTER YA
<io>	<U0451>	CYRILLIC SMALL LETTER IO
<d%>	<U0452>	CYRILLIC SMALL LETTER DJE (Serbocroatian)
<g%>	<U0453>	CYRILLIC SMALL LETTER GJE
<ie>	<U0454>	CYRILLIC SMALL LETTER UKRAINIAN IE
<ds>	<U0455>	CYRILLIC SMALL LETTER DZE
<ii>	<U0456>	CYRILLIC SMALL LETTER BYELORUSSIAN-UKRAINIAN I
<yi>	<U0457>	CYRILLIC SMALL LETTER YI (Ukrainian)
<j%>	<U0458>	CYRILLIC SMALL LETTER JE
<lj>	<U0459>	CYRILLIC SMALL LETTER LJJE
<nj>	<U045A>	CYRILLIC SMALL LETTER NJJE
<ts>	<U045B>	CYRILLIC SMALL LETTER TSHE (Serbocroatian)
<kj>	<U045C>	CYRILLIC SMALL LETTER KJE
<v%>	<U045E>	CYRILLIC SMALL LETTER SHORT U (Byelorussian)
<dz>	<U045F>	CYRILLIC SMALL LETTER DZHE
<Y3>	<U0462>	CYRILLIC CAPITAL LETTER YAT
<y3>	<U0463>	CYRILLIC SMALL LETTER YAT
<O3>	<U046A>	CYRILLIC CAPITAL LETTER BIG YUS
<o3>	<U046B>	CYRILLIC SMALL LETTER BIG YUS
<F3>	<U0472>	CYRILLIC CAPITAL LETTER FITA
<f3>	<U0473>	CYRILLIC SMALL LETTER FITA
<V3>	<U0474>	CYRILLIC CAPITAL LETTER IZHITSA
<v3>	<U0475>	CYRILLIC SMALL LETTER IZHITSA
<C3>	<U0480>	CYRILLIC CAPITAL LETTER KOPPA
<c3>	<U0481>	CYRILLIC SMALL LETTER KOPPA
<G3>	<U0490>	CYRILLIC CAPITAL LETTER GHE WITH UPTURN
<g3>	<U0491>	CYRILLIC SMALL LETTER GHE WITH UPTURN
<A+>	<U05D0>	HEBREW LETTER ALEF
<B+>	<U05D1>	HEBREW LETTER BET
<G+>	<U05D2>	HEBREW LETTER GIMEL
<D+>	<U05D3>	HEBREW LETTER DALET
<H+>	<U05D4>	HEBREW LETTER HE
<W+>	<U05D5>	HEBREW LETTER VAV
<Z+>	<U05D6>	HEBREW LETTER ZAYIN
<X+>	<U05D7>	HEBREW LETTER HET
<Tj>	<U05D8>	HEBREW LETTER TET
<J+>	<U05D9>	HEBREW LETTER YOD
<K%>	<U05DA>	HEBREW LETTER FINAL KAF
<K+>	<U05DB>	HEBREW LETTER KAF
<L+>	<U05DC>	HEBREW LETTER LAMED
<M%>	<U05DD>	HEBREW LETTER FINAL MEM
<M+>	<U05DE>	HEBREW LETTER MEM
<N%>	<U05DF>	HEBREW LETTER FINAL NUN
<N+>	<U05E0>	HEBREW LETTER NUN
<S+>	<U05E1>	HEBREW LETTER SAMEKH
<E+>	<U05E2>	HEBREW LETTER AYIN
<P%>	<U05E3>	HEBREW LETTER FINAL PE
<P+>	<U05E4>	HEBREW LETTER PE
<Zj>	<U05E5>	HEBREW LETTER FINAL TSADI
<ZJ>	<U05E6>	HEBREW LETTER TSADI
<Q+>	<U05E7>	HEBREW LETTER QOF
<R+>	<U05E8>	HEBREW LETTER RESH
<Sh>	<U05E9>	HEBREW LETTER SHIN
<T+>	<U05EA>	HEBREW LETTER TAV
<,+>	<U060C>	ARABIC COMMA
< ; +>	<U061B>	ARABIC SEMICOLON
< ? +>	<U061F>	ARABIC QUESTION MARK
<H'>	<U0621>	ARABIC LETTER HAMZA
<aM>	<U0622>	ARABIC LETTER ALEF WITH MADDA ABOVE
<aH>	<U0623>	ARABIC LETTER ALEF WITH HAMZA ABOVE
<wH>	<U0624>	ARABIC LETTER WAW WITH HAMZA ABOVE
<ah>	<U0625>	ARABIC LETTER ALEF WITH HAMZA BELOW
<yH>	<U0626>	ARABIC LETTER YEH WITH HAMZA ABOVE
<a+>	<U0627>	ARABIC LETTER ALEF
<b+>	<U0628>	ARABIC LETTER BEH
<tm>	<U0629>	ARABIC LETTER TEH MARBUTA
<t+>	<U062A>	ARABIC LETTER TEH

<tk>	<U062B>	ARABIC LETTER THEH
<g+>	<U062C>	ARABIC LETTER JEEM
<hk>	<U062D>	ARABIC LETTER HAH
<x+>	<U062E>	ARABIC LETTER KHAH
<d+>	<U062F>	ARABIC LETTER DAL
<dk>	<U0630>	ARABIC LETTER THAL
<r+>	<U0631>	ARABIC LETTER REH
<z+>	<U0632>	ARABIC LETTER ZAIN
<s+>	<U0633>	ARABIC LETTER SEEN
<sn>	<U0634>	ARABIC LETTER SHEEN
<c+>	<U0635>	ARABIC LETTER SAD
<dd>	<U0636>	ARABIC LETTER DAD
<tj>	<U0637>	ARABIC LETTER TAH
<zH>	<U0638>	ARABIC LETTER ZAH
<e+>	<U0639>	ARABIC LETTER AIN
<i+>	<U063A>	ARABIC LETTER GHAIN
<+>	<U0640>	ARABIC TATWEEL
<f+>	<U0641>	ARABIC LETTER FEH
<q+>	<U0642>	ARABIC LETTER QAF
<k+>	<U0643>	ARABIC LETTER KAF
<l+>	<U0644>	ARABIC LETTER LAM
<m+>	<U0645>	ARABIC LETTER MEEM
<n+>	<U0646>	ARABIC LETTER NOON
<h+>	<U0647>	ARABIC LETTER HEH
<w+>	<U0648>	ARABIC LETTER WAW
<j+>	<U0649>	ARABIC LETTER ALEF MAKSURA
<y+>	<U064A>	ARABIC LETTER YEH
<:+>	<U064B>	ARABIC FATHATAN
<"+>	<U064C>	ARABIC DAMMATAN
<=>	<U064D>	ARABIC KASRATAN
<//+>	<U064E>	ARABIC FATHA
<' +>	<U064F>	ARABIC DAMMA
<l+>	<U0650>	ARABIC KASRA
<3+>	<U0651>	ARABIC SHADDA
<0+>	<U0652>	ARABIC SUKUN
<0a>	<U0660>	ARABIC-INDIC DIGIT ZERO
<1a>	<U0661>	ARABIC-INDIC DIGIT ONE
<2a>	<U0662>	ARABIC-INDIC DIGIT TWO
<3a>	<U0663>	ARABIC-INDIC DIGIT THREE
<4a>	<U0664>	ARABIC-INDIC DIGIT FOUR
<5a>	<U0665>	ARABIC-INDIC DIGIT FIVE
<6a>	<U0666>	ARABIC-INDIC DIGIT SIX
<7a>	<U0667>	ARABIC-INDIC DIGIT SEVEN
<8a>	<U0668>	ARABIC-INDIC DIGIT EIGHT
<9a>	<U0669>	ARABIC-INDIC DIGIT NINE
<aS>	<U0670>	ARABIC LETTER SUPERSCRIPIT ALEF
<p+>	<U067E>	ARABIC LETTER PEH
<hH>	<U0681>	ARABIC LETTER HAH WITH HAMZA ABOVE
<tc>	<U0686>	ARABIC LETTER TCHEH
<zj>	<U0698>	ARABIC LETTER JEH
<v+>	<U06A4>	ARABIC LETTER VEH
<gf>	<U06AF>	ARABIC LETTER GAF
<A-0>	<U1E00>	LATIN CAPITAL LETTER A WITH RING BELOW
<a-0>	<U1E01>	LATIN SMALL LETTER A WITH RING BELOW
<B.>	<U1E02>	LATIN CAPITAL LETTER B WITH DOT ABOVE
<b.>	<U1E03>	LATIN SMALL LETTER B WITH DOT ABOVE
<B-.>	<U1E04>	LATIN CAPITAL LETTER B WITH DOT BELOW
<b-.>	<U1E05>	LATIN SMALL LETTER B WITH DOT BELOW
<B_>	<U1E06>	LATIN CAPITAL LETTER B WITH LINE BELOW
<b_>	<U1E07>	LATIN SMALL LETTER B WITH LINE BELOW
<C,'>	<U1E08>	LATIN CAPITAL LETTER C WITH CEDILLA AND ACUTE
<c,'>	<U1E09>	LATIN SMALL LETTER C WITH CEDILLA AND ACUTE
<D.>	<U1E0A>	LATIN CAPITAL LETTER D WITH DOT ABOVE
<d.>	<U1E0B>	LATIN SMALL LETTER D WITH DOT ABOVE
<D-.>	<U1E0C>	LATIN CAPITAL LETTER D WITH DOT BELOW
<d-.>	<U1E0D>	LATIN SMALL LETTER D WITH DOT BELOW
<D_>	<U1E0E>	LATIN CAPITAL LETTER D WITH LINE BELOW
<d_>	<U1E0F>	LATIN SMALL LETTER D WITH LINE BELOW
<D,>	<U1E10>	LATIN CAPITAL LETTER D WITH CEDILLA
<d,>	<U1E11>	LATIN SMALL LETTER D WITH CEDILLA
<D- />>	<U1E12>	LATIN CAPITAL LETTER D WITH CIRCUMFLEX BELOW
<d- />>	<U1E13>	LATIN SMALL LETTER D WITH CIRCUMFLEX BELOW
<E-!>	<U1E14>	LATIN CAPITAL LETTER E WITH MACRON AND GRAVE
<e-!>	<U1E15>	LATIN SMALL LETTER E WITH MACRON AND GRAVE
<E-'>	<U1E16>	LATIN CAPITAL LETTER E WITH MACRON AND ACUTE
<e-'>	<U1E17>	LATIN SMALL LETTER E WITH MACRON AND ACUTE
<E- />>	<U1E18>	LATIN CAPITAL LETTER E WITH CIRCUMFLEX BELOW
<e- />>	<U1E19>	LATIN SMALL LETTER E WITH CIRCUMFLEX BELOW
<E-?>	<U1E1A>	LATIN CAPITAL LETTER E WITH TILDE BELOW
<e-?>	<U1E1B>	LATIN SMALL LETTER E WITH TILDE BELOW
<E,(>	<U1E1C>	LATIN CAPITAL LETTER E WITH CEDILLA AND BREVE
<e,(>	<U1E1D>	LATIN SMALL LETTER E WITH CEDILLA AND BREVE
<F.>	<U1E1E>	LATIN CAPITAL LETTER F WITH DOT ABOVE
<f.>	<U1E1F>	LATIN SMALL LETTER F WITH DOT ABOVE
<G.>	<U1E20>	LATIN CAPITAL LETTER G WITH MACRON
<g.>	<U1E21>	LATIN SMALL LETTER G WITH MACRON
<H.>	<U1E22>	LATIN CAPITAL LETTER H WITH DOT ABOVE
<h.>	<U1E23>	LATIN SMALL LETTER H WITH DOT ABOVE

<H- .>	<U1E24>	LATIN CAPITAL LETTER H WITH DOT BELOW
<h- .>	<U1E25>	LATIN SMALL LETTER H WITH DOT BELOW
<H:>	<U1E26>	LATIN CAPITAL LETTER H WITH DIAERESIS
<h:>	<U1E27>	LATIN SMALL LETTER H WITH DIAERESIS
<H>	<U1E28>	LATIN CAPITAL LETTER H WITH CEDILLA
<h>	<U1E29>	LATIN SMALL LETTER H WITH CEDILLA
<H- (>	<U1E2A>	LATIN CAPITAL LETTER H WITH BREVE BELOW
<h- (>	<U1E2B>	LATIN SMALL LETTER H WITH BREVE BELOW
<I- ?>	<U1E2C>	LATIN CAPITAL LETTER I WITH TILDE BELOW
<i- ?>	<U1E2D>	LATIN SMALL LETTER I WITH TILDE BELOW
<I: ' >	<U1E2E>	LATIN CAPITAL LETTER I WITH DIAERESIS AND ACUTE
<i: ' >	<U1E2F>	LATIN SMALL LETTER I WITH DIAERESIS AND ACUTE
<K' >	<U1E30>	LATIN CAPITAL LETTER K WITH ACUTE
<k' >	<U1E31>	LATIN SMALL LETTER K WITH ACUTE
<K- .>	<U1E32>	LATIN CAPITAL LETTER K WITH DOT BELOW
<k- .>	<U1E33>	LATIN SMALL LETTER K WITH DOT BELOW
<K_ >	<U1E34>	LATIN CAPITAL LETTER K WITH LINE BELOW
<k_ >	<U1E35>	LATIN SMALL LETTER K WITH LINE BELOW
<L- .>	<U1E36>	LATIN CAPITAL LETTER L WITH DOT BELOW
<l- .>	<U1E37>	LATIN SMALL LETTER L WITH DOT BELOW
<L- - .>	<U1E38>	LATIN CAPITAL LETTER L WITH DOT BELOW AND MACRON
<l- - .>	<U1E39>	LATIN SMALL LETTER L WITH DOT BELOW AND MACRON
<L_ >	<U1E3A>	LATIN CAPITAL LETTER L WITH LINE BELOW
<l_ >	<U1E3B>	LATIN SMALL LETTER L WITH LINE BELOW
<L- / >	<U1E3C>	LATIN CAPITAL LETTER L WITH CIRCUMFLEX BELOW
<l- / >	<U1E3D>	LATIN SMALL LETTER L WITH CIRCUMFLEX BELOW
<M' >	<U1E3E>	LATIN CAPITAL LETTER M WITH ACUTE
<m' >	<U1E3F>	LATIN SMALL LETTER M WITH ACUTE
<M. >	<U1E40>	LATIN CAPITAL LETTER M WITH DOT ABOVE
<m. >	<U1E41>	LATIN SMALL LETTER M WITH DOT ABOVE
<M- .>	<U1E42>	LATIN CAPITAL LETTER M WITH DOT BELOW
<m- .>	<U1E43>	LATIN SMALL LETTER M WITH DOT BELOW
<N. >	<U1E44>	LATIN CAPITAL LETTER N WITH DOT ABOVE
<n. >	<U1E45>	LATIN SMALL LETTER N WITH DOT ABOVE
<N- .>	<U1E46>	LATIN CAPITAL LETTER N WITH DOT BELOW
<n- .>	<U1E47>	LATIN SMALL LETTER N WITH DOT BELOW
<N_ >	<U1E48>	LATIN CAPITAL LETTER N WITH LINE BELOW
<n_ >	<U1E49>	LATIN SMALL LETTER N WITH LINE BELOW
<N- / >	<U1E4A>	LATIN CAPITAL LETTER N WITH CIRCUMFLEX BELOW
<n- / >	<U1E4B>	LATIN SMALL LETTER N WITH CIRCUMFLEX BELOW
<O? ' >	<U1E4C>	LATIN CAPITAL LETTER O WITH TILDE AND ACUTE
<o? ' >	<U1E4D>	LATIN SMALL LETTER O WITH TILDE AND ACUTE
<O? : >	<U1E4E>	LATIN CAPITAL LETTER O WITH TILDE AND DIAERESIS
<o? : >	<U1E4F>	LATIN SMALL LETTER O WITH TILDE AND DIAERESIS
<O- ! >	<U1E50>	LATIN CAPITAL LETTER O WITH MACRON AND GRAVE
<o- ! >	<U1E51>	LATIN SMALL LETTER O WITH MACRON AND GRAVE
<O- ' >	<U1E52>	LATIN CAPITAL LETTER O WITH MACRON AND ACUTE
<o- ' >	<U1E53>	LATIN SMALL LETTER O WITH MACRON AND ACUTE
<P' >	<U1E54>	LATIN CAPITAL LETTER P WITH ACUTE
<p' >	<U1E55>	LATIN SMALL LETTER P WITH ACUTE
<P. >	<U1E56>	LATIN CAPITAL LETTER P WITH DOT ABOVE
<p. >	<U1E57>	LATIN SMALL LETTER P WITH DOT ABOVE
<R. >	<U1E58>	LATIN CAPITAL LETTER R WITH DOT ABOVE
<r. >	<U1E59>	LATIN SMALL LETTER R WITH DOT ABOVE
<R- .>	<U1E5A>	LATIN CAPITAL LETTER R WITH DOT BELOW
<r- .>	<U1E5B>	LATIN SMALL LETTER R WITH DOT BELOW
<R- - .>	<U1E5C>	LATIN CAPITAL LETTER R WITH DOT BELOW AND MACRON
<r- - .>	<U1E5D>	LATIN SMALL LETTER R WITH DOT BELOW AND MACRON
<R_ >	<U1E5E>	LATIN CAPITAL LETTER R WITH LINE BELOW
<r_ >	<U1E5F>	LATIN SMALL LETTER R WITH LINE BELOW
<S. >	<U1E60>	LATIN CAPITAL LETTER S WITH DOT ABOVE
<s. >	<U1E61>	LATIN SMALL LETTER S WITH DOT ABOVE
<S- .>	<U1E62>	LATIN CAPITAL LETTER S WITH DOT BELOW
<s- .>	<U1E63>	LATIN SMALL LETTER S WITH DOT BELOW
<S' . >	<U1E64>	LATIN CAPITAL LETTER S WITH ACUTE AND DOT ABOVE
<s' . >	<U1E65>	LATIN SMALL LETTER S WITH ACUTE AND DOT ABOVE
<S< . >	<U1E66>	LATIN CAPITAL LETTER S WITH CARON AND DOT ABOVE
<s< . >	<U1E67>	LATIN SMALL LETTER S WITH CARON AND DOT ABOVE
<S- - .>	<U1E68>	LATIN CAPITAL LETTER S WITH DOT BELOW AND DOT ABOVE
<s- - .>	<U1E69>	LATIN SMALL LETTER S WITH DOT BELOW AND DOT ABOVE
<T. >	<U1E6A>	LATIN CAPITAL LETTER T WITH DOT ABOVE
<t. >	<U1E6B>	LATIN SMALL LETTER T WITH DOT ABOVE
<T- .>	<U1E6C>	LATIN CAPITAL LETTER T WITH DOT BELOW
<t- .>	<U1E6D>	LATIN SMALL LETTER T WITH DOT BELOW
<T_ >	<U1E6E>	LATIN CAPITAL LETTER T WITH LINE BELOW
<t_ >	<U1E6F>	LATIN SMALL LETTER T WITH LINE BELOW
<T- / >	<U1E70>	LATIN CAPITAL LETTER T WITH CIRCUMFLEX BELOW
<t- / >	<U1E71>	LATIN SMALL LETTER T WITH CIRCUMFLEX BELOW
<U- : >	<U1E72>	LATIN CAPITAL LETTER U WITH DIAERESIS BELOW
<u- : >	<U1E73>	LATIN SMALL LETTER U WITH DIAERESIS BELOW
<U- ? >	<U1E74>	LATIN CAPITAL LETTER U WITH TILDE BELOW
<u- ? >	<U1E75>	LATIN SMALL LETTER U WITH TILDE BELOW
<U- / >	<U1E76>	LATIN CAPITAL LETTER U WITH CIRCUMFLEX BELOW
<u- / >	<U1E77>	LATIN SMALL LETTER U WITH CIRCUMFLEX BELOW
<U? ' >	<U1E78>	LATIN CAPITAL LETTER U WITH TILDE AND ACUTE
<u? ' >	<U1E79>	LATIN SMALL LETTER U WITH TILDE AND ACUTE
<U- : >	<U1E7A>	LATIN CAPITAL LETTER U WITH MACRON AND DIAERESIS
<u- : >	<U1E7B>	LATIN SMALL LETTER U WITH MACRON AND DIAERESIS

<V?>	<U1E7C>	LATIN CAPITAL LETTER V WITH TILDE
<v?>	<U1E7D>	LATIN SMALL LETTER V WITH TILDE
<V-.>	<U1E7E>	LATIN CAPITAL LETTER V WITH DOT BELOW
<v-.>	<U1E7F>	LATIN SMALL LETTER V WITH DOT BELOW
<W!>	<U1E80>	LATIN CAPITAL LETTER W WITH GRAVE
<w!>	<U1E81>	LATIN SMALL LETTER W WITH GRAVE
<W'>	<U1E82>	LATIN CAPITAL LETTER W WITH ACUTE
<w'>	<U1E83>	LATIN SMALL LETTER W WITH ACUTE
<W:>	<U1E84>	LATIN CAPITAL LETTER W WITH DIAERESIS
<w:>	<U1E85>	LATIN SMALL LETTER W WITH DIAERESIS
<W.>	<U1E86>	LATIN CAPITAL LETTER W WITH DOT ABOVE
<w.>	<U1E87>	LATIN SMALL LETTER W WITH DOT ABOVE
<W-.>	<U1E88>	LATIN CAPITAL LETTER W WITH DOT BELOW
<w-.>	<U1E89>	LATIN SMALL LETTER W WITH DOT BELOW
<X.>	<U1E8A>	LATIN CAPITAL LETTER X WITH DOT ABOVE
<x.>	<U1E8B>	LATIN SMALL LETTER X WITH DOT ABOVE
<X:>	<U1E8C>	LATIN CAPITAL LETTER X WITH DIAERESIS
<x:>	<U1E8D>	LATIN SMALL LETTER X WITH DIAERESIS
<Y.>	<U1E8E>	LATIN CAPITAL LETTER Y WITH DOT ABOVE
<y.>	<U1E8F>	LATIN SMALL LETTER Y WITH DOT ABOVE
<Z/>>	<U1E90>	LATIN CAPITAL LETTER Z WITH CIRCUMFLEX
<z/>>	<U1E91>	LATIN SMALL LETTER Z WITH CIRCUMFLEX
<Z-.>	<U1E92>	LATIN CAPITAL LETTER Z WITH DOT BELOW
<z-.>	<U1E93>	LATIN SMALL LETTER Z WITH DOT BELOW
<Z_>	<U1E94>	LATIN CAPITAL LETTER Z WITH LINE BELOW
<z_>	<U1E95>	LATIN SMALL LETTER Z WITH LINE BELOW
<A-.>	<U1EA0>	LATIN CAPITAL LETTER A WITH DOT BELOW
<a-.>	<U1EA1>	LATIN SMALL LETTER A WITH DOT BELOW
<A2>	<U1EA2>	LATIN CAPITAL LETTER A WITH HOOK ABOVE
<a2>	<U1EA3>	LATIN SMALL LETTER A WITH HOOK ABOVE
<A/>'>	<U1EA4>	LATIN CAPITAL LETTER A WITH CIRCUMFLEX AND ACUTE
<a/>'>	<U1EA5>	LATIN SMALL LETTER A WITH CIRCUMFLEX AND ACUTE
<A/>!>	<U1EA6>	LATIN CAPITAL LETTER A WITH CIRCUMFLEX AND GRAVE
<a/>!>	<U1EA7>	LATIN SMALL LETTER A WITH CIRCUMFLEX AND GRAVE
<A/>2>	<U1EA8>	LATIN CAPITAL LETTER A WITH CIRCUMFLEX AND HOOK ABOVE
<a/>2>	<U1EA9>	LATIN SMALL LETTER A WITH CIRCUMFLEX AND HOOK ABOVE
<A/>?>	<U1EAA>	LATIN CAPITAL LETTER A WITH CIRCUMFLEX AND TILDE
<a/>?>	<U1EAB>	LATIN SMALL LETTER A WITH CIRCUMFLEX AND TILDE
<A/>-.>	<U1EAC>	LATIN CAPITAL LETTER A WITH CIRCUMFLEX AND DOT BELOW
<a/>-.>	<U1EAD>	LATIN SMALL LETTER A WITH CIRCUMFLEX AND DOT BELOW
<A('>	<U1EAE>	LATIN CAPITAL LETTER A WITH BREVE AND ACUTE
<a('>	<U1EAF>	LATIN SMALL LETTER A WITH BREVE AND ACUTE
<A(!>	<U1EB0>	LATIN CAPITAL LETTER A WITH BREVE AND GRAVE
<a(!>	<U1EB1>	LATIN SMALL LETTER A WITH BREVE AND GRAVE
<A(2>	<U1EB2>	LATIN CAPITAL LETTER A WITH BREVE AND HOOK ABOVE
<a(2>	<U1EB3>	LATIN SMALL LETTER A WITH BREVE AND HOOK ABOVE
<A(?>	<U1EB4>	LATIN CAPITAL LETTER A WITH BREVE AND TILDE
<a(?>	<U1EB5>	LATIN SMALL LETTER A WITH BREVE AND TILDE
<A(-.>	<U1EB6>	LATIN CAPITAL LETTER A WITH BREVE AND DOT BELOW
<a(-.>	<U1EB7>	LATIN SMALL LETTER A WITH BREVE AND DOT BELOW
<E-.>	<U1EB8>	LATIN CAPITAL LETTER E WITH DOT BELOW
<e-.>	<U1EB9>	LATIN SMALL LETTER E WITH DOT BELOW
<E2>	<U1EBA>	LATIN CAPITAL LETTER E WITH HOOK ABOVE
<e2>	<U1EBB>	LATIN SMALL LETTER E WITH HOOK ABOVE
<E?>	<U1EBC>	LATIN CAPITAL LETTER E WITH TILDE
<e?>	<U1EBD>	LATIN SMALL LETTER E WITH TILDE
<E/>'>	<U1EBE>	LATIN CAPITAL LETTER E WITH CIRCUMFLEX AND ACUTE
<e/>'>	<U1EBF>	LATIN SMALL LETTER E WITH CIRCUMFLEX AND ACUTE
<E/>!>	<U1EC0>	LATIN CAPITAL LETTER E WITH CIRCUMFLEX AND GRAVE
<e/>!>	<U1EC1>	LATIN SMALL LETTER E WITH CIRCUMFLEX AND GRAVE
<E/>2>	<U1EC2>	LATIN CAPITAL LETTER E WITH CIRCUMFLEX AND HOOK ABOVE
<e/>2>	<U1EC3>	LATIN SMALL LETTER E WITH CIRCUMFLEX AND HOOK ABOVE
<E/>?>	<U1EC4>	LATIN CAPITAL LETTER E WITH CIRCUMFLEX AND TILDE
<e/>?>	<U1EC5>	LATIN SMALL LETTER E WITH CIRCUMFLEX AND TILDE
<E/>-.>	<U1EC6>	LATIN CAPITAL LETTER E WITH CIRCUMFLEX AND DOT BELOW
<e/>-.>	<U1EC7>	LATIN SMALL LETTER E WITH CIRCUMFLEX AND DOT BELOW
<I2>	<U1EC8>	LATIN CAPITAL LETTER I WITH HOOK ABOVE
<i2>	<U1EC9>	LATIN SMALL LETTER I WITH HOOK ABOVE
<I-.>	<U1ECA>	LATIN CAPITAL LETTER I WITH DOT BELOW
<i-.>	<U1ECB>	LATIN SMALL LETTER I WITH DOT BELOW
<O-.>	<U1ECC>	LATIN CAPITAL LETTER O WITH DOT BELOW
<o-.>	<U1ECD>	LATIN SMALL LETTER O WITH DOT BELOW
<O2>	<U1ECE>	LATIN CAPITAL LETTER O WITH HOOK ABOVE
<o2>	<U1ECF>	LATIN SMALL LETTER O WITH HOOK ABOVE
<O/>'>	<U1ED0>	LATIN CAPITAL LETTER O WITH CIRCUMFLEX AND ACUTE
<o/>'>	<U1ED1>	LATIN SMALL LETTER O WITH CIRCUMFLEX AND ACUTE
<O/>!>	<U1ED2>	LATIN CAPITAL LETTER O WITH CIRCUMFLEX AND GRAVE
<o/>!>	<U1ED3>	LATIN SMALL LETTER O WITH CIRCUMFLEX AND GRAVE
<O/>2>	<U1ED4>	LATIN CAPITAL LETTER O WITH CIRCUMFLEX AND HOOK ABOVE
<o/>2>	<U1ED5>	LATIN SMALL LETTER O WITH CIRCUMFLEX AND HOOK ABOVE
<O/>?>	<U1ED6>	LATIN CAPITAL LETTER O WITH CIRCUMFLEX AND TILDE
<o/>?>	<U1ED7>	LATIN SMALL LETTER O WITH CIRCUMFLEX AND TILDE
<O/>-.>	<U1ED8>	LATIN CAPITAL LETTER O WITH CIRCUMFLEX AND DOT BELOW
<o/>-.>	<U1ED9>	LATIN SMALL LETTER O WITH CIRCUMFLEX AND DOT BELOW
<O9'>	<U1EDA>	LATIN CAPITAL LETTER O WITH HORN AND ACUTE
<o9'>	<U1EDB>	LATIN SMALL LETTER O WITH HORN AND ACUTE
<O9!>	<U1EDC>	LATIN CAPITAL LETTER O WITH HORN AND GRAVE
<o9!>	<U1EDD>	LATIN SMALL LETTER O WITH HORN AND GRAVE

464	<O92>	<U1EDE>	LATIN CAPITAL LETTER O WITH HORN AND HOOK ABOVE
465	<o92>	<U1EDF>	LATIN SMALL LETTER O WITH HORN AND HOOK ABOVE
466	<O9?>	<U1EE0>	LATIN CAPITAL LETTER O WITH HORN AND TILDE
467	<o9?>	<U1EE1>	LATIN SMALL LETTER O WITH HORN AND TILDE
468	<O9-.>	<U1EE2>	LATIN CAPITAL LETTER O WITH HORN AND DOT BELOW
469	<o9-.>	<U1EE3>	LATIN SMALL LETTER O WITH HORN AND DOT BELOW
470	<U-.>	<U1EE4>	LATIN CAPITAL LETTER U WITH DOT BELOW
471	<u-.>	<U1EE5>	LATIN SMALL LETTER U WITH DOT BELOW
472	<U2>	<U1EE6>	LATIN CAPITAL LETTER U WITH HOOK ABOVE
473	<u2>	<U1EE7>	LATIN SMALL LETTER U WITH HOOK ABOVE
474	<U9'>	<U1EE8>	LATIN CAPITAL LETTER U WITH HORN AND ACUTE
475	<u9'>	<U1EE9>	LATIN SMALL LETTER U WITH HORN AND ACUTE
476	<U9!>	<U1EEA>	LATIN CAPITAL LETTER U WITH HORN AND GRAVE
477	<u9!>	<U1EEB>	LATIN SMALL LETTER U WITH HORN AND GRAVE
478	<U92>	<U1EEC>	LATIN CAPITAL LETTER U WITH HORN AND HOOK ABOVE
479	<u92>	<U1EED>	LATIN SMALL LETTER U WITH HORN AND HOOK ABOVE
480	<U9?>	<U1EEE>	LATIN CAPITAL LETTER U WITH HORN AND TILDE
481	<u9?>	<U1EEF>	LATIN SMALL LETTER U WITH HORN AND TILDE
482	<U9-.>	<U1EF0>	LATIN CAPITAL LETTER U WITH HORN AND DOT BELOW
483	<u9-.>	<U1EF1>	LATIN SMALL LETTER U WITH HORN AND DOT BELOW
484	<Y!>	<U1EF2>	LATIN CAPITAL LETTER Y WITH GRAVE
485	<y!>	<U1EF3>	LATIN SMALL LETTER Y WITH GRAVE
486	<Y-.>	<U1EF4>	LATIN CAPITAL LETTER Y WITH DOT BELOW
487	<y-.>	<U1EF5>	LATIN SMALL LETTER Y WITH DOT BELOW
488	<Y2>	<U1EF6>	LATIN CAPITAL LETTER Y WITH HOOK ABOVE
489	<y2>	<U1EF7>	LATIN SMALL LETTER Y WITH HOOK ABOVE
490	<Y?>	<U1EF8>	LATIN CAPITAL LETTER Y WITH TILDE
491	<y?>	<U1EF9>	LATIN SMALL LETTER Y WITH TILDE
492	<a*,>	<U1F00>	GREEK SMALL LETTER ALPHA WITH PSILI
493	<a*,>	<U1F01>	GREEK SMALL LETTER ALPHA WITH DASIA
494	<a*;!>	<U1F02>	GREEK SMALL LETTER ALPHA WITH PSILI AND VARIA
495	<a*;!>	<U1F03>	GREEK SMALL LETTER ALPHA WITH DASIA AND VARIA
496	<a*,'>	<U1F04>	GREEK SMALL LETTER ALPHA WITH PSILI AND OXIA
497	<a*,'>	<U1F05>	GREEK SMALL LETTER ALPHA WITH DASIA AND OXIA
498	<a*,>	<U1F06>	GREEK SMALL LETTER ALPHA WITH PSILI AND PERISPOMENI
499	<a*,>	<U1F07>	GREEK SMALL LETTER ALPHA WITH DASIA AND PERISPOMENI
500	<A*,>	<U1F08>	GREEK CAPITAL LETTER ALPHA WITH PSILI
501	<A*,>	<U1F09>	GREEK CAPITAL LETTER ALPHA WITH DASIA
502	<A*;!>	<U1F0A>	GREEK CAPITAL LETTER ALPHA WITH PSILI AND VARIA
503	<A*;!>	<U1F0B>	GREEK CAPITAL LETTER ALPHA WITH DASIA AND VARIA
504	<A*,'>	<U1F0C>	GREEK CAPITAL LETTER ALPHA WITH PSILI AND OXIA
505	<A*,'>	<U1F0D>	GREEK CAPITAL LETTER ALPHA WITH DASIA AND OXIA
506	<A*,>	<U1F0E>	GREEK CAPITAL LETTER ALPHA WITH PSILI AND PERISPOMENI
507	<A*,>	<U1F0F>	GREEK CAPITAL LETTER ALPHA WITH DASIA AND PERISPOMENI
508	<e*,>	<U1F10>	GREEK SMALL LETTER EPSILON WITH PSILI
509	<e*,>	<U1F11>	GREEK SMALL LETTER EPSILON WITH DASIA
510	<e*;!>	<U1F12>	GREEK SMALL LETTER EPSILON WITH PSILI AND VARIA
511	<e*;!>	<U1F13>	GREEK SMALL LETTER EPSILON WITH DASIA AND VARIA
512	<e*,'>	<U1F14>	GREEK SMALL LETTER EPSILON WITH PSILI AND OXIA
513	<e*,'>	<U1F15>	GREEK SMALL LETTER EPSILON WITH DASIA AND OXIA
514	<E*,>	<U1F18>	GREEK CAPITAL LETTER EPSILON WITH PSILI
515	<E*,>	<U1F19>	GREEK CAPITAL LETTER EPSILON WITH DASIA
516	<E*;!>	<U1F1A>	GREEK CAPITAL LETTER EPSILON WITH PSILI AND VARIA
517	<E*;!>	<U1F1B>	GREEK CAPITAL LETTER EPSILON WITH DASIA AND VARIA
518	<E*,'>	<U1F1C>	GREEK CAPITAL LETTER EPSILON WITH PSILI AND OXIA
519	<E*,'>	<U1F1D>	GREEK CAPITAL LETTER EPSILON WITH DASIA AND OXIA
520	<y*,>	<U1F20>	GREEK SMALL LETTER ETA WITH PSILI
521	<y*,>	<U1F21>	GREEK SMALL LETTER ETA WITH DASIA
522	<y*;!>	<U1F22>	GREEK SMALL LETTER ETA WITH PSILI AND VARIA
523	<y*;!>	<U1F23>	GREEK SMALL LETTER ETA WITH DASIA AND VARIA
524	<y*,'>	<U1F24>	GREEK SMALL LETTER ETA WITH PSILI AND OXIA
525	<y*,'>	<U1F25>	GREEK SMALL LETTER ETA WITH DASIA AND OXIA
526	<y*,>	<U1F26>	GREEK SMALL LETTER ETA WITH PSILI AND PERISPOMENI
527	<y*,>	<U1F27>	GREEK SMALL LETTER ETA WITH DASIA AND PERISPOMENI
528	<Y*,>	<U1F28>	GREEK CAPITAL LETTER ETA WITH PSILI
529	<Y*,>	<U1F29>	GREEK CAPITAL LETTER ETA WITH DASIA
530	<Y*;!>	<U1F2A>	GREEK CAPITAL LETTER ETA WITH PSILI AND VARIA
531	<Y*;!>	<U1F2B>	GREEK CAPITAL LETTER ETA WITH DASIA AND VARIA
532	<Y*,'>	<U1F2C>	GREEK CAPITAL LETTER ETA WITH PSILI AND OXIA
533	<Y*,'>	<U1F2D>	GREEK CAPITAL LETTER ETA WITH DASIA AND OXIA
534	<Y*,>	<U1F2E>	GREEK CAPITAL LETTER ETA WITH PSILI AND PERISPOMENI
535	<Y*,>	<U1F2F>	GREEK CAPITAL LETTER ETA WITH DASIA AND PERISPOMENI
536	<i*,>	<U1F30>	GREEK SMALL LETTER IOTA WITH PSILI
537	<i*,>	<U1F31>	GREEK SMALL LETTER IOTA WITH DASIA
538	<i*;!>	<U1F32>	GREEK SMALL LETTER IOTA WITH PSILI AND VARIA
539	<i*;!>	<U1F33>	GREEK SMALL LETTER IOTA WITH DASIA AND VARIA
540	<i*,'>	<U1F34>	GREEK SMALL LETTER IOTA WITH PSILI AND OXIA
541	<i*,'>	<U1F35>	GREEK SMALL LETTER IOTA WITH DASIA AND OXIA
542	<i*,>	<U1F36>	GREEK SMALL LETTER IOTA WITH PSILI AND PERISPOMENI
543	<i*,>	<U1F37>	GREEK SMALL LETTER IOTA WITH DASIA AND PERISPOMENI
544	<I*,>	<U1F38>	GREEK CAPITAL LETTER IOTA WITH PSILI
545	<I*,>	<U1F39>	GREEK CAPITAL LETTER IOTA WITH DASIA
546	<I*;!>	<U1F3A>	GREEK CAPITAL LETTER IOTA WITH PSILI AND VARIA
547	<I*;!>	<U1F3B>	GREEK CAPITAL LETTER IOTA WITH DASIA AND VARIA
548	<I*,'>	<U1F3C>	GREEK CAPITAL LETTER IOTA WITH PSILI AND OXIA
549	<I*,'>	<U1F3D>	GREEK CAPITAL LETTER IOTA WITH DASIA AND OXIA
550	<I*,>	<U1F3E>	GREEK CAPITAL LETTER IOTA WITH PSILI AND PERISPOMENI
551	<I*,>	<U1F3F>	GREEK CAPITAL LETTER IOTA WITH DASIA AND PERISPOMENI

4737	<o*,>	<U1F40>	GREEK SMALL LETTER OMICRON WITH PSILI
4738	<o*;!>	<U1F41>	GREEK SMALL LETTER OMICRON WITH DASIA
4739	<o*;!>	<U1F42>	GREEK SMALL LETTER OMICRON WITH PSILI AND VARIA
4740	<o*;!>	<U1F43>	GREEK SMALL LETTER OMICRON WITH DASIA AND VARIA
4741	<o*,'>	<U1F44>	GREEK SMALL LETTER OMICRON WITH PSILI AND OXIA
4742	<o*,'>	<U1F45>	GREEK SMALL LETTER OMICRON WITH DASIA AND OXIA
4743	<O*,>	<U1F48>	GREEK CAPITAL LETTER OMICRON WITH PSILI
4744	<O*;!>	<U1F49>	GREEK CAPITAL LETTER OMICRON WITH DASIA
4745	<O*;!>	<U1F4A>	GREEK CAPITAL LETTER OMICRON WITH PSILI AND VARIA
4746	<O*;!>	<U1F4B>	GREEK CAPITAL LETTER OMICRON WITH DASIA AND VARIA
4747	<O*,'>	<U1F4C>	GREEK CAPITAL LETTER OMICRON WITH PSILI AND OXIA
4748	<O*,'>	<U1F4D>	GREEK CAPITAL LETTER OMICRON WITH DASIA AND OXIA
4749	<u*,>	<U1F50>	GREEK SMALL LETTER UPSILON WITH PSILI
4750	<u*;!>	<U1F51>	GREEK SMALL LETTER UPSILON WITH DASIA
4751	<u*;!>	<U1F52>	GREEK SMALL LETTER UPSILON WITH PSILI AND VARIA
4752	<u*;!>	<U1F53>	GREEK SMALL LETTER UPSILON WITH DASIA AND VARIA
4753	<u*,'>	<U1F54>	GREEK SMALL LETTER UPSILON WITH PSILI AND OXIA
4754	<u*,'>	<U1F55>	GREEK SMALL LETTER UPSILON WITH DASIA AND OXIA
4755	<u*,'>	<U1F56>	GREEK SMALL LETTER UPSILON WITH PSILI AND PERISPOMENI
4756	<u*,'>	<U1F57>	GREEK SMALL LETTER UPSILON WITH DASIA AND PERISPOMENI
4757	<U*,>	<U1F59>	GREEK CAPITAL LETTER UPSILON WITH DASIA
4758	<U*;!>	<U1F5B>	GREEK CAPITAL LETTER UPSILON WITH DASIA AND VARIA
4759	<U*,'>	<U1F5D>	GREEK CAPITAL LETTER UPSILON WITH DASIA AND OXIA
4760	<U*,'>	<U1F5F>	GREEK CAPITAL LETTER UPSILON WITH DASIA AND PERISPOMENI
4761	<w*,>	<U1F60>	GREEK SMALL LETTER OMEGA WITH PSILI
4762	<w*;!>	<U1F61>	GREEK SMALL LETTER OMEGA WITH DASIA
4763	<w*;!>	<U1F62>	GREEK SMALL LETTER OMEGA WITH PSILI AND VARIA
4764	<w*;!>	<U1F63>	GREEK SMALL LETTER OMEGA WITH DASIA AND VARIA
4765	<w*,'>	<U1F64>	GREEK SMALL LETTER OMEGA WITH PSILI AND OXIA
4766	<w*,'>	<U1F65>	GREEK SMALL LETTER OMEGA WITH DASIA AND OXIA
4767	<w*,'>	<U1F66>	GREEK SMALL LETTER OMEGA WITH PSILI AND PERISPOMENI
4768	<w*,'>	<U1F67>	GREEK SMALL LETTER OMEGA WITH DASIA AND PERISPOMENI
4769	<W*,>	<U1F68>	GREEK CAPITAL LETTER OMEGA WITH PSILI
4770	<W*;!>	<U1F69>	GREEK CAPITAL LETTER OMEGA WITH DASIA
4771	<W*;!>	<U1F6A>	GREEK CAPITAL LETTER OMEGA WITH PSILI AND VARIA
4772	<W*;!>	<U1F6B>	GREEK CAPITAL LETTER OMEGA WITH DASIA AND VARIA
4773	<W*,'>	<U1F6C>	GREEK CAPITAL LETTER OMEGA WITH PSILI AND OXIA
4774	<W*,'>	<U1F6D>	GREEK CAPITAL LETTER OMEGA WITH DASIA AND OXIA
4775	<W*,'>	<U1F6E>	GREEK CAPITAL LETTER OMEGA WITH PSILI AND PERISPOMENI
4776	<W*,'>	<U1F6F>	GREEK CAPITAL LETTER OMEGA WITH DASIA AND PERISPOMENI
4777	<a*;!>	<U1F70>	GREEK SMALL LETTER ALPHA WITH VARIA
4778	<a*,'>	<U1F71>	GREEK SMALL LETTER ALPHA WITH OXIA
4779	<e*!>	<U1F72>	GREEK SMALL LETTER EPSILON WITH VARIA
4780	<e*,'>	<U1F73>	GREEK SMALL LETTER EPSILON WITH OXIA
4781	<y*!>	<U1F74>	GREEK SMALL LETTER ETA WITH VARIA
4782	<y*,'>	<U1F75>	GREEK SMALL LETTER ETA WITH OXIA
4783	<i*!>	<U1F76>	GREEK SMALL LETTER IOTA WITH VARIA
4784	<i*,'>	<U1F77>	GREEK SMALL LETTER IOTA WITH OXIA
4785	<o*!>	<U1F78>	GREEK SMALL LETTER OMICRON WITH VARIA
4786	<o*,'>	<U1F79>	GREEK SMALL LETTER OMICRON WITH OXIA
4787	<u*!>	<U1F7A>	GREEK SMALL LETTER UPSILON WITH VARIA
4788	<u*,'>	<U1F7B>	GREEK SMALL LETTER UPSILON WITH OXIA
4789	<w*!>	<U1F7C>	GREEK SMALL LETTER OMEGA WITH VARIA
4790	<w*,'>	<U1F7D>	GREEK SMALL LETTER OMEGA WITH OXIA
4791	<a*,j>	<U1F80>	GREEK SMALL LETTER ALPHA WITH PSILI AND YPOGEGRAMMENI
4792	<a*,j>	<U1F81>	GREEK SMALL LETTER ALPHA WITH DASIA AND YPOGEGRAMMENI
4793	<a*;!j>	<U1F82>	GREEK SMALL LETTER ALPHA WITH PSILI AND VARIA AND YPOGEGRAMMENI
4794	<a*;!j>	<U1F83>	GREEK SMALL LETTER ALPHA WITH DASIA AND VARIA AND YPOGEGRAMMENI
4795	<a*,'j>	<U1F84>	GREEK SMALL LETTER ALPHA WITH PSILI AND OXIA AND YPOGEGRAMMENI
4796	<a*,'j>	<U1F85>	GREEK SMALL LETTER ALPHA WITH DASIA AND OXIA AND YPOGEGRAMMENI
4797	<a*,'j>	<U1F86>	GREEK SMALL LETTER ALPHA WITH PSILI AND PERISPOMENI AND YPOGEGRAMMENI
4798	<a*,'j>	<U1F87>	GREEK SMALL LETTER ALPHA WITH DASIA AND PERISPOMENI AND YPOGEGRAMMENI
4799	<A*,J>	<U1F88>	GREEK CAPITAL LETTER ALPHA WITH PSILI AND PROSGEGRAMMENI
4800	<A*,J>	<U1F89>	GREEK CAPITAL LETTER ALPHA WITH DASIA AND PROSGEGRAMMENI
4801	<A*;!J>	<U1F8A>	GREEK CAPITAL LETTER ALPHA WITH PSILI AND VARIA AND PROSGEGRAMMENI
4802	<A*;!J>	<U1F8B>	GREEK CAPITAL LETTER ALPHA WITH DASIA AND VARIA AND PROSGEGRAMMENI
4803	<A*,'J>	<U1F8C>	GREEK CAPITAL LETTER ALPHA WITH PSILI AND OXIA AND PROSGEGRAMMENI
4804	<A*,'J>	<U1F8D>	GREEK CAPITAL LETTER ALPHA WITH DASIA AND OXIA AND PROSGEGRAMMENI
4805	<A*,'J>	<U1F8E>	GREEK CAPITAL LETTER ALPHA WITH PSILI AND PERISPOMENI AND PROSGEGRAMMENI
4806	<A*,'J>	<U1F8F>	GREEK CAPITAL LETTER ALPHA WITH DASIA AND PERISPOMENI AND PROSGEGRAMMENI
4807	<y*,j>	<U1F90>	GREEK SMALL LETTER ETA WITH PSILI AND YPOGEGRAMMENI
4808	<y*,j>	<U1F91>	GREEK SMALL LETTER ETA WITH DASIA AND YPOGEGRAMMENI
4809	<y*;!j>	<U1F92>	GREEK SMALL LETTER ETA WITH PSILI AND VARIA AND YPOGEGRAMMENI
4810	<y*;!j>	<U1F93>	GREEK SMALL LETTER ETA WITH DASIA AND VARIA AND YPOGEGRAMMENI
4811	<y*,'j>	<U1F94>	GREEK SMALL LETTER ETA WITH PSILI AND OXIA AND YPOGEGRAMMENI
4812	<y*,'j>	<U1F95>	GREEK SMALL LETTER ETA WITH DASIA AND OXIA AND YPOGEGRAMMENI
4813	<y*,'j>	<U1F96>	GREEK SMALL LETTER ETA WITH PSILI AND PERISPOMENI AND YPOGEGRAMMENI
4814	<y*,'j>	<U1F97>	GREEK SMALL LETTER ETA WITH DASIA AND PERISPOMENI AND YPOGEGRAMMENI
4815	<Y*,J>	<U1F98>	GREEK CAPITAL LETTER ETA WITH PSILI AND PROSGEGRAMMENI
4816	<Y*,J>	<U1F99>	GREEK CAPITAL LETTER ETA WITH DASIA AND PROSGEGRAMMENI
4817	<Y*;!J>	<U1F9A>	GREEK CAPITAL LETTER ETA WITH PSILI AND VARIA AND PROSGEGRAMMENI
4818	<Y*;!J>	<U1F9B>	GREEK CAPITAL LETTER ETA WITH DASIA AND VARIA AND PROSGEGRAMMENI
4819	<Y*,'J>	<U1F9C>	GREEK CAPITAL LETTER ETA WITH PSILI AND OXIA AND PROSGEGRAMMENI
4820	<Y*,'J>	<U1F9D>	GREEK CAPITAL LETTER ETA WITH DASIA AND OXIA AND PROSGEGRAMMENI
4821	<Y*,'J>	<U1F9E>	GREEK CAPITAL LETTER ETA WITH PSILI AND PERISPOMENI AND PROSGEGRAMMENI
4822	<Y*,'J>	<U1F9F>	GREEK CAPITAL LETTER ETA WITH DASIA AND PERISPOMENI AND PROSGEGRAMMENI
4823	<w*,j>	<U1FA0>	GREEK SMALL LETTER OMEGA WITH PSILI AND YPOGEGRAMMENI
4824	<w*,j>	<U1FA1>	GREEK SMALL LETTER OMEGA WITH DASIA AND YPOGEGRAMMENI

<w*,!j>	<U1FA2>	GREEK SMALL LETTER OMEGA WITH PSILI AND VARIA AND YPOGEGRAMMENI
<w*;!j>	<U1FA3>	GREEK SMALL LETTER OMEGA WITH DASIA AND VARIA AND YPOGEGRAMMENI
<w*,'j>	<U1FA4>	GREEK SMALL LETTER OMEGA WITH PSILI AND OXIA AND YPOGEGRAMMENI
<w*,'j>	<U1FA5>	GREEK SMALL LETTER OMEGA WITH DASIA AND OXIA AND YPOGEGRAMMENI
<w*,?j>	<U1FA6>	GREEK SMALL LETTER OMEGA WITH PSILI AND PERISPOMENI AND YPOGEGRAMMENI
<w*;!j>	<U1FA7>	GREEK SMALL LETTER OMEGA WITH DASIA AND PERISPOMENI AND YPOGEGRAMMENI
<W*,J>	<U1FA8>	GREEK CAPITAL LETTER OMEGA WITH PSILI AND PROSGEGRAMMENI
<W*;J>	<U1FA9>	GREEK CAPITAL LETTER OMEGA WITH DASIA AND PROSGEGRAMMENI
<W*;!J>	<U1FAA>	GREEK CAPITAL LETTER OMEGA WITH PSILI AND VARIA AND PROSGEGRAMMENI
<W*;!J>	<U1FAB>	GREEK CAPITAL LETTER OMEGA WITH DASIA AND VARIA AND PROSGEGRAMMENI
<W*,'J>	<U1FAC>	GREEK CAPITAL LETTER OMEGA WITH PSILI AND OXIA AND PROSGEGRAMMENI
<W*,'J>	<U1FAD>	GREEK CAPITAL LETTER OMEGA WITH DASIA AND OXIA AND PROSGEGRAMMENI
<W*,?J>	<U1FAE>	GREEK CAPITAL LETTER OMEGA WITH PSILI AND PERISPOMENI AND PROSGEGRAMMENI
<W*,?J>	<U1FAF>	GREEK CAPITAL LETTER OMEGA WITH DASIA AND PERISPOMENI AND PROSGEGRAMMENI
<a*>	<U1FB0>	GREEK SMALL LETTER ALPHA WITH VRACHY
<a*->	<U1FB1>	GREEK SMALL LETTER ALPHA WITH MACRON
<a*!j>	<U1FB2>	GREEK SMALL LETTER ALPHA WITH VARIA AND YPOGEGRAMMENI
<a*j>	<U1FB3>	GREEK SMALL LETTER ALPHA WITH YPOGEGRAMMENI
<a*j>	<U1FB4>	GREEK SMALL LETTER ALPHA WITH OXIA AND YPOGEGRAMMENI
<a*?>	<U1FB6>	GREEK SMALL LETTER ALPHA WITH PERISPOMENI
<a*?j>	<U1FB7>	GREEK SMALL LETTER ALPHA WITH PERISPOMENI AND YPOGEGRAMMENI
<A*>	<U1FB8>	GREEK CAPITAL LETTER ALPHA WITH VRACHY
<A*->	<U1FB9>	GREEK CAPITAL LETTER ALPHA WITH MACRON
<A*!>	<U1FBA>	GREEK CAPITAL LETTER ALPHA WITH VARIA
<A*'>	<U1FBB>	GREEK CAPITAL LETTER ALPHA WITH OXIA
<A*J>	<U1FBC>	GREEK CAPITAL LETTER ALPHA WITH PROSGEGRAMMENI
<)*>	<U1FBD>	GREEK KORONIS
<J3>	<U1FBE>	GREEK PROSGEGRAMMENI
<,,>	<U1FBF>	GREEK PSILI
<?*>	<U1FC0>	GREEK PERISPOMENI
<?:>	<U1FC1>	GREEK DIALYTIKA AND PERISPOMENI
<y*!j>	<U1FC2>	GREEK SMALL LETTER ETA WITH VARIA AND YPOGEGRAMMENI
<y*j>	<U1FC3>	GREEK SMALL LETTER ETA WITH YPOGEGRAMMENI
<y*,j>	<U1FC4>	GREEK SMALL LETTER ETA WITH OXIA AND YPOGEGRAMMENI
<y*?>	<U1FC6>	GREEK SMALL LETTER ETA WITH PERISPOMENI
<y*?j>	<U1FC7>	GREEK SMALL LETTER ETA WITH PERISPOMENI AND YPOGEGRAMMENI
<E*!!>	<U1FC8>	GREEK CAPITAL LETTER EPSILON WITH VARIA
<E*'>	<U1FC9>	GREEK CAPITAL LETTER EPSILON WITH OXIA
<Y*!>	<U1FCA>	GREEK CAPITAL LETTER ETA WITH VARIA
<Y*'>	<U1FCB>	GREEK CAPITAL LETTER ETA WITH OXIA
<Y*J>	<U1FCC>	GREEK CAPITAL LETTER ETA WITH PROSGEGRAMMENI
<,!>	<U1FCD>	GREEK PSILI AND VARIA
<,,'>	<U1FCE>	GREEK PSILI AND OXIA
<?,>	<U1FCF>	GREEK PSILI AND PERISPOMENI
<i*>	<U1FD0>	GREEK SMALL LETTER IOTA WITH VRACHY
<i*->	<U1FD1>	GREEK SMALL LETTER IOTA WITH MACRON
<i*!>	<U1FD2>	GREEK SMALL LETTER IOTA WITH DIALYTIKA AND VARIA
<i*,'>	<U1FD3>	GREEK SMALL LETTER IOTA WITH DIALYTIKA AND OXIA
<i*?>	<U1FD6>	GREEK SMALL LETTER IOTA WITH PERISPOMENI
<i*!?>	<U1FD7>	GREEK SMALL LETTER IOTA WITH DIALYTIKA AND PERISPOMENI
<I*>	<U1FD8>	GREEK CAPITAL LETTER IOTA WITH VRACHY
<I*->	<U1FD9>	GREEK CAPITAL LETTER IOTA WITH MACRON
<I*!>	<U1FDA>	GREEK CAPITAL LETTER IOTA WITH VARIA
<I*'>	<U1FDB>	GREEK CAPITAL LETTER IOTA WITH OXIA
<!>	<U1FDD>	GREEK DASIA AND VARIA
<!,'>	<U1FDE>	GREEK DASIA AND OXIA
<?;!>	<U1FDF>	GREEK DASIA AND PERISPOMENI
<u*>	<U1FE0>	GREEK SMALL LETTER UPSILON WITH VRACHY
<u*->	<U1FE1>	GREEK SMALL LETTER UPSILON WITH MACRON
<u*!>	<U1FE2>	GREEK SMALL LETTER UPSILON WITH DIALYTIKA AND VARIA
<u*,'>	<U1FE3>	GREEK SMALL LETTER UPSILON WITH DIALYTIKA AND OXIA
<r*,>	<U1FE4>	GREEK SMALL LETTER RHO WITH PSILI
<r*,j>	<U1FE5>	GREEK SMALL LETTER RHO WITH DASIA
<u*?>	<U1FE6>	GREEK SMALL LETTER UPSILON WITH PERISPOMENI
<u*!?>	<U1FE7>	GREEK SMALL LETTER UPSILON WITH DIALYTIKA AND PERISPOMENI
<U*>	<U1FE8>	GREEK CAPITAL LETTER UPSILON WITH VRACHY
<U*->	<U1FE9>	GREEK CAPITAL LETTER UPSILON WITH MACRON
<U*!>	<U1FEA>	GREEK CAPITAL LETTER UPSILON WITH VARIA
<U*'>	<U1FEB>	GREEK CAPITAL LETTER UPSILON WITH OXIA
<R*,>	<U1FEC>	GREEK CAPITAL LETTER RHO WITH DASIA
<!:>	<U1FED>	GREEK DIALYTIKA AND VARIA
<:,'>	<U1FEE>	GREEK DIALYTIKA AND OXIA
<!*>	<U1FEF>	GREEK VARIA
<w*!j>	<U1FF2>	GREEK SMALL LETTER OMEGA WITH VARIA AND YPOGEGRAMMENI
<w*j>	<U1FF3>	GREEK SMALL LETTER OMEGA WITH YPOGEGRAMMENI
<w*,'j>	<U1FF4>	GREEK SMALL LETTER OMEGA WITH OXIA AND YPOGEGRAMMENI
<w*?>	<U1FF6>	GREEK SMALL LETTER OMEGA WITH PERISPOMENI
<w*?j>	<U1FF7>	GREEK SMALL LETTER OMEGA WITH PERISPOMENI AND YPOGEGRAMMENI
<O*!>	<U1FF8>	GREEK CAPITAL LETTER OMICRON WITH VARIA
<O*'>	<U1FF9>	GREEK CAPITAL LETTER OMICRON WITH OXIA
<W*!>	<U1FFA>	GREEK CAPITAL LETTER OMEGA WITH VARIA
<W*'>	<U1FFB>	GREEK CAPITAL LETTER OMEGA WITH OXIA
<W*J>	<U1FFC>	GREEK CAPITAL LETTER OMEGA WITH PROSGEGRAMMENI
</*!>	<U1FFD>	GREEK OXIA
<!;>	<U1FFE>	GREEK DASIA
<1N>	<U2002>	EN SPACE
<1M>	<U2003>	EM SPACE
<3M>	<U2004>	THREE-PER-EM SPACE

4	<4M>	<U2005>	FOUR-PER-EM SPACE
4	<6M>	<U2006>	SIX-PER-EM SPACE
4	<LR>	<U200E>	LEFT-TO-RIGHT MARK
4	<RL>	<U200F>	RIGHT-TO-LEFT MARK
4	<1T>	<U2009>	THIN SPACE
4	<1H>	<U200A>	HAIR SPACE
4	<-1>	<U2010>	HYPHEN
4	<-N>	<U2013>	EN DASH
4	<-M>	<U2014>	EM DASH
4	<-3>	<U2015>	HORIZONTAL BAR
4	<!2>	<U2016>	DOUBLE VERTICAL LINE
4	<=2>	<U2017>	DOUBLE LOW LINE
4	<'6>	<U2018>	LEFT SINGLE QUOTATION MARK
4	<'9>	<U2019>	RIGHT SINGLE QUOTATION MARK
4	<.9>	<U201A>	SINGLE LOW-9 QUOTATION MARK
4	<9'>	<U201B>	SINGLE HIGH-REVERSED-9 QUOTATION MARK
4	<"6>	<U201C>	LEFT DOUBLE QUOTATION MARK
4	<"9>	<U201D>	RIGHT DOUBLE QUOTATION MARK
4	<:9>	<U201E>	DOUBLE LOW-9 QUOTATION MARK
4	<9">	<U201F>	DOUBLE HIGH-REVERSED-9 QUOTATION MARK
4	<//->	<U2020>	DAGGER
4	<//=>	<U2021>	DOUBLE DAGGER
4	<sb>	<U2022>	BULLET
4	<3b>	<U2023>	TRIANGULAR BULLET
4	<..>	<U2025>	TWO DOT LEADER
4	<.3>	<U2026>	HORIZONTAL ELLIPSIS
4	<.->	<U2027>	HYPHENATION POINT
4	<linesep>	<U2028>	LINE SEPARATOR
4	<parsep>	<U2029>	PARAGRAPH SEPARATOR
4	<%0>	<U2030>	PER MILLE SIGN
4	<1'>	<U2032>	PRIME
4	<2'>	<U2033>	DOUBLE PRIME
4	<3'>	<U2034>	TRIPLE PRIME
4	<1">	<U2035>	REVERSED PRIME
4	<2">	<U2036>	REVERSED DOUBLE PRIME
4	<3">	<U2037>	REVERSED TRIPLE PRIME
4	<Ca>	<U2038>	CARET
4	<<1>	<U2039>	SINGLE LEFT-POINTING ANGLE QUOTATION MARK
4	</>1>	<U203A>	SINGLE RIGHT-POINTING ANGLE QUOTATION MARK
4	<:X>	<U203B>	REFERENCE MARK
4	<!*2>	<U203C>	DOUBLE EXCLAMATION MARK
4	<' ->	<U203E>	OVERLINE
4	<-b>	<U2043>	HYPHEN BULLET
4	<//f>	<U2044>	FRACTION SLASH
4	<0s>	<U2070>	SUPERSCRIPIT ZERO
4	<4S>	<U2074>	SUPERSCRIPIT FOUR
4	<5S>	<U2075>	SUPERSCRIPIT FIVE
4	<6S>	<U2076>	SUPERSCRIPIT SIX
4	<7S>	<U2077>	SUPERSCRIPIT SEVEN
4	<8S>	<U2078>	SUPERSCRIPIT EIGHT
4	<9S>	<U2079>	SUPERSCRIPIT NINE
4	<+S>	<U207A>	SUPERSCRIPIT PLUS SIGN
4	<-S>	<U207B>	SUPERSCRIPIT MINUS
4	<=S>	<U207C>	SUPERSCRIPIT EQUALS SIGN
4	<(S>	<U207D>	SUPERSCRIPIT LEFT PARENTHESIS
4	<)S>	<U207E>	SUPERSCRIPIT RIGHT PARENTHESIS
4	<nS>	<U207F>	SUPERSCRIPIT LATIN SMALL LETTER N
4	<0s>	<U2080>	SUBSCRIPT ZERO
4	<1s>	<U2081>	SUBSCRIPT ONE
4	<2s>	<U2082>	SUBSCRIPT TWO
4	<3s>	<U2083>	SUBSCRIPT THREE
4	<4s>	<U2084>	SUBSCRIPT FOUR
4	<5s>	<U2085>	SUBSCRIPT FIVE
4	<6s>	<U2086>	SUBSCRIPT SIX
4	<7s>	<U2087>	SUBSCRIPT SEVEN
4	<8s>	<U2088>	SUBSCRIPT EIGHT
4	<9s>	<U2089>	SUBSCRIPT NINE
4	<+s>	<U208A>	SUBSCRIPT PLUS SIGN
4	<-s>	<U208B>	SUBSCRIPT MINUS
4	<=s>	<U208C>	SUBSCRIPT EQUALS SIGN
4	<(s>	<U208D>	SUBSCRIPT LEFT PARENTHESIS
4	<)s>	<U208E>	SUBSCRIPT RIGHT PARENTHESIS
4	<Ff>	<U20A3>	FRENCH FRANC SIGN
4		<U20A4>	LIRA SIGN
4	<Pt>	<U20A7>	PESETA SIGN
4	<W>	<U20A9>	WON SIGN
4	<"7>	<U20D1>	COMBINING RIGHT HARPOON ABOVE
4	<oC>	<U2103>	DEGREE CELSIUS
4	<co>	<U2105>	CARE OF
4	<of>	<U2109>	DEGREE FAHRENHEIT
4	<N0>	<U2116>	NUMERO SIGN
4	<PO>	<U2117>	SOUND RECORDING COPYRIGHT
4	<Rx>	<U211E>	PRESCRIPTION TAKE
4	<SM>	<U2120>	SERVICE MARK
4	<TM>	<U2122>	TRADE MARK SIGN
4	<Om>	<U2126>	OHM SIGN
4	<AO>	<U212B>	ANGSTROM SIGN
4	<Est>	<U212E>	ESTIMATED SYMBOL

<13>	<U2153>	VULGAR FRACTION ONE THIRD
<23>	<U2154>	VULGAR FRACTION TWO THIRDS
<15>	<U2155>	VULGAR FRACTION ONE FIFTH
<25>	<U2156>	VULGAR FRACTION TWO FIFTHS
<35>	<U2157>	VULGAR FRACTION THREE FIFTHS
<45>	<U2158>	VULGAR FRACTION FOUR FIFTHS
<16>	<U2159>	VULGAR FRACTION ONE SIXTH
<56>	<U215A>	VULGAR FRACTION FIVE SIXTHS
<18>	<U215B>	VULGAR FRACTION ONE EIGHTH
<38>	<U215C>	VULGAR FRACTION THREE EIGHTHS
<58>	<U215D>	VULGAR FRACTION FIVE EIGHTHS
<78>	<U215E>	VULGAR FRACTION SEVEN EIGHTHS
<1R>	<U2160>	ROMAN NUMERAL ONE
<2R>	<U2161>	ROMAN NUMERAL TWO
<3R>	<U2162>	ROMAN NUMERAL THREE
<4R>	<U2163>	ROMAN NUMERAL FOUR
<5R>	<U2164>	ROMAN NUMERAL FIVE
<6R>	<U2165>	ROMAN NUMERAL SIX
<7R>	<U2166>	ROMAN NUMERAL SEVEN
<8R>	<U2167>	ROMAN NUMERAL EIGHT
<9R>	<U2168>	ROMAN NUMERAL NINE
<aR>	<U2169>	ROMAN NUMERAL TEN
 	<U216A>	ROMAN NUMERAL ELEVEN
<cr>	<U216B>	ROMAN NUMERAL TWELVE
<50R>	<U216C>	ROMAN NUMERAL FIFTY
<100R>	<U216D>	ROMAN NUMERAL ONE HUNDRED
<500R>	<U216E>	ROMAN NUMERAL FIVE HUNDRED
<1000R>	<U216F>	ROMAN NUMERAL ONE THOUSAND
<1r>	<U2170>	SMALL ROMAN NUMERAL ONE
<2r>	<U2171>	SMALL ROMAN NUMERAL TWO
<3r>	<U2172>	SMALL ROMAN NUMERAL THREE
<4r>	<U2173>	SMALL ROMAN NUMERAL FOUR
<5r>	<U2174>	SMALL ROMAN NUMERAL FIVE
<6r>	<U2175>	SMALL ROMAN NUMERAL SIX
<7r>	<U2176>	SMALL ROMAN NUMERAL SEVEN
<8r>	<U2177>	SMALL ROMAN NUMERAL EIGHT
<9r>	<U2178>	SMALL ROMAN NUMERAL NINE
<ar>	<U2179>	SMALL ROMAN NUMERAL TEN
 	<U217A>	SMALL ROMAN NUMERAL ELEVEN
<cr>	<U217B>	SMALL ROMAN NUMERAL TWELVE
<50r>	<U217C>	SMALL ROMAN NUMERAL FIFTY
<100r>	<U217D>	SMALL ROMAN NUMERAL ONE HUNDRED
<500r>	<U217E>	SMALL ROMAN NUMERAL FIVE HUNDRED
<1000r>	<U217F>	SMALL ROMAN NUMERAL ONE THOUSAND
<1000RCD>	<U2180>	ROMAN NUMERAL ONE THOUSAND C D
<5000R>	<U2181>	ROMAN NUMERAL FIVE THOUSAND
<10000R>	<U2182>	ROMAN NUMERAL TEN THOUSAND
<<->	<U2190>	LEFTWARDS ARROW
<-!>	<U2191>	UPWARDS ARROW
<-/>>	<U2192>	RIGHTWARDS ARROW
<-v>	<U2193>	DOWNWARDS ARROW
<</>>	<U2194>	LEFT RIGHT ARROW
<UD>	<U2195>	UP DOWN ARROW
<<!>	<U2196>	NORTH WEST ARROW
</////>>	<U2197>	NORTH EAST ARROW
<!//>>	<U2198>	SOUTH EAST ARROW
<</////>	<U2199>	SOUTH WEST ARROW
<UD->	<U21A8>	UP DOWN ARROW WITH BASE
</>V>	<U21C0>	RIGHTWARDS HARPOON WITH BARB UPWARDS
<<=>	<U21D0>	LEFTWARDS DOUBLE ARROW
<=>	<U21D2>	RIGHTWARDS DOUBLE ARROW
<==>	<U21D4>	LEFT RIGHT DOUBLE ARROW
<FA>	<U2200>	FOR ALL
<dP>	<U2202>	PARTIAL DIFFERENTIAL
<TE>	<U2203>	THERE EXISTS
<//0>	<U2205>	EMPTY SET
<DE>	<U2206>	INCREMENT
<NB>	<U2207>	NABLA
<(->	<U2208>	ELEMENT OF
<-)>	<U220B>	CONTAINS AS MEMBER
<FP>	<U220E>	END OF PROOF
<*P>	<U220F>	N-ARY PRODUCT
<+Z>	<U2211>	N-ARY SUMMATION
<-2>	<U2212>	MINUS SIGN
<-+>	<U2213>	MINUS-OR-PLUS SIGN
<. +>	<U2214>	DOT PLUS
<*->	<U2217>	ASTERISK OPERATOR
<Ob>	<U2218>	RING OPERATOR
<Sb>	<U2219>	BULLET OPERATOR
<RT>	<U221A>	SQUARE ROOT
<0(>	<U221D>	PROPORTIONAL TO
<00>	<U221E>	INFINITY
<-L>	<U221F>	RIGHT ANGLE
<-V>	<U2220>	ANGLE
<PP>	<U2225>	PARALLEL TO
<AN>	<U2227>	LOGICAL AND
<OR>	<U2228>	LOGICAL OR
<(U>	<U2229>	INTERSECTION

<)U>	<U222A>	UNION
<In>	<U222B>	INTEGRAL
<DI>	<U222C>	DOUBLE INTEGRAL
<Io>	<U222E>	CONTOUR INTEGRAL
<. : >	<U2234>	THEREFORE
< : . >	<U2235>	BECAUSE
< : R >	<U2236>	RATIO
< : : >	<U2237>	PROPORTION
<?1>	<U223C>	TILDE OPERATOR
<CG>	<U223E>	INVERTED LAZY S
<?->	<U2243>	ASYMPTOTICALLY EQUAL TO
<?=>	<U2245>	APPROXIMATELY EQUAL TO
<?2>	<U2248>	ALMOST EQUAL TO
<=?>	<U224C>	ALL EQUAL TO
<HI>	<U2253>	IMAGE OF OR APPROXIMATELY EQUAL TO
<! =>	<U2260>	NOT EQUAL TO
<=3>	<U2261>	IDENTICAL TO
<=<>	<U2264>	LESS-THAN OR EQUAL TO
</>=>	<U2265>	GREATER-THAN OR EQUAL TO
<<*>	<U226A>	MUCH LESS-THAN
<*/>>	<U226B>	MUCH GREATER-THAN
<!<>	<U226E>	NOT LESS-THAN
<!/>>	<U226F>	NOT GREATER-THAN
<(C>	<U2282>	SUBSET OF
<)C>	<U2283>	SUPERSET OF
<(_ >	<U2286>	SUBSET OF OR EQUAL TO
<)_>	<U2287>	SUPERSET OF OR EQUAL TO
<0.>	<U2299>	CIRCLED DOT OPERATOR
<02>	<U229A>	CIRCLED RING OPERATOR
<-T>	<U22A5>	UP TACK
<.P>	<U22C5>	DOT OPERATOR
<:3>	<U22EE>	VERTICAL ELLIPSIS
<Eh>	<U2302>	HOUSE
<<7>	<U2308>	LEFT CEILING
</>7>	<U2309>	RIGHT CEILING
<7<>	<U230A>	LEFT FLOOR
<7/>>	<U230B>	RIGHT FLOOR
<NI>	<U2310>	REVERSED NOT SIGN
<(A>	<U2312>	ARC
<TR>	<U2315>	TELEPHONE RECORDER
<88>	<U2318>	PLACE OF INTEREST SIGN
<Iu>	<U2320>	TOP HALF INTEGRAL
<I1>	<U2321>	BOTTOM HALF INTEGRAL
<<//>	<U2329>	LEFT-POINTING ANGLE BRACKET
<///>>	<U232A>	RIGHT-POINTING ANGLE BRACKET
<Vs>	<U2423>	OPEN BOX
<1h>	<U2440>	OCR HOOK
<3h>	<U2441>	OCR CHAIR
<2h>	<U2442>	OCR FORK
<4h>	<U2443>	OCR INVERTED FORK
<1j>	<U2446>	OCR BRANCH BANK IDENTIFICATION
<2j>	<U2447>	OCR AMOUNT OF CHECK
<3j>	<U2448>	OCR DASH
<4j>	<U2449>	OCR CUSTOMER ACCOUNT NUMBER
<1-o>	<U2460>	CIRCLED DIGIT ONE
<2-o>	<U2461>	CIRCLED DIGIT TWO
<3-o>	<U2462>	CIRCLED DIGIT THREE
<4-o>	<U2463>	CIRCLED DIGIT FOUR
<5-o>	<U2464>	CIRCLED DIGIT FIVE
<6-o>	<U2465>	CIRCLED DIGIT SIX
<7-o>	<U2466>	CIRCLED DIGIT SEVEN
<8-o>	<U2467>	CIRCLED DIGIT EIGHT
<9-o>	<U2468>	CIRCLED DIGIT NINE
<10-o>	<U2469>	CIRCLED NUMBER TEN
<11-o>	<U246A>	CIRCLED NUMBER ELEVEN
<12-o>	<U246B>	CIRCLED NUMBER TWELVE
<13-o>	<U246C>	CIRCLED NUMBER THIRTEEN
<14-o>	<U246D>	CIRCLED NUMBER FOURTEEN
<15-o>	<U246E>	CIRCLED NUMBER FIFTEEN
<16-o>	<U246F>	CIRCLED NUMBER SIXTEEN
<17-o>	<U2470>	CIRCLED NUMBER SEVENTEEN
<18-o>	<U2471>	CIRCLED NUMBER EIGHTEEN
<19-o>	<U2472>	CIRCLED NUMBER NINETEEN
<20-o>	<U2473>	CIRCLED NUMBER TWENTY
<(1)>	<U2474>	PARENTHESESIZED DIGIT ONE
<(2)>	<U2475>	PARENTHESESIZED DIGIT TWO
<(3)>	<U2476>	PARENTHESESIZED DIGIT THREE
<(4)>	<U2477>	PARENTHESESIZED DIGIT FOUR
<(5)>	<U2478>	PARENTHESESIZED DIGIT FIVE
<(6)>	<U2479>	PARENTHESESIZED DIGIT SIX
<(7)>	<U247A>	PARENTHESESIZED DIGIT SEVEN
<(8)>	<U247B>	PARENTHESESIZED DIGIT EIGHT
<(9)>	<U247C>	PARENTHESESIZED DIGIT NINE
<(10)>	<U247D>	PARENTHESESIZED NUMBER TEN
<(11)>	<U247E>	PARENTHESESIZED NUMBER ELEVEN
<(12)>	<U247F>	PARENTHESESIZED NUMBER TWELVE
<(13)>	<U2480>	PARENTHESESIZED NUMBER THIRTEEN
<(14)>	<U2481>	PARENTHESESIZED NUMBER FOURTEEN

<(15)>	<U2482>	PARENTHESIZED NUMBER FIFTEEN
<(16)>	<U2483>	PARENTHESIZED NUMBER SIXTEEN
<(17)>	<U2484>	PARENTHESIZED NUMBER SEVENTEEN
<(18)>	<U2485>	PARENTHESIZED NUMBER EIGHTEEN
<(19)>	<U2486>	PARENTHESIZED NUMBER NINETEEN
<(20)>	<U2487>	PARENTHESIZED NUMBER TWENTY
<1.>	<U2488>	DIGIT ONE FULL STOP
<2.>	<U2489>	DIGIT TWO FULL STOP
<3.>	<U248A>	DIGIT THREE FULL STOP
<4.>	<U248B>	DIGIT FOUR FULL STOP
<5.>	<U248C>	DIGIT FIVE FULL STOP
<6.>	<U248D>	DIGIT SIX FULL STOP
<7.>	<U248E>	DIGIT SEVEN FULL STOP
<8.>	<U248F>	DIGIT EIGHT FULL STOP
<9.>	<U2490>	DIGIT NINE FULL STOP
<10.>	<U2491>	NUMBER TEN FULL STOP
<11.>	<U2492>	NUMBER ELEVEN FULL STOP
<12.>	<U2493>	NUMBER TWELVE FULL STOP
<13.>	<U2494>	NUMBER THIRTEEN FULL STOP
<14.>	<U2495>	NUMBER FOURTEEN FULL STOP
<15.>	<U2496>	NUMBER FIFTEEN FULL STOP
<16.>	<U2497>	NUMBER SIXTEEN FULL STOP
<17.>	<U2498>	NUMBER SEVENTEEN FULL STOP
<18.>	<U2499>	NUMBER EIGHTEEN FULL STOP
<19.>	<U249A>	NUMBER NINETEEN FULL STOP
<20.>	<U249B>	NUMBER TWENTY FULL STOP
<(a)>	<U249C>	PARENTHESIZED LATIN SMALL LETTER A
<(b)>	<U249D>	PARENTHESIZED LATIN SMALL LETTER B
<(c)>	<U249E>	PARENTHESIZED LATIN SMALL LETTER C
<(d)>	<U249F>	PARENTHESIZED LATIN SMALL LETTER D
<(e)>	<U24A0>	PARENTHESIZED LATIN SMALL LETTER E
<(f)>	<U24A1>	PARENTHESIZED LATIN SMALL LETTER F
<(g)>	<U24A2>	PARENTHESIZED LATIN SMALL LETTER G
<(h)>	<U24A3>	PARENTHESIZED LATIN SMALL LETTER H
<(i)>	<U24A4>	PARENTHESIZED LATIN SMALL LETTER I
<(j)>	<U24A5>	PARENTHESIZED LATIN SMALL LETTER J
<(k)>	<U24A6>	PARENTHESIZED LATIN SMALL LETTER K
<(l)>	<U24A7>	PARENTHESIZED LATIN SMALL LETTER L
<(m)>	<U24A8>	PARENTHESIZED LATIN SMALL LETTER M
<(n)>	<U24A9>	PARENTHESIZED LATIN SMALL LETTER N
<(o)>	<U24AA>	PARENTHESIZED LATIN SMALL LETTER O
<(p)>	<U24AB>	PARENTHESIZED LATIN SMALL LETTER P
<(q)>	<U24AC>	PARENTHESIZED LATIN SMALL LETTER Q
<(r)>	<U24AD>	PARENTHESIZED LATIN SMALL LETTER R
<(s)>	<U24AE>	PARENTHESIZED LATIN SMALL LETTER S
<(t)>	<U24AF>	PARENTHESIZED LATIN SMALL LETTER T
<(u)>	<U24B0>	PARENTHESIZED LATIN SMALL LETTER U
<(v)>	<U24B1>	PARENTHESIZED LATIN SMALL LETTER V
<(w)>	<U24B2>	PARENTHESIZED LATIN SMALL LETTER W
<(x)>	<U24B3>	PARENTHESIZED LATIN SMALL LETTER X
<(y)>	<U24B4>	PARENTHESIZED LATIN SMALL LETTER Y
<(z)>	<U24B5>	PARENTHESIZED LATIN SMALL LETTER Z
<A-o>	<U24B6>	CIRCLED LATIN CAPITAL LETTER A
<B-o>	<U24B7>	CIRCLED LATIN CAPITAL LETTER B
<C-o>	<U24B8>	CIRCLED LATIN CAPITAL LETTER C
<D-o>	<U24B9>	CIRCLED LATIN CAPITAL LETTER D
<E-o>	<U24BA>	CIRCLED LATIN CAPITAL LETTER E
<F-o>	<U24BB>	CIRCLED LATIN CAPITAL LETTER F
<G-o>	<U24BC>	CIRCLED LATIN CAPITAL LETTER G
<H-o>	<U24BD>	CIRCLED LATIN CAPITAL LETTER H
<I-o>	<U24BE>	CIRCLED LATIN CAPITAL LETTER I
<J-o>	<U24BF>	CIRCLED LATIN CAPITAL LETTER J
<K-o>	<U24C0>	CIRCLED LATIN CAPITAL LETTER K
<L-o>	<U24C1>	CIRCLED LATIN CAPITAL LETTER L
<M-o>	<U24C2>	CIRCLED LATIN CAPITAL LETTER M
<N-o>	<U24C3>	CIRCLED LATIN CAPITAL LETTER N
<O-o>	<U24C4>	CIRCLED LATIN CAPITAL LETTER O
<P-o>	<U24C5>	CIRCLED LATIN CAPITAL LETTER P
<Q-o>	<U24C6>	CIRCLED LATIN CAPITAL LETTER Q
<R-o>	<U24C7>	CIRCLED LATIN CAPITAL LETTER R
<S-o>	<U24C8>	CIRCLED LATIN CAPITAL LETTER S
<T-o>	<U24C9>	CIRCLED LATIN CAPITAL LETTER T
<U-o>	<U24CA>	CIRCLED LATIN CAPITAL LETTER U
<V-o>	<U24CB>	CIRCLED LATIN CAPITAL LETTER V
<W-o>	<U24CC>	CIRCLED LATIN CAPITAL LETTER W
<X-o>	<U24CD>	CIRCLED LATIN CAPITAL LETTER X
<Y-o>	<U24CE>	CIRCLED LATIN CAPITAL LETTER Y
<Z-o>	<U24CF>	CIRCLED LATIN CAPITAL LETTER Z
<a-o>	<U24D0>	CIRCLED LATIN SMALL LETTER A
<b-o>	<U24D1>	CIRCLED LATIN SMALL LETTER B
<c-o>	<U24D2>	CIRCLED LATIN SMALL LETTER C
<d-o>	<U24D3>	CIRCLED LATIN SMALL LETTER D
<e-o>	<U24D4>	CIRCLED LATIN SMALL LETTER E
<f-o>	<U24D5>	CIRCLED LATIN SMALL LETTER F
<g-o>	<U24D6>	CIRCLED LATIN SMALL LETTER G
<h-o>	<U24D7>	CIRCLED LATIN SMALL LETTER H
<i-o>	<U24D8>	CIRCLED LATIN SMALL LETTER I
<j-o>	<U24D9>	CIRCLED LATIN SMALL LETTER J

<k-o>	<U24DA>	CIRCLED LATIN SMALL LETTER K
<l-o>	<U24DB>	CIRCLED LATIN SMALL LETTER L
<m-o>	<U24DC>	CIRCLED LATIN SMALL LETTER M
<n-o>	<U24DD>	CIRCLED LATIN SMALL LETTER N
<o-o>	<U24DE>	CIRCLED LATIN SMALL LETTER O
<p-o>	<U24DF>	CIRCLED LATIN SMALL LETTER P
<q-o>	<U24E0>	CIRCLED LATIN SMALL LETTER Q
<r-o>	<U24E1>	CIRCLED LATIN SMALL LETTER R
<s-o>	<U24E2>	CIRCLED LATIN SMALL LETTER S
<t-o>	<U24E3>	CIRCLED LATIN SMALL LETTER T
<u-o>	<U24E4>	CIRCLED LATIN SMALL LETTER U
<v-o>	<U24E5>	CIRCLED LATIN SMALL LETTER V
<w-o>	<U24E6>	CIRCLED LATIN SMALL LETTER W
<x-o>	<U24E7>	CIRCLED LATIN SMALL LETTER X
<y-o>	<U24E8>	CIRCLED LATIN SMALL LETTER Y
<z-o>	<U24E9>	CIRCLED LATIN SMALL LETTER Z
<0-o>	<U24EA>	CIRCLED DIGIT ZERO
<hh>	<U2500>	BOX DRAWINGS LIGHT HORIZONTAL
<HH->	<U2501>	BOX DRAWINGS HEAVY HORIZONTAL
<vv>	<U2502>	BOX DRAWINGS LIGHT VERTICAL
<VV->	<U2503>	BOX DRAWINGS HEAVY VERTICAL
<3->	<U2504>	BOX DRAWINGS LIGHT TRIPLE DASH HORIZONTAL
<3_>	<U2505>	BOX DRAWINGS HEAVY TRIPLE DASH HORIZONTAL
<3!>	<U2506>	BOX DRAWINGS LIGHT TRIPLE DASH VERTICAL
<3//>	<U2507>	BOX DRAWINGS HEAVY TRIPLE DASH VERTICAL
<4->	<U2508>	BOX DRAWINGS LIGHT QUADRUPLE DASH HORIZONTAL
<4_>	<U2509>	BOX DRAWINGS HEAVY QUADRUPLE DASH HORIZONTAL
<4!>	<U250A>	BOX DRAWINGS LIGHT QUADRUPLE DASH VERTICAL
<4//>	<U250B>	BOX DRAWINGS HEAVY QUADRUPLE DASH VERTICAL
<dr>	<U250C>	BOX DRAWINGS LIGHT DOWN AND RIGHT
<dR->	<U250D>	BOX DRAWINGS DOWN LIGHT AND RIGHT HEAVY
<Dr->	<U250E>	BOX DRAWINGS DOWN HEAVY AND RIGHT LIGHT
<DR->	<U250F>	BOX DRAWINGS HEAVY DOWN AND RIGHT
<dL>	<U2510>	BOX DRAWINGS LIGHT DOWN AND LEFT
<dL->	<U2511>	BOX DRAWINGS DOWN LIGHT AND LEFT HEAVY
<DL->	<U2512>	BOX DRAWINGS DOWN HEAVY AND LEFT LIGHT
<LD->	<U2513>	BOX DRAWINGS HEAVY DOWN AND LEFT
<ur>	<U2514>	BOX DRAWINGS LIGHT UP AND RIGHT
<uR->	<U2515>	BOX DRAWINGS UP LIGHT AND RIGHT HEAVY
<Ur->	<U2516>	BOX DRAWINGS UP HEAVY AND RIGHT LIGHT
<UR->	<U2517>	BOX DRAWINGS HEAVY UP AND RIGHT
	<U2518>	BOX DRAWINGS LIGHT UP AND LEFT
<uL->	<U2519>	BOX DRAWINGS UP LIGHT AND LEFT HEAVY
<Ul->	<U251A>	BOX DRAWINGS UP HEAVY AND LEFT LIGHT
<UL->	<U251B>	BOX DRAWINGS HEAVY UP AND LEFT
<vr>	<U251C>	BOX DRAWINGS LIGHT VERTICAL AND RIGHT
<vR->	<U251D>	BOX DRAWINGS VERTICAL LIGHT AND RIGHT HEAVY
<Udr>	<U251E>	BOX DRAWINGS UP HEAVY AND RIGHT DOWN LIGHT
<uDr>	<U251F>	BOX DRAWINGS DOWN HEAVY AND RIGHT UP LIGHT
<Vr->	<U2520>	BOX DRAWINGS VERTICAL HEAVY AND RIGHT LIGHT
<UdR>	<U2521>	BOX DRAWINGS DOWN LIGHT AND RIGHT UP HEAVY
<uDl>	<U2522>	BOX DRAWINGS UP LIGHT AND RIGHT DOWN HEAVY
<VR->	<U2523>	BOX DRAWINGS HEAVY VERTICAL AND RIGHT
<vl>	<U2524>	BOX DRAWINGS LIGHT VERTICAL AND LEFT
<vL->	<U2525>	BOX DRAWINGS VERTICAL LIGHT AND LEFT HEAVY
<Udl>	<U2526>	BOX DRAWINGS UP HEAVY AND LEFT DOWN LIGHT
<udl>	<U2527>	BOX DRAWINGS DOWN HEAVY AND LEFT UP LIGHT
<Vl->	<U2528>	BOX DRAWINGS VERTICAL HEAVY AND LEFT LIGHT
<UdL>	<U2529>	BOX DRAWINGS DOWN LIGHT AND LEFT UP HEAVY
<uDL>	<U252A>	BOX DRAWINGS UP LIGHT AND LEFT DOWN HEAVY
<VL->	<U252B>	BOX DRAWINGS HEAVY VERTICAL AND LEFT
<dh>	<U252C>	BOX DRAWINGS LIGHT DOWN AND HORIZONTAL
<dLr>	<U252D>	BOX DRAWINGS LEFT HEAVY AND RIGHT DOWN LIGHT
<dLr>	<U252E>	BOX DRAWINGS RIGHT HEAVY AND LEFT DOWN LIGHT
<dH->	<U252F>	BOX DRAWINGS DOWN LIGHT AND HORIZONTAL HEAVY
<Dh->	<U2530>	BOX DRAWINGS DOWN HEAVY AND HORIZONTAL LIGHT
<DLr>	<U2531>	BOX DRAWINGS RIGHT LIGHT AND LEFT DOWN HEAVY
<DLr>	<U2532>	BOX DRAWINGS LEFT LIGHT AND RIGHT DOWN HEAVY
<DH->	<U2533>	BOX DRAWINGS HEAVY DOWN AND HORIZONTAL
<uh>	<U2534>	BOX DRAWINGS LIGHT UP AND HORIZONTAL
<uLr>	<U2535>	BOX DRAWINGS LEFT HEAVY AND RIGHT UP LIGHT
<uLr>	<U2536>	BOX DRAWINGS RIGHT HEAVY AND LEFT UP LIGHT
<uH->	<U2537>	BOX DRAWINGS UP LIGHT AND HORIZONTAL HEAVY
<Uh->	<U2538>	BOX DRAWINGS UP HEAVY AND HORIZONTAL LIGHT
<ULr>	<U2539>	BOX DRAWINGS RIGHT LIGHT AND LEFT UP HEAVY
<ULr>	<U253A>	BOX DRAWINGS LEFT LIGHT AND RIGHT UP HEAVY
<UH->	<U253B>	BOX DRAWINGS HEAVY UP AND HORIZONTAL
<vh>	<U253C>	BOX DRAWINGS LIGHT VERTICAL AND HORIZONTAL
<vLr>	<U253D>	BOX DRAWINGS LEFT HEAVY AND RIGHT VERTICAL LIGHT
<vLr>	<U253E>	BOX DRAWINGS RIGHT HEAVY AND LEFT VERTICAL LIGHT
<vH->	<U253F>	BOX DRAWINGS VERTICAL LIGHT AND HORIZONTAL HEAVY
<Udh>	<U2540>	BOX DRAWINGS UP HEAVY AND DOWN HORIZONTAL LIGHT
<uDh>	<U2541>	BOX DRAWINGS DOWN HEAVY AND UP HORIZONTAL LIGHT
<Vh->	<U2542>	BOX DRAWINGS VERTICAL HEAVY AND HORIZONTAL LIGHT
<UdLr>	<U2543>	BOX DRAWINGS LEFT UP HEAVY AND RIGHT DOWN LIGHT
<UdLr>	<U2544>	BOX DRAWINGS RIGHT UP HEAVY AND LEFT DOWN LIGHT
<uDlR>	<U2545>	BOX DRAWINGS LEFT DOWN HEAVY AND RIGHT UP LIGHT
<uDlR>	<U2546>	BOX DRAWINGS RIGHT DOWN HEAVY AND LEFT UP LIGHT

<UdH>	<U2547>	BOX DRAWINGS DOWN LIGHT AND UP HORIZONTAL HEAVY
<uDh>	<U2548>	BOX DRAWINGS UP LIGHT AND DOWN HORIZONTAL HEAVY
<VLr>	<U2549>	BOX DRAWINGS RIGHT LIGHT AND LEFT VERTICAL HEAVY
<VLR>	<U254A>	BOX DRAWINGS LEFT LIGHT AND RIGHT VERTICAL HEAVY
<VH->	<U254B>	BOX DRAWINGS HEAVY VERTICAL AND HORIZONTAL
<HH>	<U2550>	BOX DRAWINGS DOUBLE HORIZONTAL
<VV>	<U2551>	BOX DRAWINGS DOUBLE VERTICAL
<dR>	<U2552>	BOX DRAWINGS DOWN SINGLE AND RIGHT DOUBLE
<Dr>	<U2553>	BOX DRAWINGS DOWN DOUBLE AND RIGHT SINGLE
<DR>	<U2554>	BOX DRAWINGS DOUBLE DOWN AND RIGHT
<dL>	<U2555>	BOX DRAWINGS DOWN SINGLE AND LEFT DOUBLE
<DL>	<U2556>	BOX DRAWINGS DOWN DOUBLE AND LEFT SINGLE
<LD>	<U2557>	BOX DRAWINGS DOUBLE DOWN AND LEFT
<uR>	<U2558>	BOX DRAWINGS UP SINGLE AND RIGHT DOUBLE
<Ur>	<U2559>	BOX DRAWINGS UP DOUBLE AND RIGHT SINGLE
<UR>	<U255A>	BOX DRAWINGS DOUBLE UP AND RIGHT
	<U255B>	BOX DRAWINGS UP SINGLE AND LEFT DOUBLE
	<U255C>	BOX DRAWINGS UP DOUBLE AND LEFT SINGLE
	<U255D>	BOX DRAWINGS DOUBLE UP AND LEFT
<vR>	<U255E>	BOX DRAWINGS VERTICAL SINGLE AND RIGHT DOUBLE
<Vr>	<U255F>	BOX DRAWINGS VERTICAL DOUBLE AND RIGHT SINGLE
<VR>	<U2560>	BOX DRAWINGS DOUBLE VERTICAL AND RIGHT
<vL>	<U2561>	BOX DRAWINGS VERTICAL SINGLE AND LEFT DOUBLE
<VL>	<U2562>	BOX DRAWINGS VERTICAL DOUBLE AND LEFT SINGLE
<VL>	<U2563>	BOX DRAWINGS DOUBLE VERTICAL AND LEFT
<dH>	<U2564>	BOX DRAWINGS DOWN SINGLE AND HORIZONTAL DOUBLE
<Dh>	<U2565>	BOX DRAWINGS DOWN DOUBLE AND HORIZONTAL SINGLE
<DH>	<U2566>	BOX DRAWINGS DOUBLE DOWN AND HORIZONTAL
<uH>	<U2567>	BOX DRAWINGS UP SINGLE AND HORIZONTAL DOUBLE
<Uh>	<U2568>	BOX DRAWINGS UP DOUBLE AND HORIZONTAL SINGLE
<UH>	<U2569>	BOX DRAWINGS DOUBLE UP AND HORIZONTAL
<vH>	<U256A>	BOX DRAWINGS VERTICAL SINGLE AND HORIZONTAL DOUBLE
<Vh>	<U256B>	BOX DRAWINGS VERTICAL DOUBLE AND HORIZONTAL SINGLE
<VH>	<U256C>	BOX DRAWINGS DOUBLE VERTICAL AND HORIZONTAL
<FD>	<U2571>	BOX DRAWINGS LIGHT DIAGONAL UPPER RIGHT TO LOWER LEFT
<BD>	<U2572>	BOX DRAWINGS LIGHT DIAGONAL UPPER LEFT TO LOWER RIGHT
<TB>	<U2580>	UPPER HALF BLOCK
<LB>	<U2584>	LOWER HALF BLOCK
<FB>	<U2588>	FULL BLOCK
<LB>	<U258C>	LEFT HALF BLOCK
<RB>	<U2590>	RIGHT HALF BLOCK
<.S>	<U2591>	LIGHT SHADE
<:S>	<U2592>	MEDIUM SHADE
<?S>	<U2593>	DARK SHADE
<fs>	<U25A0>	BLACK SQUARE
<OS>	<U25A1>	WHITE SQUARE
<RO>	<U25A2>	WHITE SQUARE WITH ROUNDED CORNERS
<Rr>	<U25A3>	WHITE SQUARE CONTAINING BLACK SMALL SQUARE
<RF>	<U25A4>	SQUARE WITH HORIZONTAL FILL
<RY>	<U25A5>	SQUARE WITH VERTICAL FILL
<RH>	<U25A6>	SQUARE WITH ORTHOGONAL CROSSHATCH FILL
<RZ>	<U25A7>	SQUARE WITH UPPER LEFT TO LOWER RIGHT FILL
<RK>	<U25A8>	SQUARE WITH UPPER RIGHT TO LOWER LEFT FILL
<RX>	<U25A9>	SQUARE WITH DIAGONAL CROSSHATCH FILL
<sB>	<U25AA>	BLACK SMALL SQUARE
<SR>	<U25AC>	BLACK RECTANGLE
<Or>	<U25AD>	WHITE RECTANGLE
<UT>	<U25B2>	BLACK UP-POINTING TRIANGLE
<uT>	<U25B3>	WHITE UP-POINTING TRIANGLE
<Tr>	<U25B7>	WHITE RIGHT-POINTING TRIANGLE
<PR>	<U25BA>	BLACK RIGHT-POINTING POINTER
<Dt>	<U25BC>	BLACK DOWN-POINTING TRIANGLE
<dT>	<U25BD>	WHITE DOWN-POINTING TRIANGLE
<tl>	<U25C1>	WHITE LEFT-POINTING TRIANGLE
<PL>	<U25C4>	BLACK LEFT-POINTING POINTER
<Db>	<U25C6>	BLACK DIAMOND
<Dw>	<U25C7>	WHITE DIAMOND
<LZ>	<U25CA>	LOZENGE
<Om>	<U25CB>	WHITE CIRCLE
<Oo>	<U25CE>	BULLSEYE
<OM>	<U25CF>	BLACK CIRCLE
	<U25D0>	CIRCLE WITH LEFT HALF BLACK
<OR>	<U25D1>	CIRCLE WITH RIGHT HALF BLACK
<Sn>	<U25D8>	INVERSE BULLET
<Ic>	<U25D9>	INVERSE WHITE CIRCLE
<Fd>	<U25E2>	BLACK LOWER RIGHT TRIANGLE
<Bd>	<U25E3>	BLACK LOWER LEFT TRIANGLE
<Ci>	<U25EF>	LARGE CIRCLE
<*2>	<U2605>	BLACK STAR
<*1>	<U2606>	WHITE STAR
<TEL>	<U260E>	BLACK TELEPHONE
<tel>	<U260F>	WHITE TELEPHONE
<<H>	<U261C>	WHITE LEFT POINTING INDEX
</>H>	<U261E>	WHITE RIGHT POINTING INDEX
<Ou>	<U263A>	WHITE SMILING FACE
<OU>	<U263B>	BLACK SMILING FACE
<SU>	<U263C>	WHITE SUN WITH RAYS
<Fm>	<U2640>	FEMALE SIGN

<M1>	<U2642>	MALE SIGN
<cS>	<U2660>	BLACK SPADE SUIT
<cH>	<U2661>	WHITE HEART SUIT
<cD>	<U2662>	WHITE DIAMOND SUIT
<cC>	<U2663>	BLACK CLUB SUIT
<cS->	<U2664>	WHITE SPADE SUIT
<cH->	<U2665>	BLACK HEART SUIT
<cD->	<U2666>	BLACK DIAMOND SUIT
<cC->	<U2667>	WHITE CLUB SUIT
<Md>	<U2669>	QUARTER NOTE
<M8>	<U266A>	EIGHTH NOTE
<M2>	<U266B>	BEAMED EIGHTH NOTES
<M16>	<U266C>	BEAMED SIXTEENTH NOTES
<Mb>	<U266D>	MUSIC FLAT SIGN
<Mx>	<U266E>	MUSIC NATURAL SIGN
<MX>	<U266F>	MUSIC SHARP SIGN
<OK>	<U2713>	CHECK MARK
<XX>	<U2717>	BALLOT X
<-X>	<U2720>	MALTESE CROSS
<IS>	<U3000>	IDEOGRAPHIC SPACE
< , _>	<U3001>	IDEOGRAPHIC COMMA
< . _>	<U3002>	IDEOGRAPHIC FULL STOP
<+ ">	<U3003>	DITTO MARK
<JIS>	<U3004>	JAPANESE INDUSTRIAL STANDARD SYMBOL
<* _>	<U3005>	IDEOGRAPHIC ITERATION MARK
< ; _>	<U3006>	IDEOGRAPHIC CLOSING MARK
<0 _>	<U3007>	IDEOGRAPHIC NUMBER ZERO
<<+>	<U300A>	LEFT DOUBLE ANGLE BRACKET
</>+>	<U300B>	RIGHT DOUBLE ANGLE BRACKET
<<'>	<U300C>	LEFT CORNER BRACKET
</>'>	<U300D>	RIGHT CORNER BRACKET
<<">	<U300E>	LEFT WHITE CORNER BRACKET
</>">	<U300F>	RIGHT WHITE CORNER BRACKET
<(">	<U3010>	LEFT BLACK LENTICULAR BRACKET
<)">	<U3011>	RIGHT BLACK LENTICULAR BRACKET
<=T>	<U3012>	POSTAL MARK
<=_>	<U3013>	GETA MARK
<('>	<U3014>	LEFT TORTOISE SHELL BRACKET
<)'>	<U3015>	RIGHT TORTOISE SHELL BRACKET
<(I>	<U3016>	LEFT WHITE LENTICULAR BRACKET
<I)>	<U3017>	RIGHT WHITE LENTICULAR BRACKET
<-?>	<U301C>	WAVE DASH
<=T:)>	<U3020>	POSTAL MARK FACE
<A5>	<U3041>	HIRAGANA LETTER SMALL A
<a5>	<U3042>	HIRAGANA LETTER A
<I5>	<U3043>	HIRAGANA LETTER SMALL I
<i5>	<U3044>	HIRAGANA LETTER I
<U5>	<U3045>	HIRAGANA LETTER SMALL U
<u5>	<U3046>	HIRAGANA LETTER U
<E5>	<U3047>	HIRAGANA LETTER SMALL E
<e5>	<U3048>	HIRAGANA LETTER E
<O5>	<U3049>	HIRAGANA LETTER SMALL O
<o5>	<U304A>	HIRAGANA LETTER O
<ka>	<U304B>	HIRAGANA LETTER KA
<ga>	<U304C>	HIRAGANA LETTER GA
<ki>	<U304D>	HIRAGANA LETTER KI
<gi>	<U304E>	HIRAGANA LETTER GI
<ku>	<U304F>	HIRAGANA LETTER KU
<gu>	<U3050>	HIRAGANA LETTER GU
<ke>	<U3051>	HIRAGANA LETTER KE
<ge>	<U3052>	HIRAGANA LETTER GE
<ko>	<U3053>	HIRAGANA LETTER KO
<go>	<U3054>	HIRAGANA LETTER GO
<sa>	<U3055>	HIRAGANA LETTER SA
<za>	<U3056>	HIRAGANA LETTER ZA
<si>	<U3057>	HIRAGANA LETTER SI
<zi>	<U3058>	HIRAGANA LETTER ZI
<su>	<U3059>	HIRAGANA LETTER SU
<zu>	<U305A>	HIRAGANA LETTER ZU
<se>	<U305B>	HIRAGANA LETTER SE
<ze>	<U305C>	HIRAGANA LETTER ZE
<so>	<U305D>	HIRAGANA LETTER SO
<zo>	<U305E>	HIRAGANA LETTER ZO
<ta>	<U305F>	HIRAGANA LETTER TA
<da>	<U3060>	HIRAGANA LETTER DA
<ti>	<U3061>	HIRAGANA LETTER TI
<di>	<U3062>	HIRAGANA LETTER DI
<tU>	<U3063>	HIRAGANA LETTER SMALL TU
<tu>	<U3064>	HIRAGANA LETTER TU
<du>	<U3065>	HIRAGANA LETTER DU
<te>	<U3066>	HIRAGANA LETTER TE
<de>	<U3067>	HIRAGANA LETTER DE
<to>	<U3068>	HIRAGANA LETTER TO
<do>	<U3069>	HIRAGANA LETTER DO
<na>	<U306A>	HIRAGANA LETTER NA
<ni>	<U306B>	HIRAGANA LETTER NI
<nu>	<U306C>	HIRAGANA LETTER NU
<ne>	<U306D>	HIRAGANA LETTER NE

<no>	<U306E>	HIRAGANA LETTER NO
<ha>	<U306F>	HIRAGANA LETTER HA
<ba>	<U3070>	HIRAGANA LETTER BA
<pa>	<U3071>	HIRAGANA LETTER PA
<hi>	<U3072>	HIRAGANA LETTER HI
<bi>	<U3073>	HIRAGANA LETTER BI
<pi>	<U3074>	HIRAGANA LETTER PI
<hu>	<U3075>	HIRAGANA LETTER HU
<bu>	<U3076>	HIRAGANA LETTER BU
<pu>	<U3077>	HIRAGANA LETTER PU
<he>	<U3078>	HIRAGANA LETTER HE
<be>	<U3079>	HIRAGANA LETTER BE
<pe>	<U307A>	HIRAGANA LETTER PE
<ho>	<U307B>	HIRAGANA LETTER HO
<bo>	<U307C>	HIRAGANA LETTER BO
<po>	<U307D>	HIRAGANA LETTER PO
<ma>	<U307E>	HIRAGANA LETTER MA
<mi>	<U307F>	HIRAGANA LETTER MI
<mu>	<U3080>	HIRAGANA LETTER MU
<me>	<U3081>	HIRAGANA LETTER ME
<mo>	<U3082>	HIRAGANA LETTER MO
<yA>	<U3083>	HIRAGANA LETTER SMALL YA
<yA>	<U3084>	HIRAGANA LETTER YA
<yU>	<U3085>	HIRAGANA LETTER SMALL YU
<yU>	<U3086>	HIRAGANA LETTER YU
<yO>	<U3087>	HIRAGANA LETTER SMALL YO
<yO>	<U3088>	HIRAGANA LETTER YO
<ra>	<U3089>	HIRAGANA LETTER RA
<ri>	<U308A>	HIRAGANA LETTER RI
<ru>	<U308B>	HIRAGANA LETTER RU
<re>	<U308C>	HIRAGANA LETTER RE
<ro>	<U308D>	HIRAGANA LETTER RO
<wA>	<U308E>	HIRAGANA LETTER SMALL WA
<wA>	<U308F>	HIRAGANA LETTER WA
<wi>	<U3090>	HIRAGANA LETTER WI
<we>	<U3091>	HIRAGANA LETTER WE
<wo>	<U3092>	HIRAGANA LETTER WO
<n5>	<U3093>	HIRAGANA LETTER N
<vu>	<U3094>	HIRAGANA LETTER VU
<"5>	<U309B>	KATAKANA-HIRAGANA VOICED SOUND MARK
<05>	<U309C>	KATAKANA-HIRAGANA SEMI-VOICED SOUND MARK
<*5>	<U309D>	HIRAGANA ITERATION MARK
<+5>	<U309E>	HIRAGANA VOICED ITERATION MARK
<a6>	<U30A1>	KATAKANA LETTER SMALL A
<A6>	<U30A2>	KATAKANA LETTER A
<i6>	<U30A3>	KATAKANA LETTER SMALL I
<I6>	<U30A4>	KATAKANA LETTER I
<u6>	<U30A5>	KATAKANA LETTER SMALL U
<U6>	<U30A6>	KATAKANA LETTER U
<e6>	<U30A7>	KATAKANA LETTER SMALL E
<E6>	<U30A8>	KATAKANA LETTER E
<o6>	<U30A9>	KATAKANA LETTER SMALL O
<O6>	<U30AA>	KATAKANA LETTER O
<Ka>	<U30AB>	KATAKANA LETTER KA
<Ga>	<U30AC>	KATAKANA LETTER GA
<Ki>	<U30AD>	KATAKANA LETTER KI
<Gi>	<U30AE>	KATAKANA LETTER GI
<Ku>	<U30AF>	KATAKANA LETTER KU
<Gu>	<U30B0>	KATAKANA LETTER GU
<Ke>	<U30B1>	KATAKANA LETTER KE
<Ge>	<U30B2>	KATAKANA LETTER GE
<Ko>	<U30B3>	KATAKANA LETTER KO
<Go>	<U30B4>	KATAKANA LETTER GO
<Sa>	<U30B5>	KATAKANA LETTER SA
<Za>	<U30B6>	KATAKANA LETTER ZA
<Si>	<U30B7>	KATAKANA LETTER SI
<Zi>	<U30B8>	KATAKANA LETTER ZI
<Su>	<U30B9>	KATAKANA LETTER SU
<Zu>	<U30BA>	KATAKANA LETTER ZU
<Se>	<U30BB>	KATAKANA LETTER SE
<Ze>	<U30BC>	KATAKANA LETTER ZE
<So>	<U30BD>	KATAKANA LETTER SO
<Zo>	<U30BE>	KATAKANA LETTER ZO
<Ta>	<U30BF>	KATAKANA LETTER TA
<Da>	<U30C0>	KATAKANA LETTER DA
<Ti>	<U30C1>	KATAKANA LETTER TI
<Di>	<U30C2>	KATAKANA LETTER DI
<Tu>	<U30C3>	KATAKANA LETTER SMALL TU
<Tu>	<U30C4>	KATAKANA LETTER TU
<Du>	<U30C5>	KATAKANA LETTER DU
<Te>	<U30C6>	KATAKANA LETTER TE
<De>	<U30C7>	KATAKANA LETTER DE
<To>	<U30C8>	KATAKANA LETTER TO
<Do>	<U30C9>	KATAKANA LETTER DO
<Na>	<U30CA>	KATAKANA LETTER NA
<Ni>	<U30CB>	KATAKANA LETTER NI
<Nu>	<U30CC>	KATAKANA LETTER NU
<Ne>	<U30CD>	KATAKANA LETTER NE

<No>	<U30CE>	KATAKANA LETTER NO
<Ha>	<U30CF>	KATAKANA LETTER HA
<Ba>	<U30D0>	KATAKANA LETTER BA
<Pa>	<U30D1>	KATAKANA LETTER PA
<Hi>	<U30D2>	KATAKANA LETTER HI
<Bi>	<U30D3>	KATAKANA LETTER BI
<Pi>	<U30D4>	KATAKANA LETTER PI
<Hu>	<U30D5>	KATAKANA LETTER HU
<Bu>	<U30D6>	KATAKANA LETTER BU
<Pu>	<U30D7>	KATAKANA LETTER PU
<He>	<U30D8>	KATAKANA LETTER HE
<Be>	<U30D9>	KATAKANA LETTER BE
<Pe>	<U30DA>	KATAKANA LETTER PE
<Ho>	<U30DB>	KATAKANA LETTER HO
<Bo>	<U30DC>	KATAKANA LETTER BO
<Po>	<U30DD>	KATAKANA LETTER PO
<Ma>	<U30DE>	KATAKANA LETTER MA
<Mi>	<U30DF>	KATAKANA LETTER MI
<Mu>	<U30E0>	KATAKANA LETTER MU
<Me>	<U30E1>	KATAKANA LETTER ME
<Mo>	<U30E2>	KATAKANA LETTER MO
<YA>	<U30E3>	KATAKANA LETTER SMALL YA
<Ya>	<U30E4>	KATAKANA LETTER YA
<YU>	<U30E5>	KATAKANA LETTER SMALL YU
<Yu>	<U30E6>	KATAKANA LETTER YU
<YO>	<U30E7>	KATAKANA LETTER SMALL YO
<Yo>	<U30E8>	KATAKANA LETTER YO
<Ra>	<U30E9>	KATAKANA LETTER RA
<Ri>	<U30EA>	KATAKANA LETTER RI
<Ru>	<U30EB>	KATAKANA LETTER RU
<Re>	<U30EC>	KATAKANA LETTER RE
<Ro>	<U30ED>	KATAKANA LETTER RO
<WA>	<U30EE>	KATAKANA LETTER SMALL WA
<Wa>	<U30EF>	KATAKANA LETTER WA
<Wi>	<U30F0>	KATAKANA LETTER WI
<We>	<U30F1>	KATAKANA LETTER WE
<Wo>	<U30F2>	KATAKANA LETTER WO
<N6>	<U30F3>	KATAKANA LETTER N
<Vu>	<U30F4>	KATAKANA LETTER VU
<KA>	<U30F5>	KATAKANA LETTER SMALL KA
<KE>	<U30F6>	KATAKANA LETTER SMALL KE
<Va>	<U30F7>	KATAKANA LETTER VA
<Vi>	<U30F8>	KATAKANA LETTER VI
<Ve>	<U30F9>	KATAKANA LETTER VE
<Vo>	<U30FA>	KATAKANA LETTER VO
<.6>	<U30FB>	KATAKANA MIDDLE DOT
<-6>	<U30FC>	KATAKANA-HIRAGANA PROLONGED SOUND MARK
<*6>	<U30FD>	KATAKANA ITERATION MARK
<+6>	<U30FE>	KATAKANA VOICED ITERATION MARK
<b4>	<U3105>	BOPOMOFO LETTER B
<p4>	<U3106>	BOPOMOFO LETTER P
<m4>	<U3107>	BOPOMOFO LETTER M
<f4>	<U3108>	BOPOMOFO LETTER F
<d4>	<U3109>	BOPOMOFO LETTER D
<t4>	<U310A>	BOPOMOFO LETTER T
<n4>	<U310B>	BOPOMOFO LETTER N
<l4>	<U310C>	BOPOMOFO LETTER L
<q4>	<U310D>	BOPOMOFO LETTER G
<k4>	<U310E>	BOPOMOFO LETTER K
<h4>	<U310F>	BOPOMOFO LETTER H
<j4>	<U3110>	BOPOMOFO LETTER J
<q4>	<U3111>	BOPOMOFO LETTER Q
<x4>	<U3112>	BOPOMOFO LETTER X
<zh>	<U3113>	BOPOMOFO LETTER ZH
<ch>	<U3114>	BOPOMOFO LETTER CH
<sh>	<U3115>	BOPOMOFO LETTER SH
<r4>	<U3116>	BOPOMOFO LETTER R
<z4>	<U3117>	BOPOMOFO LETTER Z
<c4>	<U3118>	BOPOMOFO LETTER C
<s4>	<U3119>	BOPOMOFO LETTER S
<a4>	<U311A>	BOPOMOFO LETTER A
<o4>	<U311B>	BOPOMOFO LETTER O
<e4>	<U311C>	BOPOMOFO LETTER E
<eh4>	<U311D>	BOPOMOFO LETTER EH
<ai>	<U311E>	BOPOMOFO LETTER AI
<ei>	<U311F>	BOPOMOFO LETTER EI
<au>	<U3120>	BOPOMOFO LETTER AU
<ou>	<U3121>	BOPOMOFO LETTER OU
<an>	<U3122>	BOPOMOFO LETTER AN
<en>	<U3123>	BOPOMOFO LETTER EN
<aN>	<U3124>	BOPOMOFO LETTER ANG
<eN>	<U3125>	BOPOMOFO LETTER ENG
<er>	<U3126>	BOPOMOFO LETTER ER
<i4>	<U3127>	BOPOMOFO LETTER I
<u4>	<U3128>	BOPOMOFO LETTER U
<iu>	<U3129>	BOPOMOFO LETTER IU
<v4>	<U312A>	BOPOMOFO LETTER V
<nG>	<U312B>	BOPOMOFO LETTER NG

5705	<gn>	<U312C>	BOPOMOFO LETTER GN
5706	<(JU)>	<U321C>	PARENTHESESIZED HANGUL CIEUC U
5707	<1c>	<U3220>	PARENTHESESIZED IDEOGRAPH ONE
5708	<2c>	<U3221>	PARENTHESESIZED IDEOGRAPH TWO
5709	<3c>	<U3222>	PARENTHESESIZED IDEOGRAPH THREE
5710	<4c>	<U3223>	PARENTHESESIZED IDEOGRAPH FOUR
5711	<5c>	<U3224>	PARENTHESESIZED IDEOGRAPH FIVE
5712	<6c>	<U3225>	PARENTHESESIZED IDEOGRAPH SIX
5713	<7c>	<U3226>	PARENTHESESIZED IDEOGRAPH SEVEN
5714	<8c>	<U3227>	PARENTHESESIZED IDEOGRAPH EIGHT
5715	<9c>	<U3228>	PARENTHESESIZED IDEOGRAPH NINE
5716	<10c>	<U3229>	PARENTHESESIZED IDEOGRAPH TEN
5717	<KSC>	<U327F>	KOREAN STANDARD SYMBOL
5718	<am>	<U33C2>	SQUARE AM
5719	<pm>	<U33D8>	SQUARE PM
5720	<ff>	<UFB00>	LATIN SMALL LIGATURE FF
5721	<fi>	<UFB01>	LATIN SMALL LIGATURE FI
5722	<fl>	<UFB02>	LATIN SMALL LIGATURE FL
5723	<ffi>	<UFB03>	LATIN SMALL LIGATURE FFI
5724	<ffl>	<UFB04>	LATIN SMALL LIGATURE FFL
5725	<St>	<UFB05>	LATIN SMALL LIGATURE LONG S T
5726	<st>	<UFB06>	LATIN SMALL LIGATURE ST
5727	<3+i>	<UFE7D>	ARABIC SHADDA MEDIAL FORM
5728	<aM.>	<UFE82>	ARABIC LETTER ALEF WITH MADDA ABOVE FINAL FORM
5729	<aH.>	<UFE84>	ARABIC LETTER ALEF WITH HAMZA ABOVE FINAL FORM
5730	<ah.>	<UFE88>	ARABIC LETTER ALEF WITH HAMZA BELOW FINAL FORM
5731	<a+>	<UFE8D>	ARABIC LETTER ALEF ISOLATED FORM
5732	<a+.>	<UFE8E>	ARABIC LETTER ALEF FINAL FORM
5733	<b+>	<UFE8F>	ARABIC LETTER BEH ISOLATED FORM
5734	<b+.>	<UFE90>	ARABIC LETTER BEH FINAL FORM
5735	<b+;.>	<UFE91>	ARABIC LETTER BEH INITIAL FORM
5736	<b+i>	<UFE92>	ARABIC LETTER BEH MEDIAL FORM
5737	<tm->	<UFE93>	ARABIC LETTER TEH MARBUTA ISOLATED FORM
5738	<tm.>	<UFE94>	ARABIC LETTER TEH MARBUTA FINAL FORM
5739	<t+>	<UFE95>	ARABIC LETTER TEH ISOLATED FORM
5740	<t+.>	<UFE96>	ARABIC LETTER TEH FINAL FORM
5741	<t+;.>	<UFE97>	ARABIC LETTER TEH INITIAL FORM
5742	<t+i>	<UFE98>	ARABIC LETTER TEH MEDIAL FORM
5743	<tk->	<UFE99>	ARABIC LETTER THEH ISOLATED FORM
5744	<tk.>	<UFE9A>	ARABIC LETTER THEH FINAL FORM
5745	<tk;.>	<UFE9B>	ARABIC LETTER THEH INITIAL FORM
5746	<tk+i>	<UFE9C>	ARABIC LETTER THEH MEDIAL FORM
5747	<g+>	<UFE9D>	ARABIC LETTER JEEM ISOLATED FORM
5748	<g+.>	<UFE9E>	ARABIC LETTER JEEM FINAL FORM
5749	<g+;.>	<UFE9F>	ARABIC LETTER JEEM INITIAL FORM
5750	<g+i>	<UFEA0>	ARABIC LETTER JEEM MEDIAL FORM
5751	<hk->	<UFEA1>	ARABIC LETTER HAH ISOLATED FORM
5752	<hk.>	<UFEA2>	ARABIC LETTER HAH FINAL FORM
5753	<hk;.>	<UFEA3>	ARABIC LETTER HAH INITIAL FORM
5754	<hk+i>	<UFEA4>	ARABIC LETTER HAH MEDIAL FORM
5755	<x+>	<UFEA5>	ARABIC LETTER KHAH ISOLATED FORM
5756	<x+.>	<UFEA6>	ARABIC LETTER KHAH FINAL FORM
5757	<x+;.>	<UFEA7>	ARABIC LETTER KHAH INITIAL FORM
5758	<x+i>	<UFEA8>	ARABIC LETTER KHAH MEDIAL FORM
5759	<d+>	<UFEA9>	ARABIC LETTER DAL ISOLATED FORM
5760	<d+.>	<UFEAA>	ARABIC LETTER DAL FINAL FORM
5761	<dk->	<UFEAB>	ARABIC LETTER THAL ISOLATED FORM
5762	<dk.>	<UFEAC>	ARABIC LETTER THAL FINAL FORM
5763	<r+>	<UFEAD>	ARABIC LETTER REH ISOLATED FORM
5764	<r+.>	<UFEAE>	ARABIC LETTER REH FINAL FORM
5765	<z+>	<UFEAF>	ARABIC LETTER ZAIN ISOLATED FORM
5766	<z+.>	<UFEB0>	ARABIC LETTER ZAIN FINAL FORM
5767	<s+>	<UFEB1>	ARABIC LETTER SEEN ISOLATED FORM
5768	<s+.>	<UFEB2>	ARABIC LETTER SEEN FINAL FORM
5769	<s+;.>	<UFEB3>	ARABIC LETTER SEEN INITIAL FORM
5770	<s+i>	<UFEB4>	ARABIC LETTER SEEN MEDIAL FORM
5771	<sn->	<UFEB5>	ARABIC LETTER SHEEN ISOLATED FORM
5772	<sn.>	<UFEB6>	ARABIC LETTER SHEEN FINAL FORM
5773	<sn;.>	<UFEB7>	ARABIC LETTER SHEEN INITIAL FORM
5774	<sn+i>	<UFEB8>	ARABIC LETTER SHEEN MEDIAL FORM
5775	<c+>	<UFEB9>	ARABIC LETTER SAD ISOLATED FORM
5776	<c+.>	<UFEBA>	ARABIC LETTER SAD FINAL FORM
5777	<c+;.>	<UFEBB>	ARABIC LETTER SAD INITIAL FORM
5778	<c+i>	<UFEBC>	ARABIC LETTER SAD MEDIAL FORM
5779	<dd->	<UFEBD>	ARABIC LETTER DAD ISOLATED FORM
5780	<dd.>	<UFEBE>	ARABIC LETTER DAD FINAL FORM
5781	<dd;.>	<UFEBF>	ARABIC LETTER DAD INITIAL FORM
5782	<dd+i>	<UFEC0>	ARABIC LETTER DAD MEDIAL FORM
5783	<tj->	<UFEC1>	ARABIC LETTER TAH ISOLATED FORM
5784	<tj.>	<UFEC2>	ARABIC LETTER TAH FINAL FORM
5785	<tj;.>	<UFEC3>	ARABIC LETTER TAH INITIAL FORM
5786	<tj+i>	<UFEC4>	ARABIC LETTER TAH MEDIAL FORM
5787	<zH->	<UFEC5>	ARABIC LETTER ZAH ISOLATED FORM
5788	<zH.>	<UFEC6>	ARABIC LETTER ZAH FINAL FORM
5789	<zH;.>	<UFEC7>	ARABIC LETTER ZAH INITIAL FORM
5790	<zH+i>	<UFEC8>	ARABIC LETTER ZAH MEDIAL FORM
5791	<e+>	<UFEC9>	ARABIC LETTER AIN ISOLATED FORM
5792	<e+.>	<UFECA>	ARABIC LETTER AIN FINAL FORM

<e+,>	<UFECB>	ARABIC LETTER AIN INITIAL FORM
<e+>	<UFEC C>	ARABIC LETTER AIN MEDIAL FORM
<i+>	<UFECD>	ARABIC LETTER GHAIN ISOLATED FORM
<i+.>	<UFECE>	ARABIC LETTER GHAIN FINAL FORM
<i+,>	<UFECF>	ARABIC LETTER GHAIN INITIAL FORM
<i+;>	<UFED0>	ARABIC LETTER GHAIN MEDIAL FORM
<f+>	<UFED1>	ARABIC LETTER FEH ISOLATED FORM
<f+.>	<UFED2>	ARABIC LETTER FEH FINAL FORM
<f+,>	<UFED3>	ARABIC LETTER FEH INITIAL FORM
<f+;>	<UFED4>	ARABIC LETTER FEH MEDIAL FORM
<q+>	<UFED5>	ARABIC LETTER QAF ISOLATED FORM
<q+.>	<UFED6>	ARABIC LETTER QAF FINAL FORM
<q+,>	<UFED7>	ARABIC LETTER QAF INITIAL FORM
<q+;>	<UFED8>	ARABIC LETTER QAF MEDIAL FORM
<k+>	<UFED9>	ARABIC LETTER KAF ISOLATED FORM
<k+.>	<UFEDA>	ARABIC LETTER KAF FINAL FORM
<k+,>	<UFEDB>	ARABIC LETTER KAF INITIAL FORM
<k+;>	<UFEDC>	ARABIC LETTER KAF MEDIAL FORM
<l+>	<UFEDD>	ARABIC LETTER LAM ISOLATED FORM
<l+.>	<UFEDE>	ARABIC LETTER LAM FINAL FORM
<l+,>	<UFEDF>	ARABIC LETTER LAM INITIAL FORM
<l+;>	<UFEE0>	ARABIC LETTER LAM MEDIAL FORM
<m+>	<UFEE1>	ARABIC LETTER MEEM ISOLATED FORM
<m+.>	<UFEE2>	ARABIC LETTER MEEM FINAL FORM
<m+,>	<UFEE3>	ARABIC LETTER MEEM INITIAL FORM
<m+;>	<UFEE4>	ARABIC LETTER MEEM MEDIAL FORM
<n+>	<UFEE5>	ARABIC LETTER NOON ISOLATED FORM
<n+.>	<UFEE6>	ARABIC LETTER NOON FINAL FORM
<n+,>	<UFEE7>	ARABIC LETTER NOON INITIAL FORM
<n+;>	<UFEE8>	ARABIC LETTER NOON MEDIAL FORM
<h+>	<UFEE9>	ARABIC LETTER HEH ISOLATED FORM
<h+.>	<UFEEA>	ARABIC LETTER HEH FINAL FORM
<h+,>	<UFEEB>	ARABIC LETTER HEH INITIAL FORM
<h+;>	<UFEEC>	ARABIC LETTER HEH MEDIAL FORM
<w+>	<UFEE D>	ARABIC LETTER WAW ISOLATED FORM
<w+.>	<UFEEE>	ARABIC LETTER WAW FINAL FORM
<j+>	<UFEEF>	ARABIC LETTER ALEF MAKSURA ISOLATED FORM
<j+.>	<UFEF0>	ARABIC LETTER ALEF MAKSURA FINAL FORM
<y+>	<UFEF1>	ARABIC LETTER YEH ISOLATED FORM
<y+.>	<UFEF2>	ARABIC LETTER YEH FINAL FORM
<y+,>	<UFEF3>	ARABIC LETTER YEH INITIAL FORM
<y+;>	<UFEF4>	ARABIC LETTER YEH MEDIAL FORM
<lM->	<UFEF5>	ARABIC LIGATURE LAM WITH ALEF WITH MADDA ABOVE ISOLATED FORM
<lH->	<UFEF7>	ARABIC LIGATURE LAM WITH ALEF WITH MADDA ABOVE FINAL FORM
<lH.>	<UFEF8>	ARABIC LIGATURE LAM WITH ALEF WITH HAMZA ABOVE ISOLATED FORM
<lH->	<UFEF9>	ARABIC LIGATURE LAM WITH ALEF WITH HAMZA ABOVE FINAL FORM
<lH.>	<UFEFA>	ARABIC LIGATURE LAM WITH ALEF WITH HAMZA BELOW ISOLATED FORM
<la->	<UFEFB>	ARABIC LIGATURE LAM WITH ALEF WITH HAMZA BELOW FINAL FORM
<la.>	<UFEFC>	ARABIC LIGATURE LAM WITH ALEF ISOLATED FORM
<H->	<U0023>	NUMBER SIGN
<!S>	<U0024>	DOLLAR SIGN
<@>	<U0040>	COMMERCIAL AT
<Oa>	<U0040>	COMMERCIAL AT
<!C>	<U00A2>	CENT SIGN
<L->	<U00A3>	POUND SIGN
<Xo>	<U00A4>	CURRENCY SIGN
<Y->	<U00A5>	YEN SIGN
<!B>	<U00A6>	BROKEN BAR
<So>	<U00A7>	SECTION SIGN
<7!>	<U00AC>	NOT SIGN
<9I>	<U00B6>	PILCROW SIGN
<_>	<U2500>	BOX DRAWINGS LIGHT HORIZONTAL
<=>	<U2501>	BOX DRAWINGS HEAVY HORIZONTAL
<_!>	<U2502>	BOX DRAWINGS LIGHT VERTICAL
<_V/>>	<U250C>	BOX DRAWINGS LIGHT DOWN AND RIGHT
<_V<w>	<U2510>	BOX DRAWINGS LIGHT DOWN AND LEFT
<_A/>>	<U2514>	BOX DRAWINGS LIGHT UP AND RIGHT
<_A<>	<U2518>	BOX DRAWINGS LIGHT UP AND LEFT
<_!/>>	<U251C>	BOX DRAWINGS LIGHT VERTICAL AND RIGHT
<_!<>	<U2524>	BOX DRAWINGS LIGHT VERTICAL AND LEFT
<_V->	<U252C>	BOX DRAWINGS LIGHT DOWN AND HORIZONTAL
<_A>	<U2534>	BOX DRAWINGS LIGHT UP AND HORIZONTAL
<_!->	<U253C>	BOX DRAWINGS LIGHT VERTICAL AND HORIZONTAL
<_ /> />>	<U2571>	BOX DRAWINGS LIGHT DIAGONAL UPPER RIGHT TO LOWER LEFT
<_ <>	<U2572>	BOX DRAWINGS LIGHT DIAGONAL UPPER LEFT TO LOWER RIGHT
<_ . /> />>	<U25E2>	BLACK LOWER RIGHT TRIANGLE
<_ . <>	<U25E3>	BLACK LOWER LEFT TRIANGLE
<_ d!>	<U266A>	EIGHTH NOTE

7 CONFORMANCE

7.1 FDCC-set

5878 A FDCC-set description is conforming to this Technical Report if it meets the
5879 requirements in clause 4.

5880
5881 **7.2 FDCC-set category**

5882
5883 Conformance can be claimed for a category description against each of the clauses 4.3
5884 thru 4.12, and then the requirements of clause 4.1 are also met, and a
5885 LC_IDENTIFICATION category as described in clause 4.2 is specified.

5886
5887 **7.3 Charmap**

5888
5889 A charmap description is conforming to this Technical Report if it meets the requirements
5890 in clause 5.

5891
5892 **7.4 Repertoiremap**

5893
5894 A repertoiremap description is conforming to this Technical Report if it meets the
5895 requirements in clause 6.

5896

5897
5898
5899
5900
5901
5902
5903
5904
5905
5906
5907
5908
5909
5910
5911
5912
5913
5914
5915
5916
5917
5918
5919
5920
5921
5922
5923
5924
5925
5926
5927
5928
5929
5930
5931
5932
5933
5934
5935
5936
5937
5938
5939
5940
5941
5942
5943
5944
5945
5946
5947
5948

Annex A (informative)

Differences from the ISO/IEC 9945-2 standard

This Technical Report originated from the locale and charmap specifications in the ISO/IEC 9945-2 POSIX shell and utilities standard, and it intends to be backwards compatible, so that what is conformant to that standard should also be conformant to this Technical Report.

A number of enhancements have been made and a number of restrictions have been lifted in comparison to the POSIX standard:

A.1 Restrictions removed

1. Dependence on specific meaning of the character NUL as termination of a string (from the C standard) has been removed, to cater for other programming languages than C.

A.2 Enhancements

1. A description of a "repertoiremap" definition was added to facilitate descriptions of FDCC-sets without charmaps, and also to provide binding from a FDCC-set using one set of character names to charmaps using another naming set.

2. The specific POSIX locale has been replaced with the "i18n" FDCC-set, defined on the repertoire on ISO/IEC 10646.

3. Transliteration support has been added in the LC_CTYPE category.

4. Terminology has been aligned with ISO/IEC TR 11017, especially the POSIX term "locale" has been changed to "FDCC-set".

5. A date escape format "%F" has been added for ISO 8601 dates, and another date escape format "%f" has been added for weekday number with Monday being the first day of the week.

6. Added to LC_MONETARY to accommodate differences between local and international formats:

int_p_cs_precedes
int_p_sep_by_space
int_n_cs_precedes
int_n_sep_by_space

7. Section symbols have been added via the "section-symbol" keyword in the LC_COLLATE category.

8. The "order_start" keyword has got an optional "section-symbol" identifier

9. The keywords "reorder-section-after" and "reorder-section_end" have been introduced to reorder sections.

10. Symbolic ellipses (both decimal and hexadecimal) has been introduced as a notation.

- 5949 11. The "print" CTYPE class includes automatically all "graph" characters.
5950
- 5951 12. The <Uxxxx> and <Uxxxxxxxx> notations have been introduced as predefined
5952 symbolic character names, together with a number of symbolic character names derived
5953 from POSIX and the Internet.
5954
- 5955 13. New categories LC_IDENTIFICATION, LC_XLITERATE, LC_NAME,
5956 LC_ADDRESS, and LC_TELEPHONE, have been introduced.
5957
- 5958 14. The LC_CTYPE has got support for new classes, via the new keywords class and
5959 map, which corresponds to the C standard library functions iswctype() and towctrans()
5960 respectively.
5961
- 5962 15. The "digit" keyword now supports digits for multiple scripts.
5963
- 5964 16. The LC_MONETARY category provides support for multiple currencies, such as the
5965 native currency and the Euro in some European countries.
5966
- 5967 17. The LC_TIME has got a number of enhancements to cater for alternate calendars, and
5968 timezone information may be given.
5969
- 5970 18. The charmap specification has been enhanced to support ISO 2022.

5971
5972
5973
5974
5975
5976
5977
5978
5979
5980
5981
5982
5983
5984
5985
5986
5987
5988
5989
5990
5991
5992
5993
5994
5995
5996
5997
5998
5999
6000
6001
6002
6003
6004
6005
6006
6007
6008
6009
6010
6011
6012
6013
6014
6015
6016
6017
6018
6019
6020
6021
6022

Annex B (informative)

Rationale

B.1 FDCC-set Rationale

The description of FDCC-sets is based on work performed in the UniForum Technical Committee Subcommittee on Internationalisation and POSIX. Wherever appropriate, keywords were taken from the C Standard or the ISO/IEC 9945-2:1993 POSIX standard. The C and POSIX term "locale" has been changed into the term "FDCC-set" from ISO/IEC TR 11017 to align with that specification.

The POSIX utility "localedef" compiles locale sources into object files. The "object" definitions need not be portable, as long as "source" definitions are. Strictly speaking, "source" definitions are portable only between applications using the same character set(s). Such "source" definitions can, if they use symbolic names only, easily be ported between systems using different code sets as long as the characters in the portable character set (ISO 646) have common values between the code sets; this is frequently the case in historical applications. Of course, this requires that the symbolic names used for characters outside the portable character set are identical between character sets.

To avoid confusion between an octal constant and a backreference, the octal, hexadecimal, and decimal constants must contain at least two digits. As single-digit constants are relatively rare, this should not impose any significant hardship. Each of the constants includes "two or more" digits to account for systems in which the byte size is larger than eight bits. For example, an ISO/IEC 10646 system that has defined 16-bit bytes may require six octal, four hexadecimal, and five decimal digits, for some coded characters.

As an international (ISO/IEC) Technical Report this Technical Report should follow the ISO/IEC guidelines, including the ISO/IEC TR 10176. This TR has a rule that characters outside the invariant part of ISO/IEC 646 should not be used in portable specifications. The backslash and the number-sign character are not in the invariant part. As far as general usage of these symbols, they are covered by the "grandfather clause" specifying previous practise in international standards and in the industry such as in specifications from The Open Group, but for newly defined interfaces, ISO has requested that specifications provide alternate representations, and this Technical Report then follows POSIX for backward compatibility. Consequently, while the default escape character remains the backslash, and the default comment character is the number-sign, applications are required to recognize alternative representations, identified in the applicable source text via the "escape_char" and "comment_char" keywords.

B.1.1 LC_IDENTIFICATION Rationale.

The LC_IDENTIFICATION category gives meta-information on the FDCC-set, such as who created it, and what is the level of conformance for each of the FDCC sets.

B.1.2 LC_CTYPE Rationale

6023 The LC_CTYPE category primarily is used to define the encoding-independent aspects of
6024 a character set, such as character classification. In addition, certain encoding-dependent
6025 characteristics are also defined for an application via the LC_CTYPE category. This
6026 Technical Report does not mandate that the encoding used in the FDCC-set is the same as
6027 the one used by the application, because an application may decide that it is advantageous
6028 to define a FDCC-set in a system-wide encoding rather than having multiple, logically
6029 identical FDCC-sets in different encodings, and to convert from the application encoding
6030 to the system-wide encoding on usage. Other applications could require encoding-depend-
6031 ent FDCC-sets. In either case, the LC_CTYPE attributes that are directly dependent on
6032 the encoding, such as "mb_cur_max" and the display width of characters, are not user-
6033 specifiable in a locale source, and are consequently not defined as keywords.

6034
6035 As the LC_CTYPE character classes are based on the C Standard character-class
6036 definition, the category does not support multicharacter elements. For instance, the
6037 German character <sharp-s> is traditionally classified as a lowercase letter. There is no
6038 corresponding uppercase letter; in proper capitalization of German text the <sharp-s> will
6039 be replaced by SS; i.e., by two characters. This kind of conversion is outside the scope of
6040 the "toupper" and "tolower" keywords.

6041
6042 The character classes "digit", "xdigit", "lower", "upper", and "space" have a set of
6043 automatically included characters. These only need to be specified if the character values
6044 (i.e. encoding) differs from the application default values. The definition of character class
6045 "digit" allows alternate digits (e.g., Hindi) to be specified here. The definition of character
6046 class "xdigit" requires that the characters included in character class "digit" are included
6047 here also, and allows for different symbols for the hexadecimal digits 10 through 15.

6048
6049 The "combining" and "combining-level3" classes are an IT-enablement of ISO/IEC 10646
6050 definitions of combining characters. These can be used to check identifiers for consistence
6051 with the guidelines given in TR 10176 annex A.

6052 6053 6054 **B.1.3 LC_COLLATE Rationale.**

6055
6056 The LC_COLLATE category governs the collation order in the FDCC-set, and may thus
6057 be useful for the processing of the ISO/IEC 14651 string ordering and comparison
6058 standard, the C Standard strxfrm() and strcoll() functions, as well as a number of ISO/IEC
6059 9945-2:1993 POSIX utilities.

6060
6061 The rules governing collation depends to some extent on the use. At least five different
6062 levels of increasingly complex collation rules can be distinguished:

- 6063
6064 (1) Byte/machine code order. This is the historical collation order in the UNIX
6065 system and many proprietary operating systems. Collation is here done
6066 character by character, without any regard to context. The primary virtue is that
6067 it usually is quite fast, and also completely deterministic; it works well when
6068 the native machine collation sequence matches the user expectations.
- 6069 (2) Character order. On this level, collation is also done character by character,
6070 without regard to context. The order between characters is, however, not deter-
6071 mined by the code values, but on the user's expectations of the correct order
6072 between characters. In addition, such a (simple) collation order can specify that
6073 certain characters collate equal (e.g., upper and lowercase letters).
- 6074 (3) String ordering. On this level, entire strings are compared based on relatively

- 6075 straightforward rules. At this level, several "passes" may be required to deter-
 6076 mine the order between two strings. Characters may be ignored in some passes,
 6077 but not in others; the strings may be compared in different directions; and
 6078 simple string substitutions may be made before strings are compared. This level
 6079 is best described as "dictionary" ordering; it is based on the spelling, not the
 6080 pronunciation, or meaning, of the words.
- (4) 6081 Text search ordering. This is a further refinement of the previous level, best de-
 6082 scribed as "telephone book ordering"; some common homonyms (words spelled
 6083 differently but with same pronunciation) are collated together; numbers are
 6084 collated as if spelled with words, and so on.
- (5) 6085 Semantic level ordering. Words and strings are collated based on their meaning;
 6086 entire words (such as "the") are eliminated, the ordering is not deterministic.
 6087 This may requires special software, and is highly dependent on the intended
 6088 use.

6090 While the historical collation order formally is at level 1, for the English language it
 6091 corresponds roughly to elements at level 2. The user expects to see the output from the
 6092 "ls" utility sorted very much as it would be in a dictionary. While telephone book ordering
 6093 would be an optimal goal for standard collation, this was ruled out as the order would be
 6094 language dependent. Furthermore, a requirement was that the order must be determined
 6095 solely from the text string and the collation rules; no external information (e.g., "pronu-
 6096 nciation dictionaries") could be required.

6098 As a result, the goal for the collation support is at level 3. This also matches the re-
 6099 quirements for the Canadian collation order standard, as well as other, known collation
 6100 requirements for alphabetic scripts. It specifically rules out collation based on pronun-
 6101 ciation rules, or based on semantic analysis of the text. The syntax for the LC_COLLATE
 6102 category source is the result of a cooperative effort between representatives for many
 6103 countries and organizations working with international issues, such as UniForum, The
 6104 Open Group, The Unicode Consortium Inc. and ISO, and it meets the requirements for
 6105 level 3, and has been verified to produce the correct result with examples based on
 6106 Canadian and Danish collation order.

6108 The directives that can be specified in an operand to the order_start keyword are based on
 6109 the requirements specified in several proposed standards and in customary use. The
 6110 following is a rephrasing of rules defined for "lexical ordering in English and French" by
 6111 the Canadian Standards Association (text in brackets is rephrased):

- (1) 6113 Once special characters (punctuation) have been removed from original strings,
 6114 the ordering is determined by scanning forward (left to right) [disregarding case
 6115 and diacriticals].
- (2) 6116 In case of equivalence, special characters are once again removed from original
 6117 strings and the ordering is determined scanning backward (starting from the
 6118 rightmost character of the string and back), character by character, (disregarding
 6119 case but considering diacriticals).
- (3) 6120 In case of repeated equivalence, special characters are removed again from
 6121 original strings and the ordering is determined scanning forward, character by
 6122 character, (considering both case and diacriticals).
- (4) 6123 If there is still an ordering equivalence after rules (1) through (3) have been
 6124 applied, then only special characters and the position they occupy in the string
 6125 are considered to determine ordering. The string that has a special character in
 6126 the lowest position comes first. If two strings have a special character in the

6127 same position, the character [with the lowest collation value] comes first. In
6128 case of equality, the other special characters are considered until there is a
6129 difference or all special characters have been exhausted.
6130

6131 It is estimated that the Technical Report covers the mechanisms to specify data to cover
6132 the requirements for all European languages, and Cyrillic and Middle Eastern scripts.
6133

6134 The Far East (particularly Japanese/Chinese) collations are often based on contextual
6135 information. In Japan, collations of strings containing CJK characters (ideograms) are
6136 often done considering some related information such as pronunciation, which needs a
6137 bulk dictionary (and some common sense). Such collation, in general, falls outside the
6138 desired goal of this Technical Report, and this Technical Report can support only a
6139 restricted of collations used in Japan. There are, however, several other collation rules
6140 (stroke/radical, or "most common pronunciation") which can be supported with the
6141 mechanism described here. Previous drafts contained a substitute statement, which
6142 performed a regular expression style replacement before string compares. It has been
6143 withdrawn based on balloter objections that it was not required for the types of ordering
6144 this Technical Report is aimed at.
6145

6146 The character (and collating element) order is defined by the order in which characters and
6147 elements are specified between the order_start and order_end keywords. This character
6148 order is used in range expressions in regular expressions. Weights assigned to the charac-
6149 ters and elements define the collation sequence; in the absence of weights, the character
6150 order is also the collation sequence.
6151

6152 The position keyword was introduced to provide the capability to consider, in a compare,
6153 the relative position of non-IGNORED characters. As an example, consider the two strings
6154 "o-ring" and "or-ing". Assuming the hyphen is IGNORED on the first pass, the two strings
6155 will compare equal, and the position of the hyphen is immaterial. On second pass, all
6156 characters except the hyphen are IGNORED, and in the normal case the two strings would
6157 again compare equal. By taking position into account, the first collates before the second.
6158

6159 This Technical Report adds a number of facilities over the ISO/IEC 9945:1993 POSIX
6160 standard, especially in the support for the ISO/IEC 10646 UCS character set. These
6161 extended facilities are in alignment with the ISO/IEC 14651 sorting standard. In addition
6162 to the facilities provided in ISO/IEC 14651, this specification contains mechanisms to put
6163 data into a FDCC-set environment, and has added facilities to sort sections differently, has
6164 facilities to reuse FDCC-sets in different notations via the "equivalence-symbol" keyword
6165 and tables.
6166

6167 **B.1.3.1 "reorder-after" rationale** 6168

6169 Much work has been done on FDCC-sets, making them quite general. The ISO/IEC 9945-
6170 2:1993 POSIX standard introduced a "copy" command for all categories of the POSIX
6171 locale. This is useful for many purposes and it ensures that two FDCC-sets are equivalent
6172 for this category. A further step in building on previous FDCC-set work is defined in this
6173 Technical Report.
6174

6175 Collating sequences often vary a bit from country to country, and from language to
6176 language, but generally much of the collating sequence is the same. For example the
6177 Danish sequence is for the most part the same as the German or English collation, but for
6178 about a dozen letters it differs. The same can be said for Swedish or Hungarian: generally

6179 the Latin collating sequence is the same, but a few characters are different.

6180
6181 This Technical Report defines a FDCC-set defined on the character repertoire of the
6182 ISO/IEC 10646 standard, in a character set independent way. The intention is that some of
6183 the information from this FDCC-set will be acceptable in many cultures, and that it can
6184 serve as the basis for modifications in other cultures, to obtain a culturally acceptable
6185 specification. Using the "reorder-after" construct will also help improve the overview of
6186 what the changes really are for implementers and other users.

6187
6188 An example of the use of the "reorder-after" construct is the following. A default
6189 international ordering for the Latin alphabet may be adequate for Danish, with the
6190 exception of the collation rules for the letters Û, ü, Æ, æ, Ä, ä, Ø, ø, Ö, ö, Å and å. By
6191 applying the "reorder-after" construct, the Danish specification can be made more easily
6192 by copying and reordering the existing international specification, rather than specifying
6193 collation parameters for all Latin letters (with or without diacritics). There is no obligation
6194 for Denmark to take this approach, but the "reorder-after" construct provides the
6195 mechanism for doing so if it is deemed desirable.

6196
 6197
 6198
 6199
 6200
 6201
 6202
 6203
 6204
 6205
 6206
 6207
 6208
 6209
 6210
 6211
 6212
 6213
 6214
 6215
 6216
 6217
 6218
 6219
 6220
 6221
 6222
 6223
 6224
 6225
 6226
 6227
 6228
 6229
 6230
 6231
 6232
 6233
 6234
 6235
 6236
 6237
 6238
 6239
 6240
 6241
 6242
 6243
 6244
 6245
 6246
 6247
 6248
 6249

B.1.3.2 awk script for "reorder-after" construct

A script has been written in the "awk" language defined in the POSIX standard ISO/IEC 9945-2 to implement the "reorder-after" construct. It functions as follows: It reads all of the FDCC-set and if in the LC_COLLATE category, it processes the line, else it just outputs the line. For the LC_COLLATE category it reads the lines and puts it into a double linked list of strings identified by a line number; at the end of the LC_COLLATE category all the lines are output. If the line is a "copy" keyword and it reads the file referenced, extracting the LC_COLLATE section of the file in to the list of strings. If the line is a "reorder-after" keyword, it sets a pointer to be the line number of the symbol to of the "reorder-after" keyword. If the line is part of the "reorder-after" specification, it is entered into the double linked list at this point, and the previous entry in the double linked list for the <collation-element> is removed from the list. A "reorder-end" keyword terminates the reordering.

```

BEGIN { comment = "%"; back[0]= follow[0] = 0; }
/LC_COLLATE/ { coll=1 }
/END LC_COLLATE/ { coll=0; for (lnr= 1; lnr; lnr= follow[lnr]) print c-
ont[lnr] }

{ if (coll == 0) print $0 ;
  else { if ($1 == "copy")    {
        file = $2
        while (getline < file )
        if ( $1 == "LC_COLLATE" ) copy_lc = 1
        else if ( $1 == "END" && $2 == "LC_COLLATE" ) copy_lc = 0
        else if (copy_lc) {
            lnr++
            follow[lnr-1] = lnr; back [ lnr ] = lnr-1
            cont[lnr] = $0; symb[ $1 ] = lnr
        }
        close (file )
    }
  else if ($1 == "reorder-after") { ra=1 ; after = symb [ $2 ] }
  else if ($1 == "reorder-end") ra = 0
  else {
    lnr++
    if (ra) follow [ lnr ] = follow [ after ]
    if (ra) back [ follow [ after ] ] = lnr
    follow[after] = lnr; back [ lnr ] = after
    cont[lnr] = $0
    if ( ra && $1 != comment && $1 != " ) {
      old = symb [ $1 ];
      follow [ back [ old ] ] = follow [ old ];
      back [ follow [ old ] ] = back [ old ];
      symb[ $1 ] = lnr;
    }
    after = lnr
  }
}
}

```

```

6250 B.1.3.3 Sample FDCC-set specification for Danish
6251
6252 escape_char /
6253 comment_char %
6254 repertoiremap "i18nrep"
6255 charset "ISO_8859-1:1987"
6256 % Distribution and use is free, also
6257 % for commercial purposes.
6258
6259 LC_VERSION
6260 title "Danish language FDCC-set for Denmark"
6261 source "Danish Standards Association"
6262 address "Kollegievej 6, DK-2920 Charlottenlund, Danmark"
6263 contact "Keld Simonsen"
6264 email "Keld.Simonsen@dkuug.dk"
6265 tel "+45 - 3996-6101"
6266 fax "+45 - 3996-6202"
6267 language "da"
6268 territory "DK"
6269 revision "4.2"
6270 date "1997-12-22"
6271
6272 category i18n:2000;LC_IDENTIFICATION
6273 category i18n:2000;LC_CTYPE
6274 category i18n:2000;LC_COLLATE
6275 category i18n:2000;LC_TIME
6276 category posix:1993;LC_NUMERIC
6277 category i18n:2000;LC_MONETARY
6278 category posix:1993;LC_MESSAGES
6279 category i18n:2000;LC_XLITERATE
6280 category i18n:2000;LC_NAME
6281 category i18n:2000;LC_ADDRESS
6282 category i18n:2000;LC_TELEPHONE
6283
6284 END LC_VERSION
6285
6286 LC_CTYPE
6287 copy "i18n"
6288 END LC_CTYPE
6289
6290 LC_COLLATE
6291 % The ordering algorithm is in accordance
6292 % with Danish Standard DS 377 (1980)
6293 % and the Danish Orthography Dictionary
6294 % (Retskrivningsordbogen, 2. udgave, 1996).
6295 % It is also in accordance with
6296 % Greenlandic orthography.
6297
6298 collating-element <A-A> from "<A><A>"
6299 collating-element <A-a> from "<A><a>"
6300 collating-element <a-A> from "<a><A>"
6301 collating-element <a-a> from "<a><a>"
6302 collating-symbol <SPECIAL>
6303 copy i18n
6304 reorder-after <CAPITAL>
6305 <CAPITAL>
6306 <CAPITAL-SMALL>
6307 <SMALL-CAPITAL>
6308 <SMALL>
6309 reorder-after <q8>
6310 <kk> <Q>;<SPECIAL>;<SMALL>;IGNORE
6311 reorder-after <t8>
6312 <TH> "<T><H>";"<TH><TH>";"<CAPITAL><CAPITAL>";IGNORE
6313 <th> "<T><H>";"<TH><TH>";"<SMALL><SMALL>";IGNORE
6314 reorder-after <y8>
6315 % <U:> and <U"&> are treated as <Y> in Danish
6316 <U:> <Y>;<U:>;<CAPITAL>;IGNORE
6317 <u:> <Y>;<U:>;<SMALL>;IGNORE
6318 <U"&> <Y>;<U"&>;<CAPITAL>;IGNORE
6319 <u"&> <Y>;<U"&>;<SMALL>;IGNORE

```

```

6320 reorder-after <z8>
6321 % <AE> is a separate letter in Danish
6322 <AE> <AE>;<NONE>;<CAPITAL>;IGNORE
6323 <ae> <AE>;<NONE>;<SMALL>;IGNORE
6324 <AE'> <AE>;<ACUTE>;<CAPITAL>;IGNORE
6325 <ae'> <AE>;<ACUTE>;<SMALL>;IGNORE
6326 <A3> <AE>;<MACRON>;<CAPITAL>;IGNORE
6327 <a3> <AE>;<MACRON>;<SMALL>;IGNORE
6328 <A:> <AE>;<SPECIAL>;<CAPITAL>;IGNORE
6329 <a:> <AE>;<SPECIAL>;<SMALL>;IGNORE
6330 % <O//> is a separate letter in Danish
6331 <O//> <O//>;<NONE>;<CAPITAL>;IGNORE
6332 <o//> <O//>;<NONE>;<SMALL>;IGNORE
6333 <O//'> <O//>;<ACUTE>;<CAPITAL>;IGNORE
6334 <o//'> <O//>;<ACUTE>;<SMALL>;IGNORE
6335 <O:> <O//>;<DIAERESIS>;<CAPITAL>;IGNORE
6336 <o:> <O//>;<DIAERESIS>;<SMALL>;IGNORE
6337 <O"> <O//>;<DOUBLE-ACUTE>;<CAPITAL>;IGNORE
6338 <o"> <O//>;<DOUBLE-ACUTE>;<SMALL>;IGNORE
6339 % <AA> is a separate letter in Danish
6340 <AA> <AA>;<NONE>;<CAPITAL>;IGNORE
6341 <aa> <AA>;<NONE>;<SMALL>;IGNORE
6342 <A-A> <AA>;<A-A>;<CAPITAL>;IGNORE
6343 <A-a> <AA>;<A-A>;<CAPITAL-SMALL>;IGNORE
6344 <a-A> <AA>;<A-A>;<SMALL-CAPITAL>;IGNORE
6345 <a-a> <AA>;<A-A>;<SMALL>;IGNORE
6346 <AA'> <AA>;<AA'>;<CAPITAL>;IGNORE
6347 <aa'> <AA>;<AA'>;<SMALL>;IGNORE
6348 reorder-end
6349 END LC_COLLATE
6350
6351 LC_MONETARY
6352 int_curr_symbol " <D><K><K><SP> "
6353 currency_symbol " <k><r> "
6354 mon_decimal_point " <,> "
6355 mon_thousands_sep " <.> "
6356 mon_grouping 3;3
6357 positive_sign " "
6358 negative_sign " <-> "
6359 int_frac_digits 2
6360 frac_digits 2
6361 p_cs_precedes 1
6362 p_sep_by_space 2
6363 n_cs_precedes 1
6364 n_sep_by_space 2
6365 p_sign_posn 4
6366 n_sign_posn 4
6367 END LC_MONETARY
6368
6369 LC_NUMERIC
6370 decimal_point " <,> "
6371 thousands_sep " <.> "
6372 grouping 3;3
6373 END LC_NUMERIC
6374
6375 LC_TIME
6376 abday " <m><a><n> " ; /
6377 " <t><i><r> " ; " <o><n><s> " ; /
6378 " <t><o><r> " ; " <f><r><e> " ; /
6379 " <l><o//><r> " ; " <s><o//><n>
6380 day " <m><a><n><d><a><g> " ; /
6381 " <t><i><r><s><d><a><g> " ; /
6382 " <o><n><s><d><a><g> " ; /
6383 " <t><o><r><s><d><a><g> " ; /
6384 " <f><r><e><d><a><g> " ; /
6385 " <l><o//><r><d><a><g> " /
6386 " <s><o//><n><d><a><g> " ;
6387 week 7;19971201;4
6388 abmon " <j><a><n> " ; " <f><e><b> " ; /
6389 " <m><a><r> " ; " <a><p><r> " ; /

```

```

6390         "<m><a><j>" ; "<j><u><n>" ; /
6391         "<j><u><l>" ; "<a><u><g>" ; /
6392         "<s><e><p>" ; "<o><k><t>" ; /
6393         "<n><o><v>" ; "<d><e><c>"
6394 mon      "<j><a><n><u><a><r>" ; /
6395         "<f><e><b><r><u><a><r>" ; /
6396         "<m><a><r><t><s>" ; /
6397         "<a><p><r><i><l>" ; /
6398         "<m><a><j>" ; /
6399         "<j><u><n><i>" ; /
6400         "<j><u><l><i>" ; /
6401         "<a><u><g><u><s><t>" ; /
6402         "<s><e><p><t><e><m><b><e><r>" ; /
6403         "<o><k><t><o><b><e><r>" ; /
6404         "<n><o><v><e><m><b><e><r>" ; /
6405         "<d><e><c><e><m><b><e><r>"
6406 d_t_fmt   "<%><a><SP><%><F><SP><%><T><SP><%><Z>"
6407 d_fmt     "<%><O><d><.><SP><%><B><SP><%><Y>"
6408 atl_digits "<0><.>;<1><.>;<2><.>;<3><.>;<4><.>; /
6409         <5><.>;<6><.>;<7><.>;<8><.>;<9><.>; /
6410         <1><0><.>;<1><1><.>;<1><2><.>;<1><3><.>;<1><4><.>; /
6411         <1><5><.>;<1><6><.>;<1><7><.>;<1><8><.>;<1><9><.>; /
6412         <2><0><.>;<2><1><.>;<2><2><.>;<2><3><.>;<2><4><.>; /
6413         <2><5><.>;<2><6><.>;<2><7><.>;<2><8><.>;<2><9><.>; /
6414         <3><0><.>;<3><1><.>"
6415 t_fmt     "<%><T>"
6416 am_pm    ";" ; ""
6417 t_fmt_ampm ""
6418 timezone "<C><E><T><-><1><C><E><T><SP><D><S><T><,><M><3><.><5><.><0>/
6419         <,><M><1><0><.><5><.><0>"
6420 END LC_TIME
6421
6422 LC_MESSAGES
6423 yesexpr  "<<(><1><J><j><Y><y><)/>><.><*>"
6424 noexpr   "<<(><0><N><n><)/>><.><*>"
6425 END LC_MESSAGES
6426
6427 LC_NAME
6428 name_fmt "<%><p><%><t><%><g><%><t><%><m><%><t><%><f>"
6429 name_gen ""
6430 name_mr  "<h><r>"
6431 name_mrs "<f><r><u>"
6432 name_miss "<f><r><o>/><k><e><n>"
6433 name_ms  "<f><r>"
6434 END LC_NAME
6435
6436 LC_ADDRESS
6437 country_name "<D><a><n><m><a><r><k>"
6438 country_post "<D><K>"
6439 lang_ab      "<d><a>"
6440 lang_term   "<d><a><n>"
6441 postal_fmt  "<%><a><%><N><%><f><%><N><%><d><%><N><%><b><%><N><%>/
6442         <%><s><SP><%><h><SP><%><e><SP><%><r><%><N>/
6443         <%><C><-><%><z><SP><%><T><%><N><%><C><%><N>"
6444 END LC_ADDRESS
6445
6446 LC_TELEPHONE
6447 tel_int_fmt "<+><%><c><SP><%><a><SP><%><l>"
6448 tel_dom_fmt "<%><l>"
6449 int_select "<0><0>"
6450 int_prefix "<4><5>"
6451 END LC_TELEPHONE

```

6453 B.1.4 LC_MONETARY Rationale.

6454

6455 The currency symbol does not appear in LC_MONETARY because it is not defined in the
6456 C Standard's C locale. The C Standard limits the size of decimal points and thousands
6457 delimiters to single-byte values. In FDCC-sets based on multibyte coded character sets this

6458 cannot be enforced, obviously; this Technical Report does not prohibit such characters, but
6459 makes the behaviour unspecified (in the text "In contexts where other standards . . .").
6460

6461 The grouping specification is based on, but not identical to, the C Standard . The "-1"
6462 signals that no further grouping is performed, the equivalent of (CHAR_MAX) in the C
6463 Standard).
6464

6465 The FDCC-set definition is an extension of the C Standard localeconv() specification. In
6466 particular, rules on how currency_symbol is treated are extended to also cover int_
6467 curr_symbol, and p_set_by_space and n_sep_by_space have been augmented with the
6468 value 2, which places a space between the sign and the symbol (if they are adjacent;
6469 otherwise it should be treated as a 0). The following table shows the result of various
6470 combinations:
6471

		p_sep_by_space			
		2	1	0	
6476	p_cs_precedes = 1	p_sign_posn = 0	(\$ 1.25)	(\$ 1.25)	(\$1.25)
6477		p_sign_posn = 1	+ \$1.25	+\$ 1.25	+\$1.25
6478		p_sign_posn = 2	\$1.25 +	\$ 1.25+	\$1.25+
6479		p_sign_posn = 3	+ \$1.25	+\$ 1.25	+\$1.25
6480		p_sign_posn = 4	\$ +1.25	\$+ 1.25	\$+1.25
6481					
6482	p_cs_precedes = 0	p_sign_posn = 0	(1.25 \$)	(1.25 \$)	(1.25\$)
6483		p_sign_posn = 1	+1.25 \$	+1.25 \$	+1.25\$
6484		p_sign_posn = 2	1.25\$ +	1.25 \$+	1.25\$+
6485		p_sign_posn = 3	1.25+ \$	1.25 +\$	1.25+\$
6486		p_sign_posn = 4	1.25\$ +	1.25 \$+	1.25\$+
6487					
6488					

6489 The following is an example of the interpretation of the mon_grouping keyword.
6490 Assuming that the value to be formatted is 123456789 and the mon_thousands_sep is "",
6491 then the following table shows the result. The third column shows the equivalent C
6492 Standard string that would be used to accommodate this grouping. It is the responsibility
6493 of the utility to perform mappings of the formats in this clause to those used by language
6494 bindings such as the C Standard .
6495

6497	Mon_grouping	Formatted Value	C String
6498	3;-1	123456'789	"\3\177"
6499	3	123'456'789	"\3"
6500	3;2;-1	1234'56'789	"\3\2\177"
6501	3;2	12'34'56'789	"\3\2"
6502	-1	123456789	"177"
6503			

6504 In these examples, the octal value of (CHAR_MAX) is 177.
6505

6506 The multiple currency support is specified such that a FDCC-set can be used without
6507 change during the transition period in a static environment. For example in the case of the
6508 Euro currency as being employed in a number of European countries, there is no need to
6509 change the FDCC-set when shifting from one currency to two concurrent currencies; and

6510 there is no need to change FDCC-set, when changing to the Euro as the only currency.
 6511 Also the same application call can be made to be valid for countries with a single
 6512 currency and countries with dual currencies. The specifications can also be used without
 6513 change of the FDCC-set on an installation, when converting from one national currency to
 6514 another, for example when removing some zeroes to form a new currency.
 6515

6516 The following example illustrates the support for multiple currencies; the example is for
 6517 the Euro in Germany:
 6518

```

6519     LC_MONETARY
6520     valid_from           " ";                "19990101"
6521     valid_to            "20020630";          " "
6522     conversion_rate     1;                    195/100
6523     int_curr_symbol     "<D><E><M><SP>";      "<E><U><R><SP>"
6524     currency_symbol     "<D><M>";          "<E><U><R>"
6525     mon_decimal_point   "<,>"
6526     mon_thousands_sep "<.>"
6527     mon_grouping        3;3
6528     positive_sign       " "
6529     negative_sign       "<->"
6530     int_frac_digits     2;                    2
6531     frac_digits         2;                    2
6532     p_cs_precedes       1;                    1
6533     p_sep_by_space      2;                    2
6534     n_cs_precedes       1;                    1
6535     n_sep_by_space      2;                    2
6536     p_sign_posn         4;                    4
6537     n_sign_posn         4;                    4
6538
6539     END LC_MONETARY
  
```

6540 **B.1.5 LC_NUMERIC Rationale.**

6541 See the rationale for LC_MONETARY (B1.3) for a description of the behaviour of
 6542 grouping.
 6543

6544 **B.1.6 LC_TIME Rationale.**

6545 The LC_TIME descriptions of abday, day, and abmon imply a Gregorian style calendar
 6546 (7-day weeks, 12-month years, leap years, etc.). Other calendars can be supported, for
 6547 example calendars with a fixed week length.
 6548

6549 In some FDCC-sets the field descriptors for weekday and month names will be given with
 6550 an initial small letter. Programs using these fields may need to adjust the capitalization if
 6551 the output is going to be used at the beginning of a sentence.
 6552

6553 The field descriptors corresponding to the optional keywords consist of a modifier
 6554 followed by a traditional field descriptor (for instance %Ex). If the optional keywords are
 6555 not supported by the application or are unspecified for the current FDCC-set, these field
 6556 descriptors are treated as the traditional field descriptor. For instance, assume the
 6557 following keywords:
 6558

```

6559     alt_digits "0th";"1st";"2nd";"3rd";"4th";"5th";"6th";"7th";"8th";"9th";"10th"
6560     d_fmt      "The %Od day of %B in %Y"
  
```

6561 On 7/4/1776, the %x field descriptor would result in "The 4th day of July in 1776," while
 6562 7/14/1789 would come out as "The 14 day of July in 1789." It can be noted that the above
 6563 example is for illustrative purposes only; the %o modifier is primarily intended to provide
 6564

6568 for Kanji or Hindi digits in date formats. While it is clear that an alternate year format is
6569 required, there is no consensus on the format or the requirements. As a result, while these
6570 keywords are reserved, the details are left unspecified. It is expected that National
6571 Standards Bodies will provide specifications.

6572
6573

6574 **B.1.7 LC_MESSAGES Rationale.**

6575
6576 The LC_MESSAGES category is described in clause 4 as affecting the language used by
6577 utilities for their output. The mechanism used by the application to accomplish this, other
6578 than the responses shown here in the FDCC-set definition, is not specified by this version
6579 of this Technical Report. The ISO internationalization working group is developing an
6580 interface that would allow applications (and, presumably some of the standard utilities) to
6581 access messages from various message catalogs, tailored to a user's LC_MESSAGES
6582 value.

6583
6584

6585 **B.1.8 LC_XLITERATE Rationale.**

6586
6587 Transliteration is often language dependent, transliterating one specific language to another
6588 specific language. For example transliteration from Russian to English, and from Serbian
6589 to German would normally be quite different, although the same repertoire of characters
6590 would be transliterated. Even transliteration of two languages using the same script into
6591 one language (for example from Russian to Danish and from Serbian to Danish), or
6592 transliteration of the same language (for example Russian into English or German) may be
6593 different. The language to be transliterated to is identified with the FDCC-set, which may
6594 also be used to identify a specific language to be transliterated from. Transliteration may
6595 also be to a specific repertoire of characters, determined for example by limitations of
6596 displaying equipment, or what the user can intelligibly read. The capabilities here allows
6597 for multiple fallback, so that the specification can be valid for all target character
6598 repertoires, eliminating the need for specific data for each target repertoire.

6599
6600

6601 **B.1.9 LC_NAME Rationale.**

6602 The LC_NAME category gives information to prepare a text for addressing a person, for
6603 example as a part of a postal address on an envelope, or as a saluting line in a letter.
6604 The information is intended to be given to an API that has the various naming information
6605 as parameters and yields a formatted string as the return value.

6606
6607 The "profession" entry is intended for either the general profession of the person in
6608 question, or the job title, for use in letters or as part of the address on an envelope.

6609
6610

6611 **B.1.10 LC_ADDRESS Rationale.**

6612
6613 The LC_ADDRESS category gives information to prepare a text for writing an address,
6614 for example as a part of a postal address on an envelope. The information is intended to
6615 be given to an API that has the various address information as parameters and yields a
6616 formatted string as the return value.

6617
6618

6619 **B.1.11 LC_TELEPHONE Rationale.**

6620
6621 The LC_TELEPHONE category gives information to prepare a text for writing a telephone
6622 number. The information is intended to be given to an API that has the various
6623 information on a telephone number as parameters and yields a formatted string as the
6624 return value. Both an international and a domestic formatting possibility is available.
6625

6626 **B.2 Character Set Rationale.**

6627
6628 This Technical Report poses no requirement that multiple character sets or code sets be
6629 supported, leaving this as a marketing differentiation for implementors. Although multiple
6630 charmaps are supported, it is the responsibility of the application to provide the file(s); if
6631 only one is provided, only that one will be accessible.
6632

6633
6634 The character set description text provides the capability to describe character set attributes
6635 (such as collation order or character classes) independent of character set encoding, and
6636 using only the characters in the portable character set. This makes it possible to create
6637 "generic" FDCC-set source texts for all code sets that share the portable character set
6638 (such as the ISO/IEC 8859 family or IBM Extended ASCII).
6639

6640 Applications are free to describe more than one code set in a character set description text.
6641 For example, if an application defines ISO/IEC 8859-1 as the primary code set, and
6642 ISO/IEC 8859-2 as an alternate set, with each character from the alternate code set
6643 preceded in data by a shift code, a character set description text could contain a complete
6644 description of the primary set and those characters from the secondary that are not
6645 identical, the encoding of the latter including the shift code.
6646

6647 Applications are free to choose their own symbolic names, as long as the names identified
6648 by this Technical Report are also defined; this provides support for already existing
6649 "character names".
6650

6651 The charmap was introduced to resolve problems with the portability of, especially,
6652 FDCC-set sources. While the portable character set (in Table 1) is a constant across all
6653 FDCC-sets for a particular application, this is not true for the extended character set.
6654 However, the particular coded character set used for an application does not necessarily
6655 imply different characteristics or collation: on the contrary, these attributes should in many
6656 cases be identical, regardless of codeset. The charmap provides the capability to define a
6657 common FDCC-set definition for multiple codesets (the same FDCC-set source can be
6658 used for codesets with different extended characters; the ability in the charmap to define
6659 "empty" names allows for characters missing in certain codesets).
6660

6661 In addition, some implementors have expressed an interest in using the charmap to define
6662 certain other characteristics of codesets, such as the <mb_cur_max> value for the
6663 particular codeset. (Note that <mb_cur_max> has to be equal to or lower than the C
6664 Standard {MB_LEN_MAX}, which is the application limit). Such extensions are not
6665 described here; but may be added in a later revision of this Technical Report.
6666

6667 The <escape_char> declaration was added at the request of the international community to
6668 ease the creation of portable charmaps on terminals not implementing the default
6669 backslash escape. (This approach was adopted because this is a new interface invented by
6670 ISO/IEC 9945-2:1993 POSIX. Historical interfaces, such as the shell command language
6671 and awk, have not been modified to accommodate this type of terminal.)

6672 The octal number notation was selected to match those of POSIX "awk" and "tr" utilities
6673 and is consistent with that used by the POSIX localedef utility.

6674
6675 The charmap capability implements a facility available at some X/Open compatible
6676 applications. Its prime virtue is to support "generic" collation sequence source definitions.
6677 An implementor or an applications developer can produce a template definition that can be
6678 used to produce several codeset-dependent "compiled" FDCC-set definitions. The facility
6679 also removes any dependency in many source definitions on characters outside the
6680 character set defined in this clause.

6681
6682 The charmap allows specification of more than one encoding of a character. This allows
6683 for encodings that can encode items in more than one way. For example, an item can be
6684 encoded once as a fully composed character and again as a base character plus combining
6685 character. This would allow either representation to be recognized. As only the first
6686 occurrence of the character may be output, this technique could be used to normalize a
6687 character stream.

6688
6689 The ISO 2022 support introduced gives the possibility to refer other definitions via
6690 charmaps, so the full encoding does not have to be replicated. It supports shifting with G0,
6691 G1, G2 and G3 sets, and also general shifting of coded character sets via escape
6692 sequences.

6693
6694

6695 **B.3 Repertoiremap Rationale.**

6696
6697 The repertoiremap was introduced to make FDCC-sets independent of the availability of
6698 charmaps. With the repertoiremap it is possible to use a FDCC-set encoded with one set
6699 of symbolic character names, together with charmaps with other symbolic character
6700 naming schemes, provided there are repertoiremaps available for both naming schemes.

6701
6702 Repertoiremaps are also useful to describe repertoires of characters, to be used for
6703 example for transliteration.

Annex C (informative)

BNF Grammar

C.1 BNF Syntax Rules

The syntax used here is near to ISO/IEC 14977, but "_" is allowed in identifiers, and comma is not used as concatenator, as the items are just concatenated.

Definitions between <angle brackets> make use of terms not defined in this BNF syntax, and assume general English usage.

Other conventions:

* means 0 or more repetitions of a token.

+ means one or more repetitions of a token

Brackets [] indicate optional occurrence of a token.

Comments start with a % on a separate line.

There may be more specifications in the normative text that describes restrictions on the grammar.

C.2 Grammar for FDCC-sets

```
% The following is the overall FDCC-set grammar
FDCC_set_definition = [ global_statement* ] category+ ;
global_statement  = 'escape_char' SP char_symbol EOL
                  | 'comment_char' SP char_symbol EOL
                  | 'repertoiremap' SP quoted_string EOL
                  | 'charmap' SP quoted_string EOL ;
category          = lc_identification | lc_ctype | lc_collate
                  | lc_monetary | lc_numeric | lc_time
                  | lc_messages | lc_xliterate | lc_telephone
                  | lc_name | lc_address ;

% The following is the LC_IDENTIFICATION category grammar
lc_ident          = ident_head ident_keyword* ident_tail
                  | ident_head copy_FDCC_set ident_tail ;
ident_head        = 'LC_IDENTIFICATION' EOL ;
ident_keyword     = ident_keyword_string SP quoted_string EOL ;
ident_keyword_string = 'title' | 'source' | 'address' | 'contact'
                      | 'email' | 'tel' | 'fax' | 'language'
                      | 'territory' | 'audience' | 'application'
                      | 'abbreviation' | 'revision' | 'date' ;
ident_tail        = 'END' SP 'LC_IDENTIFICATION' EOL ;

% The following is the LC_CTYPE category grammar
lc_ctype          = ctype_head ctype_keyword* ctype_tail
                  | ctype_head copy_FDCC_set ctype_tail ;
ctype_head        = 'LC_CTYPE' EOL ;
ctype_keyword     = charclass_keyword SP charclass_list EOL
                  | charconv_keyword SP charconv_list EOL
                  | 'width' SP width_list EOL ;
charclass_keyword = 'upper' | 'lower' | 'alpha' | 'digit' |
                  | 'alnum' | 'punct' | 'xdigit' | 'space' |
                  | 'print' | 'graph' | 'blank' | 'cntrl' |
                  | 'outdigit'
                  | 'class' charclass_name semicolon ;
charclass_name    = '"combining"' | '"combining_level3"'
                  | "' identifier "' ;
```

```

6766 charclass_list = charclass_list semicolon char_symbol
6767 | charclass_list semicolon ctype_abs_ellipsis
6768 semicolon char_symbol
6769 | charclass_list semicolon charsymbol
6770 ctype_symbolic_ellipses charsymbol
6771 | char_symbol ;
6772 width_list = charclass_list ':' number
6773 | width_list semicolon width_list ;
6774 charconv_keyword = 'toupper' | 'tolower'
6775 | 'map' '' identifier '' semicolon ;
6776 charconv_list = charconv_list semicolon charconv_entry
6777 | charconv_entry ;
6778 charconv_entry = '(' char_symbol comma char_symbol ')' ;
6779 ctype_symbolic_ellipses = '...' | '....' | '..(2)..' ;
6780 ctype_abs_ellipses = '...' ;
6781 ctype_tail = 'END' SP 'LC_TYPE' EOL ;
6782
6783 % The following is the LC_COLLATE category grammar
6784 lc_collate = collate_head collate_keywords collate_tail ;
6785 collate_head = 'LC_COLLATE' EOL ;
6786 collate_keywords = [ opt_statement* ] order_statements ;
6787 opt_statement = 'collating-symbol' SP collsymbol* EOL
6788 | 'collating-element' SP collelement SP 'from'
6789 SP collelem_string EOL
6790 | 'section-symbol' space+ sectionsymbol EOL
6791 | 'copy' SP FDCC_set_name EOL
6792 | 'col_weight_max' SP number EOL
6793 | 'symbol-equivalence' SP collsymbol SP
6794 collsymbol ;
6795 collelem_string = '' char_symbol+ '' ;
6796 order_statements = order_start collation_order order_end ;
6797 order_start = 'order_start' SP sectionsymbol [ semicolon
6798 order_opts ] EOL
6799 | 'order_start' SP [ order_opts ] EOL ;
6800 order_opts = order_opt [ semicolon order_opt ] ;
6801 order_opt = order_opt [ comma opt_word ] ;
6802 opt_word = 'forward' | 'backward' | 'position' ;
6803 collation_order = collation_statement* ;
6804 collation_statement = collsymbol EOL
6805 | collating_element [ SP weight_list ] EOL ;
6806 collating_element = char_symbol | collelement
6807 | ellipses | 'UNDEFINED' ;
6808 weight_list = weight_symbol [ semicolon weight_symbol ]* ;
6809 weight_symbol = <empty>
6810 | char_symbol
6811 | collsymbol
6812 | '' elem_list ''
6813 | '' symb_list '' | 'IGNORE' ;
6814 ellipses = '...' | '..' | '....' ;
6815 reorder_after = 'reorder-after' SP collsymbol EOL ;
6816 reorder_end = 'reorder-end' EOL ;
6817 reorder_section_after = 'reorder-section-after' SP sectionsymbol SP
6818 sectionsymbol EOL;
6819 reorder_section_end = 'reorder-section-end' EOL ;
6820 order_end = 'order_end' EOL ;
6821 collate_tail = 'END' SP 'LC_COLLATE' EOL ;
6822
6823 % The following is the LC_MESSAGES category grammar
6824 lc_messages = messages_head messages_keyword* messages_tail
6825 | messages_head copy_FDCC_set messages_tail ;
6826 messages_head = 'LC_MESSAGES' EOL ;
6827 messages_keyword = 'yesexpr' SP '' extended_reg_expr '' EOL
6828 | 'yesexpr' SP '' extended_reg_expr '' EOL ;
6829 messages_tail = 'END' SP 'LC_MESSAGES' EOL ;
6830
6831 % The following is the LC_MONETARY category grammar
6832 lc_monetary = monetary_head monetary_keyword* monetary_tail
6833 | monetary_head copy_FDCC_set monetary_tail ;
6834 monetary_head = 'LC_MONETARY' EOL ;
6835 monetary_keyword = mon_keyword_string SP quoted_string EOL

```

```

6836 | mon_keyword_strings SP mon_string_list EOL
6837 | mon_keyword_char SP mon_number_list EOL
6838 | mon_keyword_date SP mon_date_list EOL
6839 | 'conversion_rate' SP mon_conv_list EOL
6840 | 'mon_grouping' SP mon_group_list EOL ;
6841 mon_keyword_string = 'mon_decimal_point' | 'mon_thousands_sep'
6842 | 'positive_sign' | 'negative_sign' ;
6843 mon_keyword_strings = 'int_curr_symbol' | 'currency_symbol' ;
6844 mon_keyword_char = 'int_frac_digits' | 'frac_digits'
6845 | 'p_cs_precedes' | 'p_sep_by_space'
6846 | 'n_cs_precedes' | 'n_sep_by_space'
6847 | 'int_p_cs_precedes' | 'int_p_sep_by_space'
6848 | 'int_n_cs_precedes' | 'int_n_sep_by_space'
6849 | 'p_sign_posn' | 'n_sign_posn'
6850 | 'int_p_sign_posn' | 'int_n_sign_posn' ;
6851 mon_keyword_date = 'valid_from' | 'valid_to' ;
6852 mon_date_list = mon_date | mon_date_list semicolon mon_date ;
6853 mon_date = "' 8 * digit "' ;
6854 mon_group_list = number | mon_group_list semicolon number ;
6855 mon_string_list = quoted_string [ semicolon quoted_string]* ;
6856 mon_number_list = mon_number | mon_number_list semicolon
6857 mon_number ;
6858 mon_number = number | -1 ;
6859 mon_conv_list = mon_pair | mon_conv_list semicolon mon_pair ;
6860 mon_pair = number spaces* '/' spcaes* number ;
6861 monetary_tail = 'END' SP 'LC_MONETARY' EOL ;
6862
6863 % The following is the LC_NUMERIC category grammar
6864 lc_numeric = numeric_head numeric_keyword* numeric_tail
6865 | numeric_head copy_FDCC_set numeric_tail ;
6866 numeric_head = 'LC_NUMERIC' EOL ;
6867 numeric_keyword = num_keyword_string SP quoted_string EOL
6868 | num_keyword_grouping SP num_group_list EOL ;
6869 num_keyword_string = 'decimal_point' | 'thousands_sep' ;
6870 num_keyword_grouping = 'grouping' ;
6871 num_group_list = number
6872 | num_group_list semicolon number ;
6873 numeric_tail = 'END' SP 'LC_NUMERIC' EOL ;
6874
6875 % The following is the LC_TIME category grammar
6876 lc_time = time_head time_keyword* time_tail
6877 | time_head copy_FDCC_set time_tail ;
6878 time_head = 'LC_TIME' EOL ;
6879 time_keyword = time_keyword_name SP time_list EOL
6880 | time_keyword_fmt SP quoted_string EOL
6881 | time_keyword_opt SP time_list EOL
6882 | 'week' SP number semicolon mon_date semicolon
6883 number EOL
6884 | time_keyword_num SP number EOL
6885 | 'timezone' SP time_list EOL ;
6886 time_keyword_name = 'abday' | 'day' | 'abmon' | 'mon' | 'am_pm' ;
6887 time_keyword_fmt = 'd_t_fmt' | 'd_fmt' | 't_fmt' | 't_fmt_ampm' ;
6888 time_keyword_opt = 'era' | 'era_year' | 'era_d_fmt' | 'alt_digits'
6889 | era_d_t_fmt | era_t_fmt ;
6890 time_keyword_week = 'week' ;
6891 time_keyword_num = 'first_weekday' | 'first_workday'
6892 | 'cal_direction' ;
6893 time_list = time_list semicolon quoted_string
6894 | quoted_string ;
6895 time_tail = 'END' SP 'LC_TIME' EOL ;
6896
6897 % The following is the LC_XLITERATE category grammar
6898 lc_xliterate = translit_head [translit_include]
6899 [default_missing] translit_statement*
6900 translit_tail | translit_head copy_FDCC_set
6901 translit_tail ;
6902 translit_head = 'LC_XLITERATE' EOL ;
6903 translit_include = 'include' SP FDCC_set_name semicolon
6904 quoted_nonempty_string EOL ;
6905 default_missing = 'default_missing' SP quoted_string EOL ;

```

```

6906 translit_ignore = 'translit_ignore' SP charclass_list EOL ;
6907 translit_statement = char_or_string SP char_or_string [ semicolon
6908 char_or_string ]* EOL ;
6909 translit_tail = 'END' SP 'LC_XLITERATE' EOL ;
6910
6911 % The following is the LC_NAME category grammar
6912 lc_name = name_head name_keyword* name_tail
6913 | name_head copy_FDCC_set name_tail ;
6914 name_head = 'LC_NAME' EOL ;
6915 name_keyword = name_keyword_string SP quoted_string EOL ;
6916 name_keyword_string = 'name_fmt' | 'name_gen' | 'name_mr'
6917 | 'name_mrs' | 'name_ms' | 'name_miss'
6918 | 'name_ms' ;
6919 name_tail = 'END' SP 'LC_NAME' EOL ;
6920
6921 % The following is the LC_ADDRESS category grammar
6922 lc_address = address_head address_keyword* address_tail
6923 | address_head copy_FDCC_set address_tail ;
6924 address_head = 'LC_ADDRESS' EOL ;
6925 address_keyword = address_keyword_string SP quoted_string EOL ;
6926 address_keyword_string = 'postal_fmt' | 'country_name' |
6927 'country_post' | 'lang_name' | 'lang_ab2' |
6928 'lang_ab3_term' | 'lang_ab3_lib' ;
6929 address_tail = 'END' SP 'LC_ADDRESS' EOL ;
6930
6931 % The following is the LC_TELEPHONE category grammar
6932 lc_tel = tel_head tel_keyword* tel_tail
6933 | tel_head copy_FDCC_set tel_tail ;
6934 tel_head = 'LC_TELEPHONE' EOL ;
6935 tel_keyword = tel_keyword_string SP quoted_string EOL ;
6936 tel_keyword_string = 'tel_int_fmt' | 'tel_dom_fmt' | 'int_select'
6937 | 'int_prefix' ;
6938 tel_tail = 'END' SP 'LC_TELEPHONE' EOL ;
6939
6940 % The following grammar rules are common to all categories
6941 char = <any character except those that makes an End
6942 Of Line>
6943 graphic_char = <any char except control_chars and space> ;
6944 space = ' ' | <TAB> ;
6945 SP = space+ ;
6946 EOL = <anything that makes an End Of Line (EOL) in
6947 the operating system employed> | comment EOL ;
6948 comment_char = <defined by the 'comment_char' keyword> ;
6949 escape_char = <defined by the 'escape_char' keyword> ;
6950 charsymbol = simple_symbol | ucs_symbol ;
6951 collsymbol = simple_symbol ;
6952 collelement = simple_symbol ;
6953 sectionsymbol = simple_symbol ;
6954 octdigit = '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' ;
6955 digit = '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9' ;
6956 hex_upper = 'A' | 'B' | 'C' | 'D' | 'E' | 'F' digit ;
6957 hexdigit = hex_upper | 'a' | 'b' | 'c' | 'd' | 'e' | 'f' ;
6958 letter = 'a' | 'b' | 'c' | 'd' | 'e' | 'f' | 'g' | 'h' | 'i' | 'j' | 'k'
6959 | 'l' | 'm' | 'n' | 'o' | 'p' | 'q' | 'r' | 's' |
6960 | 't' | 'u' | 'v' | 'w' | 'x' | 'y' | 'z' | 'A' | 'B' | 'C' | 'D'
6961 | 'E' | 'F' | 'G' | 'H' | 'I' | 'J' | 'K' | 'L' | 'M' | 'N' | 'O'
6962 | 'P' | 'Q' | 'R' | 'S' | 'T' | 'U' | 'V' | 'W' | 'X' | 'Y' | 'Z' ;
6963 portable_graph_gtr = letter | digit | '!' | '"' | '#' | '$' | '%' | '&'
6964 | "'" | '(' | ')' | '*' | '+' | ',' | '-' | '.' | '/' | ':' | ';'
6965 | '<' | '=' | '?' | '@' | '[' | '\' | ']' | '^' | '_'
6966 | '{' | '|' | '}' | '~' ;
6967 portable_graph = portable_graph_gtr | '>' ;
6968 portable_char = portable_graph | ' ' | <NUL> | <ALERT>
6969 | <BACKSPACE> | <TAB> | <CARRIAGE_RETURN>
6970 | <NEWLINE> | <VERTICAL_TAB> | <FORM_FEED> ;
6971 octal_char = escape_char octdigit octdigit octdigit* ;
6972 hex_char = escape_char 'x' hexdigit hexdigit hexdigit* ;
6973 decimal_char = escape_char 'd' digit digit digit* ;
6974 number = digit+ ;
6975 id_part = letter | digit | '-' | '_' ;

```

```

6976 four_digit_hex_string = hex_upper hex_upper hex_upper hex_upper ;
6977 identifier = letter id_part* ;
6978 simple_symbol = space* '<' portable_graph_gtr+ '>' ;
6979 ucs_symbol = space* '<U' four_digit_hex_string
6980 [ four_digit_hex_string ] '>' ;
6981 quoted_string = ''' char_symbol* ''' ;
6982 quoted_nonempty_string = ''' char_symbol+ ''' ;
6983 char_symbol = char | charsymbol
6984 | octal_char | hex_char | decimal_char ;
6985 elem_list = elem+ ;
6986 elem = char_symbol | collsymbol | collelement ;
6987 symb_list = collsymbol+ ;
6988 FDCC_set_name = FDCC-name | ''' FDCC-name ''' ;
6989 copy_FDCC_set = 'copy' FDCC_set_name EOL ;
6990 FDCC-name = portable_graph+ ;
6991 semicolon = space* ';' space* ;
6992 comma = space* ',' space* ;
6993 comment = comment_char char* ;
6994

```


6995
6996
6997
6998
6999
7000
7001
7002
7003
7004
7005
7006
7007
7008
7009
7010
7011
7012
7013
7014
7015
7016
7017
7018
7019
7020
7021
7022
7023
7024
7025
7026
7027
7028
7029
7030
7031
7032
7033
7034
7035
7036
7037
7038
7039
7040
7041
7042

Annex D
(informative)

Issues list

This Technical Reports presents a trial for defining a general mechanism to specify cultural conventions. Though its contents are developed in order to form a standard, it is decided to be a technical report in order to give information to public earlier.

The issues includes but are not limited to:

- 1) Whether the features which have their origin in ISO/IEC 9945-2 --POSIX Part 2 -- works well after its separation from ISO/IEC 9945-2 or not.
- 2) Whether it makes sense or not to have a default value, which may be considered as a recommendation, for each cultural convention item.
- 3) Whether each specification form fits for world-wide cultural variations or not.

The preparer of this report, ISO/IEC JTC1/SC22, expects the rapid progress of internationalization in the field of information technology will solve the above mentioned issues and this technical report will be used as a base for a new standard in near future.

D.1 Comments from the Japanese member body

Japan considered this document should not be published as an international standard for the following reasons:

- 1) It is not clear whether the features which have their origin in ISO/IEC 9945-2 -- POSIX Part 2 -- works well or not, after its separation from ISO/IEC 9945-2. Japan considers some mechanisms, e.g. "copy", will not work outside the POSIX environments.
- 2) It is not clear whether it makes sense or not to have a default value, which may be considered as a recommendation, for each cultural convention item. Japan is afraid that those default values are considered as Global Uniformity values -- see ISO/IEC TR 11017:1998 for details.
- 3) It is not clear whether each specification form fits for world-wide cultural variations or not.

		Annex E	
		(informative)	
		Index	
7043			
7044			
7045			
7046			
7047			
7048	abbreviation	4.2	col_weight_max 4.4, 4.4.3
7049	abday	4.7	collating-element 4.4
7050	abmon	4.7	collating statements 4.4.1
7051	absolute ellipses	4.3	collating-symbol 4.4.6
7052	address	4.2	collating element 3.1.13
7053	addresses	4.11	collating sequence 3.1.15
7054	addset	5.1	collating-element 4.4.5
7055	alpha	4.3.1	collating-symbol 4.4
7056	alt_digits	4.7	collation 3.1.12
7057	am_pm	4.7	combining 4.3.1
7058	application	4.2	combining_level3 4.3.1
7059	audience	4.2	comment_char 4.1.4.1, 5.1
7060	blank	4.3.1	conformance 7
7061	block_separator	4.3.1	contact 4.2
7062	byte	3.1.1	continuation line 4.1.2
7063	cal_direction	4.7	control characters 4.3.1
7064	category	4.2	conversion_rate 4.5
7065	category names	4.1	copy 3, 4.2, 4.3.1, 4.4.2, 4.5, 4.6, 4.7, 4.8,
7066	category trailer	4.1	4.9,
7067	category header	4.1	4.10, 4.11, 4.12
7068	category body	4.1	country_ab2 4.11
7069	character	3.1.2	country_ab3 4.11
7070	character, graphic	4.3.1	country_car 4.11
7071	character, special	4.3.1	country_isbn 4.11
7072	character representation	4.1.1	country_name 4.11
7073	character, native digit	4.3.1	country_num 4.11
7074	character, hexadecimal digit	4.3.1	country_post 4.11
7075	character, multibyte	4.1.1	cultural convention 3.1.5
7076	character, decimal constant	4.1.1	currency_symbol 4.5
7077	character, hexadecimal constant	4.1.1	d_fmt 4.7
7078	character, space	4.3.1	d_t_fmt 4.7
7079	character, octal constant	4.1.1	date field descriptors 4.7.1
7080	character, control	4.3.1	date 4.2
7081	character, blank	4.3.1	day 4.7
7082	character, digit	4.3.1	decimal_point 4.6
7083	character, punctuation	4.3.1	default_missing 4.3.2
7084	character, printable	3.1.10	definitions 3.1
7085	character class	3.1.9	digit 4.3.1
7086	character, coded	3.1.3	ellipses 4.3, 4.4.1, 5.1
7087	Character set rationale	B.2	ellipses, absolute 4.3, 4.4.1
7088	charmap text	5.1	ellipses, symbolic 4.3, 4.4.1, 5.1
7089	charmap 5, 4.1.4.4, 3.1.7		email 4.2
7090	charmap rationale	B.2	equivalence class 3.1.16
7091	class	4.3.1	era 4.7
7092	cntrl	4.3.1	era_d_fmt 4.7
7093	code_set_name	5.1	era_year 4.7
7094	coded character	3.1.3	escape_char 4.1.4.2, 5.1,

7095	6esqseq	5.1	LC_IDENTIFICATION	4.2
7096	euro	B.1.3	LC_IDENTIFICATION rationale	B.1.1
7097	extended regular expression	4.8	LC_MESSAGES	4.8
7098	fax	4.2	LC_MESSAGES rationale	B.1.7
7099	FDCC-set, definition	4.1	LC_MONETARY	4.5
7100	FDCC-set	4f	LC_MONETARY rationale	B.1.4
7101	FDCC-set	3.1.6	LC_NAME	4.10
7102	FDCC-set rationale	B.1	LC_NAME rationale	B.1.9
7103	first_weekday	4.7	LC_NUMERIC	4.6
7104	first_workday	4.7	LC_NUMERIC rationale	B.1.5
7105	frac_digits	4.5	LC_TELEPHONE	4.12
7106	graph	4.3.1	LC_TELEPHONE rationale	B.1.11
7107	graphic chracters	4.3.1	LC_TIME	4.7
7108	grouping	4.6	LC_TIME rationale	B.1.6
7109	height	4.9	LC_XLITERATE	4.9
7110	include	4.3.2	LC_XLITERATE rationale	B.1.8
7111	include	5.1	LC_X_	4
7112	include	4.3.2.2	line continuation	4.1.4
7113	int_curr_symbol	4.5	lower	4.3.1
7114	int_frac_digits	4.5	map	4.3.1
7115	int_n_cs_precedes	4.5	mb_cur_max	5.1
7116	int_n_sep_by_space	4.5	mb_cur_min	5.1
7117	int_n_sign_posn	4.5	messages	4.8
7118	int_p_cs_precedes	4.5	modified date field descriptors	
7119	int_p_sep_by_space	4.5	4.7.2mon	4.7
7120	int_p_sign_posn	4.5	mon_decimal_point	4.5
7121	int_prefix	4.12	mon_grouping	4.5
7122	int_select	4.12	mon_thousands_sep	4.5
7123	keywords	4.1	monetary	4.5
7124	lang_ab	4.11	multicharacter collating element	3.1.14
7125	lang_lib	4.11	n_cs_precedes	4.5
7126	lang_name	4.11	n_sep_by_space	4.5
7127	lang_term	4.11	n_sign_posn	4.5
7128	language	4.2	name formatting	4.10
7129	LC_ADDRESS	4.11	name_fmt	4.10
7130	LC_ADDRESS rationale	B.1.10	name_gen	4.10
7131	LC_COLLATE	4.4	name_miss	4.10
7132	LC_COLLATE rationale	B.1.3	name_mr	4.10
7133	LC_CTYPE	4.3	name_mrs	4.10
7134	LC_CTYPE rationale	B.1.2	name_ms	4.10
			negative_sign	4.5
			noexpr	4.8
			notations	3.2
			numeric	4.6
			operands	4.1
			order_end	4.4.9, 4.4
			order_start	4.4, 4.4.8
			outdigit	4.3.1
			p_cs_precedes	4.5
			p_sep_by_space	4.5
			p_sign_posn	4.5
			paper format	4.9

7135	portable character set	3.2.4	transliteration statements	4.9.1
7136	positive_sign	4.5	upper	4.3.1
7137	POSIX	1	valid_from	4.5
7138	POSIX differences	A	valid_to	4.5
7139	POSIX conformance	4.2	visible glyph portable characters	3.2.4
7140	postal addresses	4.11	week	4.7
7141	postal_fmt	4.11	white space	3.1.11
7142	pre-category statements	4.1.4	width	4.9
7143	print	4.3.1	xdigit	4.3.1
7144	printable character	3.1.10	yesexpr	4.8
7145	punct	4.3.1		
7146	punctuation characters	4.3.1		
7147	redefine	4.3.2		
7148	references	2		
7149	reorder-section-end	4.4.13		
7150	reorder-section-after	4.4.12		
7151	reorder-section-after	4.4		
7152	reorder-after	4.4		
7153	reorder-end	4.4		
7154	reorder-section-end	4.4		
7155	reorder-after	4.4.10		
7156	reorder-end	4.4.11		
7157	reorder-after rationale	B.1.2.1		
7158	repertoire rationale	B.3		
7159	repertoire	6		
7160	repertoiremap	6, 3.1.8, 5.1, 4.1.4.3		
7161	revision	4.2		
7162	scope	1		
7163	section	4.4, 4.4.4		
7164	source	4.2		
7165	space	4.3.1		
7166	special characters	4.3.1		
7167	symbol-equivalence	4.4, 4.4.7		
7168	symbolic ellipses	4.3, 5.1		
7169	symbolic name	4.1.1		
7170	syntax format	3.2.1		
7171	t_fmt	4.7		
7172	t_fmt_ampm	4.7		
7173	tel	4.2		
7174	tel_dom_fmt	4.12		
7175	tel_int_fmt	4.12		
7176	telephone numbers	4.12		
7177	territory	4.2		
7178	text file	3.1.4		
7179	thousands_sep	4.6		
7180	timezone	4.7		
7181	title	4.2		
7182	tolower	4.3.1		
7183	tosymmetric	4.3.1		
7184	toupper	4.3.1		
7185	translit_ignore	4.9		
7186	transliteration	4.9		

BIBLIOGRAPHY

- 7187
7188
7189 The following specifications are considered relevant to this Technical Report, in addition
7190 to the normative references.
7191
7192 CEPT, CEPT-MAILCODE, *Country code for mail*.
7193
7194 ISO 646, *Information technology - ISO 7-bit coded character set for information inter-*
7195 *change*.
7196
7197 ISO/IEC 9899, *Information technology - Programming language C*.
7198
7199 ISO/IEC 14977, *Information technology - Syntactic metalanguage - Extended BNF*.
7200
7201 The Unicode Consortium: *The Unicode Standard, Version 2.0*, Addison Wesley
7202 Developers Press, July 1996. ISBN 0-201-48345-9.
7203
7204 IBM: *National Language Design Guide Volume 2 - National Language Support Reference*
7205 *Manual*, IBM SE09-8002-03, August 1994.
7206
7207 STRÍ: *Nordic Cultural Requirements on Information Technology (Summary report)*, STRÍ
7208 TS3, Libris, Reykjavík, Iceland 1992. ISBN 9979-9004-3-1.
7209
7210
7211
7212
7213
7214
7215
7216
7217
7218
7219
7220
7221
7222
7223
7224
7225
7226
7227
7228