

CZ-0053

Clause/Subclause/Annex: Part 4/2.15.1.28

Paragraph/Figure/Table/Note: p. 1158/l. 19

Type: te

| (5) | (6) |
|--|--|
| Comment (justification for change) by the MB | Proposed change by the MB |
| Text assumes that Unicode string is represented using UCS-2 encoding where each character is stored in exactly two bytes. Nowadays Unicode contains almost 100000 characters and other encodings with full Unicode coverage like UTF-16 have to be used. In UTF-16 some characters are stored in four bytes using surrogate pairs. | Specify which encoding is used for Unicode string representation. Instead of using high and low bytes base description on octet positions. |

Proposed Disposition

Agreed; the following changes will be made to make this assumption explicit in each algorithm:

Part 4, §2.15.1.28, page 1,158, line 15:

First, the [UTF-16LE encoded](#) password shall be hashed using the following algorithm. [If there is a leading BOM character \(U+FEFF\) in the encoded password it is removed before hash calculation.](#) [The following steps assume that all words are unsigned, the word size is two bytes, and that bit-level SHL/SHR operations shift in the direction of the highest-order and lowest-order bit, respectively. \[Example: 0x61 SHR 1 is 0xC2, as 01100001 shifted one position in the direction of its highest-order bit is 11000010. end example\]:](#)

Similar Comments: [BR-0006](#), [CL-0091](#), [CO-0098](#), [DE-0088](#), [GB-0220](#), [GH-0009](#), [GR-0027](#), [IN-0075](#), [IR-0013](#), [MY-0021](#), [PT-0039](#), [US-0052](#), [UY-0017](#), [VE-0018](#)