The Global Network for B2B Integration in High Tech Industries

EDIFICE Guideline for Web and keyboard compatible encoding with ASC Data Identifiers

Issue 1.0

Endorsed on 2 January 2018

Notwithstanding the fact that the utmost care has been observed in the collecting, drawing up and formulating of data, EDIFICE can under no circumstances be held liable for errors, omissions or misinterpretations as a result of the information compiled in the guidelines.

EDIFICE
The Global Network for B2B Integration in High Tech Industries
EDIFICE secretariat
Dora Cresens
Tiensestraat 12
B-3320 Hoegaarden
Belgium
Tel: +32 16 437 415
Email: Dora.Cresens@edifice.org

Publication Summary

Title:	Guideline for Web and keyboard compatible encoding with ASC Data Identifiers
Author (s):	EDIFICE Task Group ADC (Automatic Data Capture)
Issue number:	Issue 1.0
Date of Issue:	05 December 2017
Number of Pages:	11
Readership:	All
Language:	English
Abstract:	Web and keyboard compatible encoding of data-elements in machine readable media using ASC Data Identifiers
Comment:	Comments and change requests to this document should be submitted to: EDIFICE secretariat dora.cresens@edifice.org
References:	EDIFICE ADC Glossary Issue 1 ANS MH10.8.2 Data Identifier & Application Identifier Standard AFNOR NFZ63-400 Système de codification multisectorielle des entities DIN 66403 System Identifiers (with "." as "System Identifier") DIN 16599 Automatic identification and data capture techniques — Traceability
Applications:	Applications within the Electronic and supply industries are potentially known everywhere where data are scanned from AIDC media and to be transmitted through WEB interfaces, keyboard interfaces or virtual keyboard interfaces. It is known that the subject of the EDIFICE guideline is experienced in AIDC applications already like for unique identification within Healthcare processes, and that it is part of the CEN project "Fishbox Label".

Table of Content

.... 1
Table of Content.. 3
Comparison to previous issue.. 4
1. Introduction 4
  1.1 General Context 4
  1.2 Problems to Solve.. 5
  1.3 Scope.. 6
2. Definitions... 6
3. Flag Character and Rules... 7
  3.1 Flag Character and Rules 7
  3.2 Example Encoding Flow 7
  3.3 Example Decoding Flow 7
  3.4 Summary 7
4. Examples 7
  4.1 Example of "Unique Identification Mark – UIM"................................... 7
  4.2 Example of a stock location code 8
  4.3 Example with two concatenated data elements.................................. 9
ANNEX A: (INFORMATIVE) FLAG CHARACTER UNAMBIGUITY................................10
ANNEX B: (INFORMATIVE) INNER SEPARATOR CHARCTERS 11

Table 1 - Examples of ambiguity 5
Table 2 - Summary 7

Figure 1 - Layers in AIDC Media 4
Figure 2 - Example scenario with scanner and keyboard interface and internet browser 5
Figure 3 - Example scenario where scanned data have to pass USB KB interfaces and virtual KB interfaces 6
Figure 4 - Example scenario with virtual keyboard and internet browser 6
Figure 5 - UIM on a test tube applied with flag character 8
Figure 6 Stock location code (source: Klinikum of the Friedrich Schiller University of Jena (UKJ), Germany) 8
Figure 7 - Example Code 128 with two ASC DI headed data elements concatenated by separator "^" 9

Comparison to previous issue

This is the first edition of the guideline, no previous version exists.

1. Introduction

1.1 General Context

Media for Automatic Identification and Data Capture such as linear barcodes, two-dimensional barcodes and RFID are used to transfer data. The AIDC media with the contained data follows specific standards for the different layers.

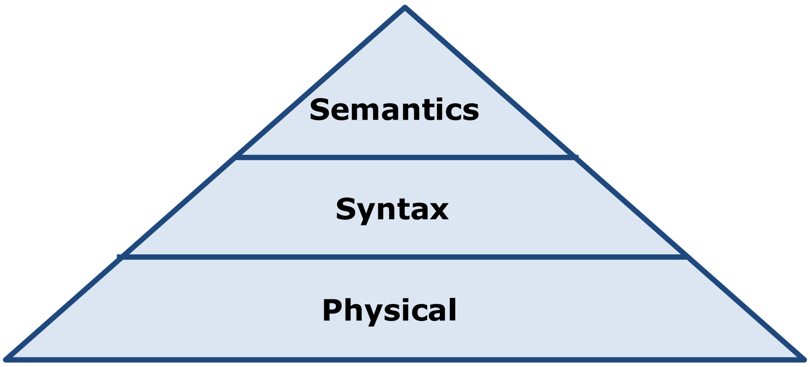


Figure 1 - Layers in AIDC Media

Examples for standards for the physical layer are linear Barcodes such as Code 128 according to ISO/IEC 15417, two-dimensional codes such as DataMatrix according to ISO/IEC 16022 or RFID in various technologies described in specific standards.

An example for standards for the syntax layer is the Data Transfer Syntax according ISO/IEC 15434.

Examples for standards for the semantics layer are the Data Identifier and Application Identifier according to ANS MH10.8.2.

Several organizations originating from different industry sectors have established consistent sets of syntactical and semantical rules for specific applications. Typically a "Flag Character" as first character introduces the encoded data, and is followed by data in syntax and sematic associated to this "Flag Character".

Examples are the "+" (plus-character) from the Health Industry Business Communications Council (HIBCC) and the "FNC1" (function code 1 character) from Global Standards One (GS1).

Historically barcode and RFID has been implemented under the assumption that for automatic identification in open supply chain systems data syntax and semantics always follows standardized rules such as ISO/IEC 15434 in combination with MH10.8.2 ASC DIs. Barcodes not following standards for syntax and semantics have been expected to be used in closed loop, internal applications only.

Nevertheless the development of barcode applications increased rapidly as seen with the wide spread use of QR Code specifically for web connectivity.

In the past Web and keyboard compatible interfacing was neglected by promoting complex and non-Web and keyboard compatible syntax like ISO/IEC 15434.

1.2 Problems to Solve

1.2.1 Ambiguity of data elements applied with ASC Data Identifiers without any Flag Character

Data elements headed by ASC Data Identifiers are not secured against overlapping with encoded data of other nature or other structurers like "IUID" headed by Text Element Identifiers (TEI). Table 1 is addressing some examples.

Table 1 - Examples of ambiguity

ASC DI headed data element	Data element of other generic structure	Data element with Text Element Identifier (TEI)	Observation
S1234567	S1A2B3C4	SER1234567	Any of the 3 data elements could be interpreted as an ASC DI headed Serial No.
P12345678	P123XYZ	PNR4321A234	Any of the 3 data elements could be interpreted as an ASC DI headed Product Code
1T98765432	1T123XYZ		Any of the two data elements could be interpreted as an ASC DI headed LOT No.

The today's solution is the use of ISO/IEC 15434 Syntax for high capacity media introducing the problems of non-Web and keyboard compatibility described in chapter 2.2.2.

1.2.2 Non-Web/Keyboard compatibility of ISO/IEC 15434 Syntax

Unique syntax using control characters outside the keyboard set, like ISO/IEC 15434 Syntax for high capacity media does (<rs>, <gs>, <eot>, etc), do not pass such physical or virtual interfaces and may even invoke unwanted functionalities that may be assigned to corresponding keys by the application.

The solution analysis of application scenarios is covering the aspects of CODE → Scanner → Interface → Systems in different scenarios.

Figure 2 below illustrates a scenario for a regular application where an off the shelf scanner shall transmit the data to the determined system through USB KB Interface to an Internet browser.

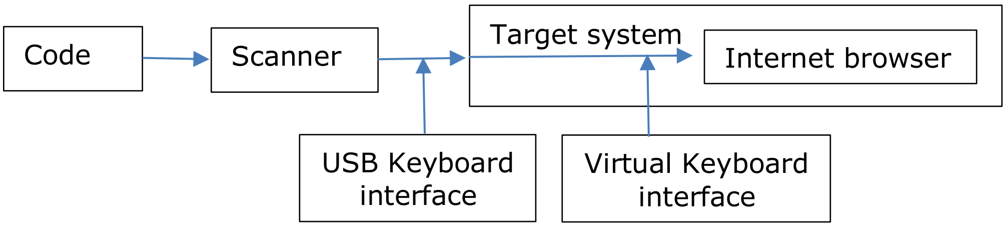


Figure 2 - Example scenario with scanner and keyboard interface and internet browser

The example scenario in figure 3 illustrates situations where scanned data have to pass USB KB interfaces and virtual KB interfaces

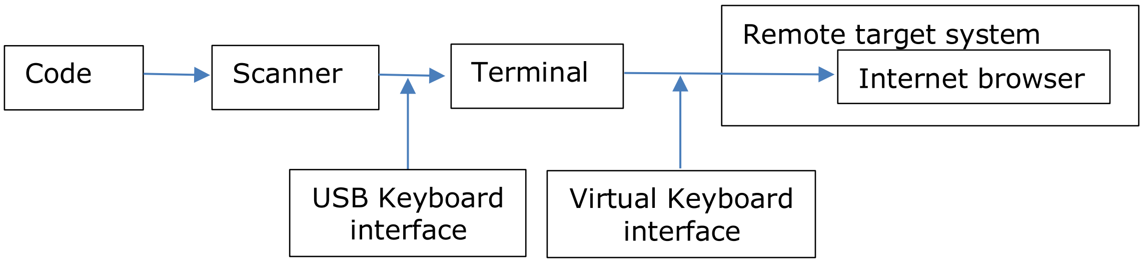


Figure 3 - Example scenario where scanned data have to pass USB KB interfaces and virtual KB interfaces

The example in figure 4 illustrates a scenario with a code scanning and processing by smartphone and App for using the code content for Internet browsing

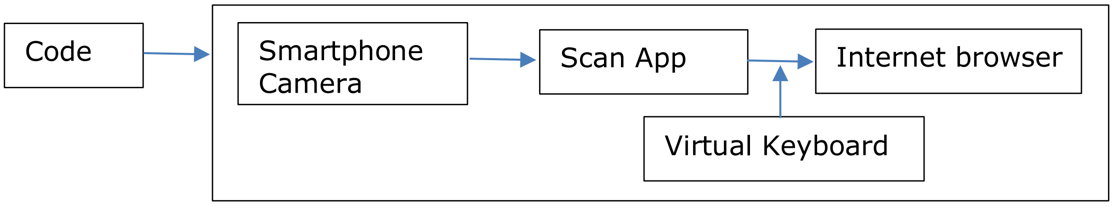


Figure 4 - Example scenario with virtual keyboard and internet browser

The purpose of this EDIFICE guideline is to describe a Web and keyboard compatible encoding of data in machine readable media covering the scenarios above.

1.3 Scope

The scope of this guideline is the definition of a unique encoding scheme of data elements applied with ASC Data Identifiers for use in applications where Web and keyboard compatible syntax is required and unambiguity compared to non ASC DI data elements.

Although the background of EDIFICE is the High-Tech Industry, the application of this guideline is not limited to a specific industry sector or subset of Data Identifiers.

2. Definitions

See ADC Glossary available at www.edifice.org.

Flag character	Header of data strings identifying the syntax being used, e.g. "+" for the Health Industry Bar Code (HIBC), "FNC1" for the Global Standards One (GS1) Application Identifier syntax, "=" for the ICCBBA ISBT 128 system, etc.

3. Flag Character and Rules

3.1 Flag Character and Rules

- The Flag-Character is the "." (dot) character: this is the very first character of the data string.
- All data elements are headed by ASC Data Identifiers.
- Data elements shall not contain a "^" (circumflex) character.
- If more than one data element is encoded in the same symbol, then the data elements are separated by a "^" (circumflex) character.

3.2 Example Encoding Flow

An example for encoding flow is:

1. *Ensure that all data elements are headed by the appropriate ASC Data Identifiers, and that the data elements do not contain a "^" (circumflex) character.*
2. *Take the first data element and put a "." (dot) character in front of this data element*
3. *In case of further data elements put a "^" (circumflex) before the next data element and then append the next data element.*
4. *Repeat step 3 until no further data elements are to be encoded.*

3.3 Example Decoding Flow

An example for decoding flow is:

1. *Read entire string of characters encoded in the symbol*
2. *Check whether the first character of the encoded data string is a "." dot character.*
3. *Remove the leading "." dot character.*
4. *Split the data string at each "^" (circumflex) character into single data elements.*
5. *Interpret the single data elements by its Data Identifiers (first 1…4 characters) using the data element definitions of ASC MH10.8.2.*

3.4 Summary

Table 2 below gives a summary for the usage of the dot character as flag character.

Table 2 - Summary

FLAG CHARACTER	EXPLANATON
.	The "DOT" character is the flag character to identify Web and keyboard compatible encoding with ASC Data Identifiers. The separator between encoded data elements is the character "^" (Circumflex)

4. Examples

4.1 Example of "Unique Identification Mark – UIM"

The example in figure 5 shows a globally unique serial number with flag character "." identifying a test tube.

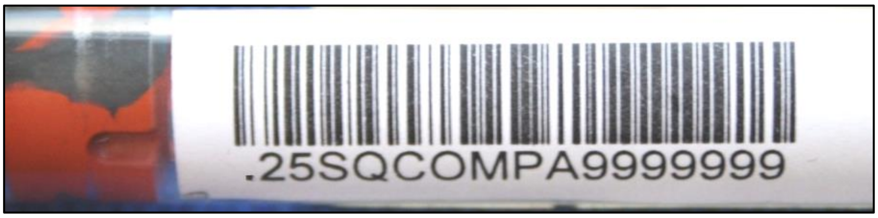


Figure 5 - UIM on a test tube applied with flag character

Data sequence of the example Fig. 5)
.25SQCCOMPA9999999
Serial Number: A9999999
CIN: COMP
IAC: QC
ASC Data Identifier: 25S
System Identifier for the ASC DI structure: "**.**"

4.2 Example of a stock location code

Figure 6 below shows a pallet stock with stock location codes.

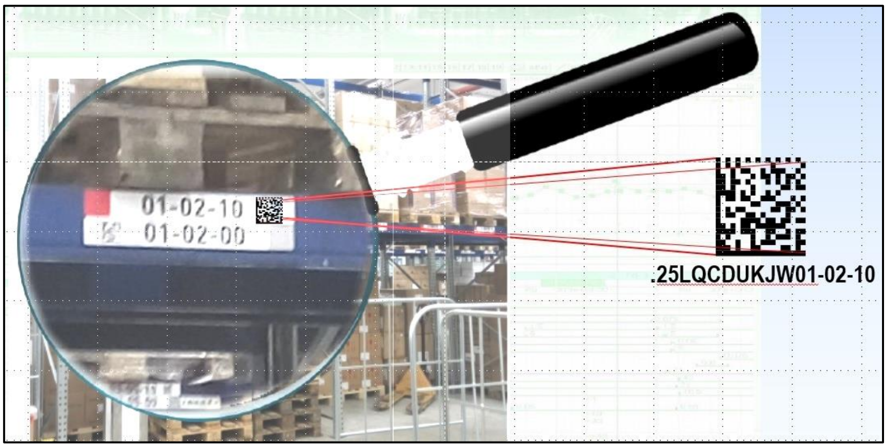


Figure 6 - Stock location code (source: Klinikum of the Friedrich Schiller University of Jena (UKJ), Germany)

Data sequence of the example Fig. 6)
.25LQC DUKJW01-02-10
Stock location of UKJ: W01-02-10
CIN: DUKJ
IAC: QC
ASC Data Identifier: 25L
System Identifier for the ASC DI structure: "**.**"

4.3 Example with two concatenated data elements

Figure 7 shows two concatenated data elements.

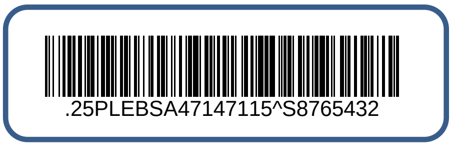
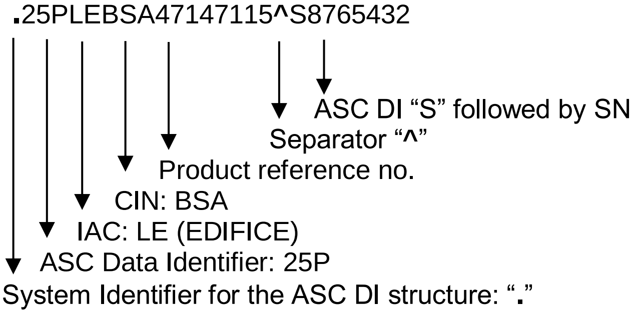


Figure 7 - Example Code 128 with two ASC DI headed data elements concatenated by separator "^"

Data sequence of the example Fig. 7)



.25PLEBSA47147115**^**S8765432

ASC DI "S" followed by SN

Separator "**^**"

Product reference no.

CIN: BSA

IAC: LE (EDIFICE)

ASC Data Identifier: 25P

System Identifier for the ASC DI structure: "**.**"

ANNEX A: (INFORMATIVE) FLAG CHARACTER UNAMBIGUITY

This guideline defines the "." character as Flag Character to identify data elements applied with ASC Data Identifiers.

It is known that the "." is already used in some legacy applications as first character of encoded data.

Typically these legacy encodings are internal applications that encode numbers with a leading zero, without that any global standard for the semantical layer is used.

These non-standard legal applications must have their own mechanisms to ensure that only their internal data are processed. Not following standards does not give any reason to expect that no conflicting cases might appear.

Users decoding data according to this guideline have to be aware that the "." may show up not as a Flag Character but as non-standard encoding. An additional check that the characters following the "." do correspond to a ASC MH10 Data Identifier relevant for the application will help to filter out many non-applicable cases: a small risk of errors will remain.

However the reliability of the detection of data encoded with ASC MH10 Data Identifiers is very much better than without the flag character.

Using Flag Characters such as "+", "." and "FNC1"does not give the same level of unambiguity than using the rules of ISO/IEC 15434, but it is a compromise between unambiguity and ease of application.

ANNEX B: (INFORMATIVE) INNER SEPARATOR CHARCTERS

Some ASC DIs define "inner data element separators" that structure data elements within the data element. Often the "+" is defined for that purpose in ANS MH 10.8.2.

In other cases the content data include special characters to structure this content. For example, in URLs the "@", the ":" and the "/"-characters are used to structure the data.

The character "^" (Circumflex) has been selected to separate data elements from each other in order to avoid conflict with other characters that are used as inner data element separators.

It is safe to use these inner separator characters in conjunction with "^" (Circumflex) as data element separator. See this example ("<" and ">" are used for illustration purposes only) :

<.><DI1><Data element 1><field 1><+><field 2><^><Data element 2>