Annex A (normative): National Language Tables

A.1 Introduction

This annex contains character tables for scripts that use other scripts than the Latin script, though all also support A-Z, a-z. The tables in subsections A.2 and A.3 are all deprecated (except that the default alphabet still has some uses). Se subsection A.4 for recommended tables.

A.2 National Language Single Shift 2 (SS2) Tables (deprecated)

The tables here are all deprecated, except that the default alphabet must be used where explicitly called for in the SMS and CBS standards (but is still deprecated for SMS and CBS *messages*), and the default alphabet should still be used for USSD. For SMS and CBS *messages* use the alphabets given in subclause A.4 or UCS2/UTF16BE (subclause 6.2.3).

A.2.0 ~~GSM~~ Default Alphabet Single Shift 2 (SS2) (deprecated for SMS and CBS messages)

00000000 (0x00)	Default alphabet (Latin (and uppercase Greek)) (deprecated, use alphabet with id code 0x10 (European Latin) or 0x11 (Greek) instead for SMS and CBS messages). Tables are given in subclause 6.2.1. These are the tables (except the Greek variant) that should be used for USSD, though deprecated for SMS and CBS.	Use tables in subclause A.4.10 European languages (Latin script) [plus Turkish, Vietnamese, Tagalog] instead for the Latin script. Use tables in subclause A.4.11 (Greek) instead for the Greek script.

A.2.1 Turkish National Language Single Shift 2 (SS2) Table (deprecated)

00000001 (0x01)	Turkish (deprecated, use 0x10 instead) Tables are not reproduced in this version.	Use tables in subclause A.4.10 European languages (Latin script) instead.

A.2.2 Spanish National Language Single Shift 2 (SS2) Table (deprecated)

00000010 (0x02)	Spanish (deprecated, use 0x10 instead) Tables are not reproduced in this version.	Use tables in subclause A.4.10 European languages (Latin script) instead.

A.2.3 Portuguese National Language Single Shift 2 (SS2) Table (deprecated)

00000011 (0x03)	Portuguese (deprecated, use 0x10 instead) Tables are not reproduced in this version.	Use tables in subclause A.4.10 European languages (Latin script) instead.

A.2.4 Bengali National Language Single Shift 2 (SS2) Table (deprecated)

00000100 (0x04)	Bengali (deprecated, use 0x14 instead) Tables are not reproduced in this version.	Use tables in subclause A.4.14 Bengali/Bangla instead

A.2.5 Gujarati National Language Single Shift 2 (SS2) Table (deprecated)

00000101 (0x05)	Gujarati (deprecated, use 0x15 instead) Tables are not reproduced in this version.	Use tables in subclause A.4.15 Gujarati instead

A.2.6 Hindi National Language Single Shift 2 (SS2)Table (deprecated)

00000110 (0x06)	Hindi (deprecated, use 0x16 instead) Tables are not reproduced in this version.	Use tables in subclause A.4.16 Hindi (Devanagari) instead

A.2.7 Kannada National Language Single Shift 2 (SS2) Table (deprecated)

00000111 (0x07)	Kannada (deprecated, use 0x17 instead) Tables are not reproduced in this version.	Use tables in subclause A.4.17 Kannada instead

A.2.8 Malayalam National Language Single Shift 2 (SS2) Table (deprecated)

00001000 (0x08)	Malayalam (deprecated, use 0x18 instead) Tables are not reproduced in this version.	Use tables in subclause A.4.18 Malayalam instead

A.2.9 Oriya National Language Single Shift 2 (SS2) Table (deprecated)

00001001 (0x09)	Oriya (deprecated, use 0x19 instead) Tables are not reproduced in this version.	Use tables in subclause A.4.19 Oriya/Odia instead

A.2.10 Punjabi National Language Single Shift 2 (SS2) Table (deprecated)

00001010 (0x0A)	Punjabi (deprecated, use 0x1A instead) Tables are not reproduced in this version.	Use tables in subclause A.4.1A Punjabi (Gurmukhi) instead

A.2.11 Tamil National Language Single Shift 2 (SS2) Table (deprecated)

00001011 (0x0B)	Tamil (deprecated, use 0x1B instead) Tables are not reproduced in this version.	Use tables in subclause A.4.1B Tamil instead

A.2.12 Telugu National Language Single Shift 2 (SS2) Table (deprecated)

00001100 (0x0C)	Telugu (deprecated, use 0x1C instead) Tables are not reproduced in this version.	Use tables in subclause A.4.1C Telugu instead

A.2.13 Urdu National Language Single Shift 2 (SS2) Table (deprecated)

00001101 (0x0D)	Urdu (deprecated, use 0x13 instead) Tables are not reproduced in this version.	Use tables in subclause A.4.13 Urdu (Eastern Arabic) instead

A.3 National Language Locking Shift Tables for alphabets 0x01 to 0x0D (deprecated)

The tables here are all deprecated, except that the default alphabet must be used where explicitly called for in the SMS and CBS standards (but is still deprecated for SMS and CBS *messages*), and the default alphabet should still be used for USSD. For SMS and CBS *messages* use the alphabets given in subclause A.4 or UCS2/UTF16BE (subclause 6.2.3).

A.3.0 ~~GSM~~ Default Alphabet Locking Shift Table (deprecated for SMS and CBS messages)

00000000 (0x00)	Default alphabet (Latin (and uppercase Greek)) (deprecated, use alphabet with id code 0x10 (European Latin) or 0x11 (Greek) instead for SMS and CBS messages). Tables are given in subclause 6.2.1. These are the tables (except the Greek variant) that should be used for USSD, though deprecated for SMS and CBS.	Use tables in subclause A.4.10 European languages (Latin script) [plus Turkish, Vietnamese, Tagalog] instead for the Latin script. Use tables in subclause A.4.11 (Greek) instead for the Greek script.

A.3.1 Turkish National Language Locking Shift Table (deprecated)

00000001 (0x01)	Turkish (deprecated, use 0x10 instead) Tables are not reproduced in this version.	Use tables in subclause A.4.10 European languages (Latin script) instead.

A.3.2 Void

A.3.3 Portuguese National Language Locking Shift Table (deprecated)

00000011 (0x03)	Portuguese (deprecated, use 0x10 instead) Tables are not reproduced in this version.	Use tables in subclause A.4.10 European languages (Latin script) instead.

A.3.4 Bengali National Language Locking Shift Table (deprecated)

00000100 (0x04)	Bengali (deprecated, use 0x14 instead) Tables are not reproduced in this version.	Use tables in subclause A.4.14 Bengali/Bangla instead

A.3.5 Gujarati National Language Locking Shift Table (deprecated)

00000101 (0x05)	Gujarati (deprecated, use 0x15 instead) Tables are not reproduced in this version.	Use tables in subclause A.4.15 Gujarati instead

A.3.6 Hindi National Language Locking Shift Table (deprecated)

00000110 (0x06)	Hindi (deprecated, use 0x16 instead) Tables are not reproduced in this version.	Use tables in subclause A.4.16 Hindi (Devanagari) instead

A.3.7 Kannada National Language Locking Shift Table (deprecated)

00000111 (0x07)	Kannada (deprecated, use 0x17 instead) Tables are not reproduced in this version.	Use tables in subclause A.4.17 Kannada instead

A.3.8 Malayalam National Language Locking Shift Table (deprecated)

00001000 (0x08)	Malayalam (deprecated, use 0x18 instead) Tables are not reproduced in this version.	Use tables in subclause A.4.18 Malayalam instead

A.3.9 Oriya National Language Locking Shift Table (deprecated)

00001001 (0x09)	Oriya (deprecated, use 0x19 instead) Tables are not reproduced in this version.	Use tables in subclause A.4.19 Oriya/Odia instead

A.3.10 Punjabi National Language Locking Shift Table (deprecated)

00001010 (0x0A)	Punjabi (deprecated, use 0x1A instead) Tables are not reproduced in this version.	Use tables in subclause A.4.1A Punjabi (Gurmukhi) instead

A.3.11 Tamil National Language Locking Shift Table (deprecated)

00001011 (0x0B)	Tamil (deprecated, use 0x1B instead) Tables are not reproduced in this version.	Use tables in subclause A.4.1B Tamil instead

A.3.12 Telugu National Language Locking Shift Table (deprecated)

00001100 (0x0C)	Telugu (deprecated, use 0x1C instead) Tables are not reproduced in this version.	Use tables in subclause A.4.1C Telugu instead

A.3.13 Urdu National Language Locking Shift Table (deprecated)

00001101 (0x0D)	Urdu (deprecated, use 0x13 instead) Tables are not reproduced in this version.	Use tables in subclause A.4.13 Urdu (Eastern Arabic) instead

A.4 7-bit alphabet tables (base table and SS2 and SS3 extension tables) for alphabets 0x10 to 0x24

The third part of the subsection numbers in A.4 are in hexadecimal, 0x10 and upward, same as the reference number for the 7-bit alphabet given in tables in the subsection.

Setting, in the SMS, CSB, or USSD protocols, as locking shift *or* single shift alphabet table one with a reference number 0x10 or higher sets all three (locking shift, SS2, SS3) tables. In most cases the SS3 table is empty (and thus not given at all below), except for 0x10 where there are a few entries (but no table per se, since there are so few entries, just a sentence).

A.4.10 SMS/CBS 7-bit European languages (Latin script) Alphabet (0x10)

[for now: see separate file]

A.4.11 SMS/CBS 7-bit Greek Alphabet (0x11)

[for now: see separate file]

A.4.12 SMS/CBS 7-bit Hebrew Alphabet (0x12)

[for now: see separate file]

A.4.13 SMS/CBS 7-bit Urdu (Eastern Arabic script) Alphabet (0x13)

[for now: see separate file]

A.4.14 SMS/CBS 7-bit Bengali/Bangla Alphabet (0x14)

[for now: see separate file]

A.4.15 SMS/CBS 7-bit Gujarati Alphabet (0x15)

[for now: see separate file]

A.4.16 SMS/CBS 7-bit Hindi (Devanagari script) Alphabet (0x16)

[for now: see separate file]

A.4.17 SMS/CBS 7-bit Kannada Alphabet (0x17)

[for now: see separate file]

A.4.18 SMS/CBS 7-bit Malayalam Alphabet (0x18)

[for now: see separate file]

A.4.19 SMS/CBS 7-bit Oriya/Odia Alphabet (0x19)

[for now: see separate file]

A.4.1A SMS/CBS 7-bit Punjabi (Gurmukhi script) Alphabet (0x1A)

[for now: see separate file]

A.4.1B SMS/CBS 7-bit Tamil Alphabet (0x1B)

[for now: see separate file]

A.4.1C SMS/CBS 7-bit Telugu Alphabet (0x1C)

[for now: see separate file]

A.4.1D SMS/CBS 7-bit Thai Alphabet (0x1D)

[for now: see separate file]

A.4.1E SMS/CBS 7-bit Lao Alphabet (0x1E)

[for now: see separate file]

A.4.1F SMS/CBS 7-bit Khmer Alphabet (0x1F)

[for now: see separate file]

A.4.20 SMS/CBS 7-bit Manipuri Alphabet (0x20)

[for now: see separate file]

A.4.21 SMS/CBS 7-bit Sinhala Alphabet (0x21)

[for now: see separate file]

A.4.22 SMS/CBS 7-bit Armenian Alphabet (0x22)

[for now: see separate file]

A.4.23 SMS/CBS 7-bit Georgian Alphabet (0x23)

[for now: see separate file]

A.4.24 SMS/CBS 7-bit Ukrainian (Cyrillic script) Alphabet (0x24)

[for now: see separate file]

Annex B (informative): Guidelines for creating 7-bit alphabet language tables (base and extension)

B.1 Introduction

Please see the numerous examples above for format. The tables need to be carefully constructed.

When reviewing it is important to consider which letters are in modern use, which punctuation and currency symbols are in modern use, as well as the Unicode normalisation (in particular NFC) and which characters may be produced by composition in case there is no space (in the locking shift and SS2 tables; SS3 should be avoided). If not all relevant letters fit in the tables, some may be accessible using combining marks.

In some cases, one may need to rely on the decimal character reference mechanism (section D) for some characters. Certain internationally useful symbols should be included, at the same position as in the existing 7-bit alphabets. Certain characters, LF, CR, SS2, FF, CSI, SS3, SP, as well as *, #, /, \, +, 0-9, a-z (lowercase) must be in the same positions as all 7-bit alphabet tables.

B.2 Templates for defining 7-bit alphabet language tables

See numerous examples in the subsections A.4.x.1 above for how base tables are defined. LF, CR, SS2, SP, as well as *, #, /, \, +, 0-9, _, a-z (lowercase) must be in the same positions in all 7-bit alphabet tables.

See numerous examples in the subsections A.4.x.2 above for how single shift tables are defined. FF, CSI, and SS3 must be in the same positions in all 7-bit alphabet tables (except that CSI does not exist for 0x00-0x0F, nor does really SS3).

B.3 Void

Annex C (Informative): Example mechanism for setting base and single shift 7-bit language tables

C.1 Introduction

This annex gives an overview on how the alphabet locking shift and single shift mechanism of the 7-bit alphabets work. This annex shows how a message with an alphabet indication of the European languages (Latin script) (0x10) Alphabet Identifier is decoded, but the same principles apply to other alphabets. Note that the locking shift and single shift alphabets must be set to the same value (that there are two settings is a remnant from earlier versions).

C.2 Example of setting 7-bit alphabet language tables

This example outlines the behaviour of both supporting and non-supporting receiving entities where the Thai National Language Single Shift Table (0x1D) **or** the Thai National Language Locking Shift Table (or both) is indicated in the received message. Note that setting either the Single Shift table or the Locking shift table, since the reference code (0x1D) is in the 0x10 or larger range, will set all three (Base, SS2, SS3) tables. If they are explicitly set inconsistently, it is same as not setting them.

A non-supporting receiving entity will ignore the National Language IEs, and decode the message contents using the 7-bit default alphabet tables (0x00) defined in subclause 6.2.1 (except 6.2.1.3), including possible SS2 characters to the 7-bit default alphabet extension table specified in subclause 6.2.1.2. This will for the most part result in completely garbled text, but such things as USSD commands (using *, +, #, 0-9) and 2-letter language tags (using lowercase a-z) will remain unchanged; likewise for much of English language text *in lowercase* and some simple punctuation. A non-supporting receiving entity may use the 0x10 alphabet as fallback instead of 0x00, if the 0x10 alphabet (European languages) is supported.

A receiving entity that supports the Thai Language Tables will detect a National Language IE in a TP User Data Header. This IE tells the receiving entity that the Thai Alphabet (all three tables, locking shift, SS2, SS3) is used. A supporting receiving entity will notice the language code, in this example coded as 0x1D, and therefore use the Thai National Language Tables defined in subclause A.4.1D instead of the 7 bit default alphabet extension table defined in subclause 6.2.1.

For 7-bit alphabets with reference number 0x10 or higher, submessages of a message need not use the same 7-bit alphabet, they may be mixed and may also mix in the UCS2 (UTF-16BE) alphabet for some of the submessages, but no 7-bit alphabet with reference number 0x00 (even as default) to 0x0F.

C.3 Void

Annex D Control sequences, using CSI

The (new) alphabets in subsection A.4 have a control code CSI, CONTROL SEQUENCE INTRODUCER. It comes from the standard ECMA-48, also ISO/IEC 6429 which is technically the same standard in ISO/IEC dress. Some CSI based control sequences are commonly implemented in terminal emulators, and some of the control sequences are for use in terminal emulators only. However, we are here (Annex D and E) referring to control sequences that are useful also outside of terminal emulators.

We will use the following general syntax for control sequences:

control-sequence_s ::= **CSI**[0-9:;=?]*[@A-Z_a-z]

This is slightly more limited compared to the ECMA-48 specification, but quite sufficient here. Indeed, we will only use the final characters "_" and "m". Note that we refer to characters, not byte codes.

Any error compared to this syntax (i.e. start with CSI, but then not fulfill the given syntax) results in REPLACEMENT CHARACTER for the CSI and as is for the rest of the sequence. Other syntax errors, specific to "_" or "m" also result in REPLACEMENT CHARACTER for the CSI and as is for the rest. So does, for SMS and CBS, using other terminating characters than "_" or "m". (Other systems may support more control sequences than those defined for SMS/CBS.)

D.1 Decimal character references

Unicode character references, in ECMA-48 style, for (Unicode) scalar values greater than U+009F. This mechanism is similar to HTML's decimal numeric character references. A decimal character reference as a control sequence, compatible with ECMA-48 (but not included in ECMA-48, since the last update of ECMA-48 was in 1976, long before Unicode (and ISO/IEC 10646) first versions):

CSI*n*_, where *n* is a Unicode scalar value in decimal form (using characters 0-9), without leading zeroes and only referring to a valid Unicode scalar value (i.e. less than 1114112 and not referring to a "surrogate") and is larger than 159 (decimal), but subject to the mappings in subsection 6.2.3. This form of character reference is similar to HTML's (decimal) character references: &#n;, using **CSI** instead of &# and _ instead of ;.

Note that hexadecimal cannot be used inside a control sequence due to the ECMA-48 syntax, where A-F (and a-f) are control sequence terminating characters.

This mechanism can be used to refer to a character that is not included in the used 7-bit encoding. It can be used for U+0127 (LATIN SMALL LETTER H WITH STROKE), ħ: **CSI**295_, used in Maltese, without needing to go over to UTF-16BE representation for the (sub)message. (Note that ħ is included in the SS3 table for alphabet 0x10.)

However, there should be only a small number of such character references in a message, since each take up several 7-bit code units. At some point it is more advantageous to use UTF16BE (or possibly another 7-bit alphabet, depending).

The interpretation of control sequences, like **CSI***n*_, must be done after the entire message is composed from all of the submessages it is composed from. If an implementation allows for partial display of a message during the collection of submessages, then only the complete control sequences are interpreted. Partial ones result in REPLACEMENT CHARACTER being displayed.

D.2 Text styling

Styling text using the mechanisms specified in ECMA-48 (ISO/IEC 6429) is popularly implemented in terminal emulators. But styling text is of course not limited to terminal emulators, nor is that particular mechanism as such limited to terminal emulators. However, styling by the mechanisms afforded by HTML/CSS cannot be used for SMS/CBS. Way too verbose, and not compatible with "plain text". However, the mechanism afforded by ECMA-48 is compatible with "plain text" (which is why it at all can be used in terminal emulators), and thus compatible with SMS/CBS messages.

The text style (and text colour) are given as "style state changes", and can be combined. E.g. bold and underlined can be combined, italics and red can be combined. Changes in one text style (or text colour) aspect does not change any other text style/colour aspect.

Note that the space after "CSI" is for readability in the subsections below. It must not occur in the actual control sequence.

D.2.1 Bold

- **CSI 1m** Bold font variant. "Medium" boldness.
- **CSI 22m** Normal weight (not bold). (Default.)

Note: there is no change in nominal colour in any way for these font weight changes, other than the overall optical change due to the change in weight.

D.2.2 Italic/oblique

- **CSI 3m** Italicized or oblique. Serifed fonts usually have a special variant for italics. Sans-serif fonts are often just slanted (oblique) relative to the upright variant. The slant should be around −7° to −10°.
- **CSI 23m** Upright (a.k.a. roman). (Default.)

D.2.3 Raised to superscript position

Proper superscript styling is not part of ECMA-48 5th edition (from 1976), but it is a common typographical styling with meaning. Common enough to warrant inclusion here (for SMS messages). These superscripts are not really intended for math expressions, but may still be used for exponentiation of units, as can the characters ^{2} and ^{3}, included in all the new alphabets above.

- **CSI 56:1m** Raised to first superscript level and slightly smaller, about 80% current set font size. Used for ordinals (e.g. 5th, 1er), and other uses (e.g. $\mathrm{M^{lle}}$, $\mathrm{XX^{e}}$).
- **CSI 56m** Not raised and use to the set font size. (Default.) This reset is auto-applied just before LF, SP and punctuation/symbol (except if a **CSI 56:1m** is just before the punctuation/symbol).

The two "ordinal indicator" letters (included in the new Default Alphabet), mainly for Iberian languages, should look just like the proper display of CSI 56:1m**a**CSI 56m and CSI 56:1m**o**CSI 56m. Likewise for ^{2} and ^{3}.

D.2.4 Underline

- **CSI 4m** Underlined. Single underline of "medium" weight.
- **CSI 24m** Not underlined. (Default.)

D.3 Text colouring

Text colouring is usually seen as a type of text styling. But purely for presentation reasons, we here split text colouring into its own subsection (this one).

Care should be taken (by the message author) so that the colours chosen results in a readable text, rather than a hidden or hard to read text.

D.3.1 Text “marker pen” colouring

For truly black and white (no greyscale) displays, the colour changes have no effect (not even black and white). For greyscale, all the colours here (even yellow, black, white) may be displayed as light grey without transparency.

Text background colour, or better referred to as text highlight colour or “marker pen” colour. The default ‘colour’ for this is fully transparent (so one gets the background colour that is “behind” the text block). This colour “plane” is “below” the text “foreground” colour, but “above” the text block background.

Colour is given as RGB (Red, Green, Blue) additive colour component values; 0 is none, 255 is max. Transparency for this highlight (“marker pen”) colour is implementation defined, like 20 % transparent, but could be opaque. The colours in the list are too “harsh” as is as marker pen colouring, but with transparency they are more “marker pen”-like.

CSI 40m : RGB: **0:0:0**	Pure black ⬛
CSI 41m : RGB: **205:0:0**	Dull red 🟥
CSI 42m : RGB: **0:205:0**	Dull green 🟩
CSI 43m : RGB: **255:215:0**	Dull yellow 🟨
CSI 44m : RGB: **0:0:205**	Dull blue 🟦
CSI 45m : RGB: **205:0:205**	Dull magenta 🟪
CSI 46m : RGB: **0:205:205**	Dull cyan 🟦
CSI 47m : RGB: **255:255:255**	Pure white
CSI 49m Fully transparent (default); transparency **255**.	
CSI 100m : RGB: **105:105:105**	Dark grey ■
CSI 101m : RGB: **255:0:0**	Clear red 🟥
CSI 102m : RGB: **0:255:0**	Clear green 🟩
CSI 103m : RGB: **255:255:0**	Clear yellow 🟨
CSI 104m : RGB: **0:0:255**	Clear blue 🟦
CSI 105m : RGB: **255:0:255**	Clear magenta 🟪
CSI 106m : RGB: **0:255:255**	Clear cyan 🟦
CSI 107m : RGB: **220:220:220**	Light grey ⬜
CSI 108m : RGB: **169:169:169**	Medium grey ■
CSI 109m : RGB: **255:140:0**	Orange 🟧

The space after “CSI” is for readability here. It must not occur in the actual control sequence. The 108 and 109 are additions compared to existing terminal emulators.

D.3.2 Text colour (“foreground” colour)

Colour for the text itself. For truly black and white (no greyscale) displays, the colour changes have no effect (not even black and white). For greyscale, all the colours here (even yellow, black, white) may be displayed as dark grey. The colour of underlines (if any) follows the colour of the text.

These colours should not have any transparency (though that is implementation defined). Note that this colour is “above” the “marker pen”-like colouring. These colours are not transparent. The colour RGB values is a compromise between the existing terminal emulators (with a preference for xterm) colour RGB values. The 98 and 99 are additions compared to existing terminal emulators.

CSI 30m : RGB: **0:0:0** (common default)	Pure black ⬛

CSI 31m : RGB: **205:0:0**	Dull red 🟥
CSI 32m : RGB: **0:205:0**	Dull green 🟩
CSI 33m : RGB: **255:215:0**	Dull yellow 🟨
CSI 34m : RGB: **0:0:205**	Dull blue 🟦
CSI 35m : RGB: **205:0:205**	Dull magenta 🟪
CSI 36m : RGB: **0:205:205**	Dull cyan 🟦
CSI 37m : RGB: **255:255:255**	Pure white
CSI 39m Default text colour (implementation defined)	
CSI 90m : RGB: **105:105:105**	Dark grey ⬛
CSI 91m : RGB: **255:0:0**	Clear red 🟥
CSI 92m : RGB: **0:255:0**	Clear green 🟩
CSI 93m : RGB: **255:255:0**	Clear yellow 🟨
CSI 94m : RGB: **0:0:255**	Clear blue 🟦
CSI 95m : RGB: **255:0:255**	Clear magenta 🟪
CSI 96m : RGB: **0:255:255**	Clear cyan 🟦
CSI 97m : RGB: **220:220:220**	Light grey ⬜
CSI 98m : RGB: **169:169:169**	Medium grey ⬛
CSI 99m : RGB: **255:140:0**	Orange 🟧

The space after “CSI” is for readability here. It must not occur in the actual control sequence.

Usage advice: the text marking and text colour settings should have good contrast, i.e. not try to hide the text or make it hard to read; at least when assuming reasonable default colours (light background (incl. marking), dark text).

D.3.3 Style control abbreviations

A sequence of style setting can be put together in a single CSI-sequence, with semicolon as separator. E.g. bold (CSI1m) clear red (CSI91m), can be specified as CSI1;91m.

An empty sequence, CSIm, is not a no-op, instead it resets all styles to default style (upright, normal weight, not underlined, not superscript, default colours).

These abbreviations are specified in ECMA-48 (ISO/IEC 6429) and have been commonly implemented in terminal emulators.