

# **AutoID URL**

**Version 1.2**

**URL representation of standard AIDC  
messages using ASC data identifiers for  
smartphone bar code reading and NFC  
technology**

AutoID URL specification with demonstrator  
r2023-10-24

---

Eurodata Council Institute e.V.(EDCi), Kösener Str. 85, 06618 Naumburg  
phone +49(0)3445 781160, mail: [info@e-d-c.info](mailto:info@e-d-c.info), [www.e-d-c.info](http://www.e-d-c.info)

## Publication Summary

- Title:** **AutoID URL**  
URL representation of standard AIDC messages using ASC data identifiers for smartphone bar code reading and NFC technology
- Author (s):** Joint Working Group ADC, demonstrator and samples by ELMICRON
- Issue, date:** Issue 1.0, published by Eurodata Council Institute e.V. (EDCi) 2022-12-13  
Issue 1.1, published by Eurodata Council Institute e.V. (EDCi) 2023-02-27  
Issue 1.2, published by Eurodata Council Institute e.V. (EDCi) 2023-10-24
- Terms:** Copyright © EDCi 2022  
All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means without prior permission. Notwithstanding the fact that utmost care has been applied in the collecting, drawing up and formulating of the content, the authors can under no circumstances be held liable for errors, omissions or misinterpretations as a result of the information compiled in this specification.
- Abstract:** The AutoID URL specification is a complementary option to 'DIN 16598 Syntax for WEB & keyboard compatible encoding with ASC Data Identifiers' and is suited for smartphone barcode reading and NFC compatible applications. Different to the standard encoding principle "UID with ASC DI first", the AutoID URL begins "URL first". The AutoID URL with its included attributes enables simple extraction of the data remotely by a WEB server as well as by scanners with local apps or attached to ERP systems (dual use). The AutoID URL specification includes an optional digital signature for applications where authentication is appropriate.
- References:**
- ANS MH10.8.2 Data Identifier  
<https://my.mhi.org/s/store#/store/browse/detail/a153h0000051JuRAAU>
  - DIN 66403 System Identifiers <<https://www.beuth.de/en>>
  - DIN 16598 Syntax keyboard and Web compatible encoding of data elements in machine readable symbols applied with ASC Data Identifiers < <https://www.beuth.de/en> >
  - IEC 61402-2 Identification Link - Part 2: Encoding of product types, lots, batches and characteristics
  - ISO/IEC 20248 Digital signature meta structure
  - RFC3986 Uniform Resource Identifier (URI): Generic Syntax <https://www.rfc-editor.org/rfc/rfc3986>
  - RFC1738 Uniform Resource Locators (URL) < <https://www.rfc-editor.org/rfc/rfc1738> >
  - UTF-8 Character Set for encoding URL query string values < <https://en.wikipedia.org/wiki/UTF-8> >
- Applications:** Where best practices for access to ERP systems are defining the use of a standard syntax featuring "UID first" in ASC DI format, the AutoID URL Specification "URL first" is recommended for applications where the code has to be read by smartphones or devices with internet access, and relevant information shall be opened via the browser. RFID applications can also use NFC technology. Due to the functionality using ASC DIs for data element identification, AutoID URLs enable "dual applications" for processing locally for ERPs, as well as remotely by linking to a WEB site for processing by the WEB server and responding a WEB page as well. For applications with additional security requirements the option "Digital Signature" applies.

## Maintenance history

Date	Action	content
2022-12-13	Initial Publication	Issue 1.0
2023-02-24	Update to version 1.1	URL domains for the AutoID URL codes changed from small in capital letters for code size reduction, new section (4.3) added: "4.3 Biuniqueness of the URL string and canonical representation".
2023-10-24	Update to version 1.2	ISO/IEC 20248 Digital Signature option added. Redundancy in URL taken out: "/" before "?"

## Table of Content

Table of Content.....	3
1. Problems to Solve .....	4
2. Scope .....	5
3. Definitions.....	5
4. AutoID URL Definition and Rules .....	6
4.1 General Structure of an AutoID URL.....	6
4.2 Examples of AutoID URLs .....	6
4.3 Symbology for AutoID URL .....	7
4.4 Print quality of 2d-Symbols.....	7
4.5 Uniqueness through WEB domain and ISO/IEC 15459 rules .....	7
4.6 Biuniqueness of the URL string and canonical representation.....	8
4.7 Use cases of Dual application for WEB server and ERP local .....	8
4.8 Option Digital Signature for coded data verification.....	9
5. Summary .....	11
6. Annex Example of an AutoID URL QR code of a battery.....	12
7. Annex Encoding and Decoding Flows .....	12
7.1 Example Encoding Flow .....	12
7.2 Example Decoding Flow for local use (ERP) .....	12
7.3 Decoding example of the AutoID URL fig. 5 for local use and feeding the data elements in the fields of an ERP application.....	13
7.4 Example Decoding Flow WEB server use.....	13
8. Annex AutoID URL demonstrator.....	15
8.1 Example of an Auto ID URL applied to a battery and its digital twin in a WEB server .....	15
8.2 Example "Motor" with an AutoID URL with digital twin at WEB server .....	16
8.3 Example AutoID URL for an "Instrument" .....	17

## 1. Problems to Solve

In traditional AIDC applications that are used with ERP systems, data elements identified by ASC Data Identifiers and encoded in traditional syntax cannot be used for direct access to the WEB, because the data string does not build a URL. If a link to a WEB based application is required, a syntax according to 'URL' characteristics and rules is needed. Table 1 shows examples of elements encoded in AIDC media in the traditional way, i.e. without the potential to link to the WEB.

Table 1 – Examples of non-WEB compatible standard strings for transmission to ERP systems

Syntax	Flag	ASC DI "unique product ID"	Value, e.g.	Separator	ASC DI "serial number"	Value, e.g.	Terminator
a) ISO/IEC 15434	[>rs06gs	25P	LEIBMPROD4711	gs	S	1234XYZ	rs eot
ASC data string for encodation a) [>rs06 gs 25P LEIBMPROD4711 gs S1234XYZ rs eot							
b) DIN 16598 WEB&KB compatible syntax	.	25P	LEIBMPROD4711	^	S	1234XYZ	
ASC data string for encodation b) .25PLEIBMPROD4711^S1234XYZ							

For design of WEB compatible codes for scanning and processing by smartphones, URL syntax must be applied and its permitted characters be used.

The problem to solve is to merge Data elements identified by ASC Data Identifiers in the syntax compatible with URL rules in a single data string, so that the data can be read by smartphones using barcode and NFC technology with a single scan.

Table 2 shows an example of the generic syntax of a URL compliant message that carries additional elements that use ASC Data Identifiers.

Table 2 – Example of a WEB compatible design for transmission to the internet

Syntax	Flag/URL root	ASC DI "unique product ID"
RFC1738 URL	https://srv.de/ART?	Concatenated data elements identified by ASC DIs
<b>Example RFC 1738 compatible data string:</b> https://srv.de/ART?.DI=value&.DI=value&etc.		

The AutoID URL definition is enabling encoding of data in a RFC-WEB and NFC-RFID compatible way (= URL first), using ASC data identifiers in compatibility to standard barcode syntax, putting in practice the paradigm:

"Same data elements – different syntax."

## 2. Scope

The scope of this specification is the definition of a unique encoding scheme for data elements applied with ASC Data Identifiers for use in applications where smartphones with barcode readers and/or devices with NFC technology are primarily intended for use, but also for dual use through WEB applications as well as locally by ERP systems, and where same unambiguity is required like in traditional syntax using ASC DIs.

This specification shall enable parsing of encoded data elements applied with ASC DIs out of an URL similar to the functionality of standard ASC codes using ISO/IEC 15434 format "06" or DIN 16598 Syntax for Web and keyboard compatible encoding with ASC Data Identifiers.

The AutoID URL shall provide to applications with the functionality for direct access from an item to a WEB destination, like the "Digital Product Passport (DPP)". An optional digital signature shall enable verification for applications with increased security requirements.

This guideline is not limited to a specific industry sector or subset of ASC Data Identifiers.

## 3. Definitions

Specific terms and Abbreviations used for this specification:

ASC DI	ASC Data Identifier according to ISO/IEC 15418, part ANS MH 10.8.2 Data Identifiers
AutoID URL	An AutoID URL applied with min. one ASC Data Identifier with ISO/IEC 15459 sequence with Issuing Agency Code and CIN issued by an ISO/IEC 15459 Issuing Agency supporting ASC DIs like EDCi.
Flag	Flag character, header of data strings identifying the syntax being used, e.g. "[>rs" or "." for codes with ASC DIs, "+" for the Health Industry Bar Code (HIBC), "FNC1" for the Global Standards One (GS1) Application Identifier syntax, "=" for the ICCBBA ISBT 128 system, "https://www" for URLs according to RFC syntax, etc.
NFC	Near Field Communication, RFID option according to ISO/IEC 14443.
SI	System Identifier for data strings composed of ASC data identifiers, see DIN 66303 and DIN 16598.
SIDI	Combined System Identifier (SI) for ASC DIs with Data Identifier (DI): Dot "." and adjacent ASC data identifier as a set "DOT DI", e.g. ".25P"; see "query string parameter" below (see also terms IEC 61406-2).
URL root	Leading segment of an AutoID URL composed of the URL components URL protocol HTTP(s), domain name and question mark.
URL path	Segment of the URL root including the ASC DI headed data elements targeting to information inside the WEB server related to the whole AutoID URL.
Query string separator	The question mark "?" placed between URL root and URL path
Query string parameter	See SIDI, Dot with ASC DI, like ".1P" for product no., ".S" for serial number, etc.
Query string value	Value following the query string parameter (SIDI) with data

## 4. AutoID URL Definition and Rules

### 4.1 General Structure of an AutoID URL

An AutoID URL is composed of the URL protocol and a WEB domain name applied building the URL root, followed by a question mark "?" and followed by the query value of data elements identified by ASC DIs and data value. Each data element is starting with the SIDI "•"(Dot + DI)) followed Equal sign "=", followed by the value. Concatenated data elements are separated by the Ampersand "&". UTF-8 coding is used for the value.

Table 3 – Structure of an AutoID URL

<u>URL Protocol http(s) + WEB domain</u>	<u>Question mark</u>	<u>SI</u>	<u>DI</u>	<u>Equal sign</u>	<u>Value</u>	<u>Ampersand</u>					
	?	•	DI	=	value	&	•	DI	=	Value	etc.
URL root		Query value, first data element				2 <sup>nd</sup> data element (optional)			etc.		
AutoID URL											

### 4.2 Examples of AutoID URLs

#### 4.2.1 **AutoID URL including the data elements unique product number and expiration date:**

URL root: [HTTPS://SRV.DE/ART/](https://srv.de/art/)

ASC DI for unique product ID: 25P

Data value: LEIBMPRODX12 (sequence <IAC><CIN><Product no.>)

followed by next data element with:

ASC DI for expiration date: 14D

Data value: 20241231 (sequence <yyyymmdd>)

Table 4 – Example of an AutoID URL applied with data element values above

<b>URL root</b>	<b>?</b>	<b>•</b>	<b>DI</b>	<b>=</b>	<b>value</b>	<b>&amp;</b>	<b>•</b>	<b>DI</b>	<b>=</b>	<b>Value</b>
<a href="https://srv.de/art/">HTTPS://SRV.DE/ART</a>	?	•	25P	=	LEIBMPRODX12	&	•	14D	=	20241231

Fig. 1 shows the AutoID URL example of table 4 encoded with QR Code.



Fig. 1 QR Code carrying the AutoID URL of Table 4.

Note: Capital letters are preferred for the URL root. In comparison with small letters it reduced the code sizes.

#### 4.2.2 **AutoID URL including data elements unique product number, expiration date, LOT and SN:**

URL root: <https://srv.de/art/>

ASC DI for unique product number: 25P

Value: LEIBMPRODX12 (sequence<IAC><CIN><Product no.>)

followed by next data element with

ASC DI for expiration date: 14D

and data value: 20241231 (sequence <YYYYMMDD>)

Followed by data element with

ASC DI for LOT/BATCH 1T and data value: PXBL

ASC DI for Serial no. S and data value:2BB3

Table 5 – Example of an AutoID URL applied with data element values above

URL root	?	•	DI	=	value	&	•	DI	=	Value	continued
HTTPS://SRV.DE/ART	?	•	25P	=	LEIBMPRODX12	&	•	14D	=	20241231	
Continued →	&	•	1T	=	PXBL	&	•	S	=	2BB3	

Fig. 2 shows the AutoID URL example of table 5 with product ID (ASC DI "25P") and LOT (1T) and Serial number (S) encoded with Data Matrix (see decoded data string fig. 7).



[HTTPS://SRV.DE/ART?.25P=LEIBMPRODX12&.14D=20241231&.1TP=XBL&.S=2BB3](https://SRV.DE/ART?.25P=LEIBMPRODX12&.14D=20241231&.1TP=XBL&.S=2BB3)

Fig. 2 Data Matrix Code carrying the AutoID URL of Table 5.

### 4.3 Symbology for AutoID URL

As symbology, any applicable ISO/IEC standard symbology is suitable for AutoID URLs, but recommended symbologies are ISO/IEC 18004 QR Code, ISO/IEC 16022 Data Matrix and ISO/IEC 21471 DMRE as the preferred setting for smart phones but also NFC standards in case of Radio Frequency Identification (RFID).

*Note: Lower case characters like "http" may take more code space then capital characters like "HTTP". Therefore, it is recommended to use capital letters to keep the symbol as small as possible.*

### 4.4 Print quality of 2d-Symbols

Print quality of printed 2d-symbols measured according to ISO/IEC 15415 shall result in a quality grade not lower than grade "C". Direct part marked (DPM) symbols measured according to ISO/IEC 29158 shall result in a quality grade not lower than grade "C 2.4" Continuous Grading for ISO/IEC15415 Parameters".

### 4.5 Uniqueness through WEB domain and ISO/IEC 15459 rules

Common standards for marking and identification take ISO/IEC 15459 rules as basis for uniqueness and identifying responsible parties for objects and items. Accredited Issuing Agencies (according to ISO/IEC 15459-2) supply Company Identification Numbers (CIN) for labelers, and the labeler applies/adds its own reference. This is defined by ISO/IEC standard ISO/IEC 15459 Unique Identification.

The AutoID URL is opening an additional path to uniqueness by the WEB domain.

Accordingly, responsible parties can be identified by:

- 1 ASC DI with ISO/IEC 15459 sequence <IAC><CIN><data value>, like ASC DIs "18V", "25P", "25S", etc.
- 2 WEB domain, e.g. by help of access to the imprint of the WEB site

For unique identification of items, priority shall be to use ASC DIs in ISO/IEC 15459 structure (IAC, CIN, Data) which includes the party ID responsible for the item. Otherwise the WEB domain name shall be used as party identifier. In this case the web domain must be stored together with the data.

**Note:** to ensure consistent data processing using data identifiers, it is planned to request a data identifier for unique party identification by a web domain. As an alternative path, ISO 15459 may be extended to support web domains.

#### 4.6 Biuniqueness of the URL string and canonical representation

In some applications, the URL string may be used as a bi-unique ID. For example, IEC 61406-1 has the requirement, that the URL string shall not be modified.

To be compliant to those applications, the URL string shall not be modified, once assigned to an item.

In general an URL is not biunique by itself as some modifications lead to the same web site.

Those modifications include case change of case invariant parts (protocol, host name), percent decoding, punycode decoding, fragment removal. Each web-server may have additional tolerances like: case change in path (typically on Windows OS servers), query string parameter order change, shortcuts implemented by rewrite rules, tiny URL services, addition of anyway ignored query parameters.

To comply to those applications, the following rules may be used. URL strings in compliance to those rules are called canonic URL:

- Don't change the URL string once attributed to an item in any mean. Take precaution when storing or communicating the URL string.
- Use upper case to any case-invariant part
- Avoid unneeded parts: fragment, login, password, unneeded query parameters
- Order the SIDI parameters in natural form from general to detail scope. Example: Manufacturer (.18V), Product (.1P), Serial number (.S), manufacturing date (.16D). The order may be specified by an appropriate application guideline. This is common within data identifiers in general.

#### 4.7 Use cases of Dual application for WEB server and ERP local

The AutoID URL enables "dual application", meaning it can be used locally for capturing an AutoID URL by online scanner for processing by an ERP system and/or scanned by smartphones for transmitting the AutoID URL to WEB server responding with the AutoID URL relevant WEB page.

##### 4.7.1 AutoID URL scanned by smartphone for response by a WEB server.

AutoID URL is scanned by a smartphone device or other devices with internet browser, transmitting the original AutoID URL straight to the destination, e.g. to a manufacturers WEB server. The server extracts the ASC DI headed data elements and responds with the WEB page with the relevant item information identified by the product code and its related data like Expiry Date, LOT, SN, etc. The WEB page might lead to instruction information, data sheets or even to status information.

Fig. 3 illustrates the function of scanning an AutoID URL by smartphone, opening an item relevant WEB page by the browser of the smartphone (or any other device).

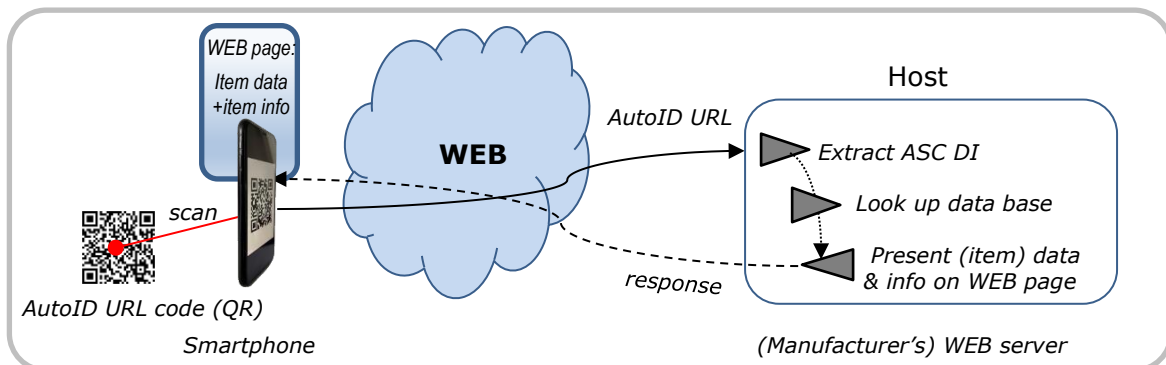


Fig. 3 AutoID URL scanned by smartphone for opening a web page presenting (manufacturers) item/product relevant information



#### 4.7.2 **AutoID URL scanned locally for processing by an ERP system**

In local scanning applications the ASC DI headed data elements will be extracted out of an AutoID URL prior to processing by a host computer (ERP) and its application software, e.g. for "goods receipt". This extraction might be done by the scanner with extraction app feature or by simple scanners and an extraction app that is running on a work station connected to the host. Identifying the data elements is handled off-line, and there is no WEB connection required.

Fig. 4 illustrates an AutoID URL scanned locally for data processing by the ERP system.

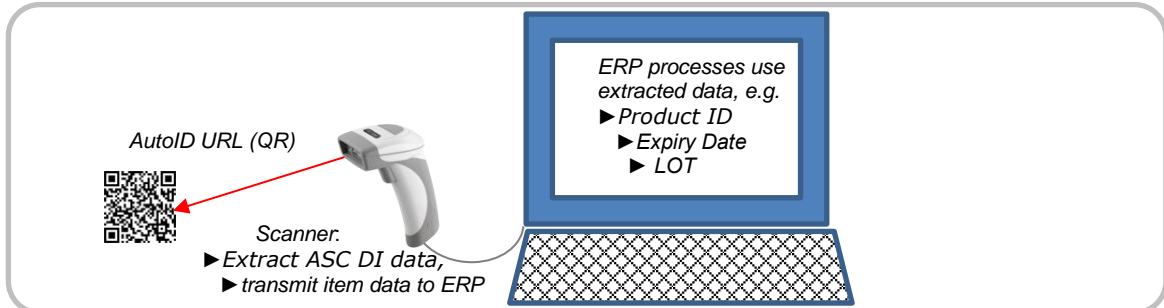


Fig. 4 AutoID URL scanned locally for processing the item/product data by ERP system

### 4.8 **Option Digital Signature for coded data verification**

Applied with a DigSig, AutoID URL codes can be verified in regard of authentication. ISO/IEC 20248 (DigSig) is one solution for it and recommended. ISO/IEC 20248 is an X.509 application standard utilising digital certificates and signature with the supporting public key infrastructure (PKI), the de facto standard for Internet services. ISO/IEC specifies a digital signature envelope (a DigSig), a language to place data field values in the envelope, a method to place the data description in the digital certificate, a digital certificate repository access method, revocation of certificates and DigSigs, and interfaces for the encoding-generations and decoding-verification of DigSigs. It has been designed to work with any combination of barcodes, RFID and other data carrier, including human entered data.

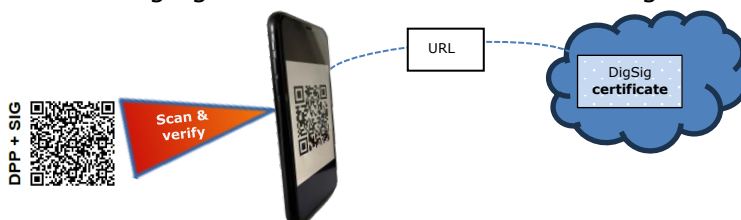
DigSigs are designed to be a standalone (offline) data structure for barcodes and RFID, or an add-on to any of data carrier messages supporting ANSI MH 10.8.2 Data Identifiers (DIs) and other identifier systems.

Illustration of offline (a) and online functionality (b) for verification/decryption:

- a) Offline: The DigSig certificate is available in the reading device



- b) Online: The DigSig certificate will be reached through the WEB



In codes structured with DIs, the DigSig is preceded by DI "6R". Flagged by DI "6R" the data string is clearly identified as a DigSig. The add-on method allows for hybrid environments, i.e. systems which can and can not handle DigSigs works seamlessly and non-DigSig-enabled systems can be enabled at a later stage. The DigSig verification remains to be an optional action as determined by the reading application.

The DigSig for a specific code might be generated by software tools or by a DigSig service provider.

A trusted service provider must host the certificate and revocation lists for it.

In this example the DigSig encoding is represented in Base54 URL as specified for the DI "6R". The DigSig Data Description contains a directive that the primary data carrier is a DI-based message. Which means that the full message is signed.

#### **4.8.1 DigSigs message generation**

a. Construct the message:

`HTTPS://WWW.E-D-C.INFO/AUTOID?.25P=QCELIMIBATT01X&.S=123456XYZ`

b. Sign the message: `DigSigEncodeGenerate("HTTPS...123456XYZ")`

→ `"Oo586eJAMEYCIQCf31EqIJMLGclBpHLIRgBdOxYz1230e0o49"`

c. Add the DigSig to the message.

*Note 1 It can be added anywhere.*

*Note 2 Any change to the message will be rejected by the verification, including the URL. This prevents splicing attacks.*

→ `"HTTPS....123456XYZ&.6R=Oo586eJAMEYCIQCf31EqIJMLGclBpHLIRgBdOxYz1230e0o49"`

d. Program the message in the data carrier.

#### **4.8.2. DigSigs message reading**

4.8.2.1 Reading at an application which does not know of the DI 6R

a. Read the message from the data carrier.

b. Optionally pass the message to the data stack after either:

b.1. Stripping DI 6R and its value (the DigSig)

because the rest of the data stack may be confused.

b.2. Keeping DI 6R, since other parts in the data stack may know what to do with 6R.

c. Use the message data locally ignoring 6R.

4.8.2.2 Reading at an application which knows of the DI 6R:

a. Read the message from the data carrier.

b. The application notes the DI 6R. It now either:

b.1. Ignore the 6R DI, as such does not do a verification.

Optionally pass the message to the data stack after either:

b.1.1 Stripping DI 6R (and the DigSig) it

because the rest of the data stack may be confused.

b.1.2 Keeping DI 6R,

since other parts in the data stack may like to verify the message.

b.1.3 And, then use the message data.

b.2 Does a verification by

b.2.1. Split the message in two: The message with DI 6R and the DigSig:

E.g. `"HTTPS.....123456XYZ"` and

`"Oo586eJAMEYCIQCf31EqIJMLGclBpHLIRgBdOxYz1230e0o49"`

*Note the part "&.6R=" is deleted.*

b.2.2. Send the two parts to the DigSig decoder verifier

with the DigSig as the first data element.

`DigSigDecodeVerify("Oo586e...49", "HTTPS...123456XYZ")`

→ The Signature is Accepted or Rejected.

b.2.3. The application then:

Use the trusted data from the verified message.

Optionally pass the trusted message to the data stack, with or without the DI 6R.

*Note: Details on this process can be found in Annex K of ISO/IEC 20248.*

## 5. Summary

The AutoID URL is combining direct WEB access through smartphones with unique identification of data elements using ASC Data Identifiers as practiced with standard codes used in global supply chain management. The URL root links to the WEB portal and the attributes, like product part number and/or serial number, may direct to the individual path inside portal or cloud. Accordingly, the AutoID URL can be used as a tool for tracking and tracing of items, but also to open access to item specific information.

Where smartphones with barcode and/or NFC technology may process AutoID URLs by default, existing ERP interfaces might need a conversion of the AutoID URL WEB syntax to fit to the ERP interfaces. Nevertheless, the use of the given ASC Data Identifiers eases the conversion and keeps the data elements unique.

In addition to transmission to WEB destinations determined by URL root and attributes for response with specific data information or dialog, the AutoID URL would enable offline parsing of the encoded data elements if functional Apps are provided. It is also performing for dual use e.g. a) access to WEB server and b) for local ERP processing. Where an EDCi AutoID URL includes the ISO/IEC 15459 Issuing Agency Code of EDCi "QC" and by EDCi issued Company IDs (CINs), other Issuing Agencies may support use of same specification but with their own Issuing Agency Code and CIN. The optional digital signature enables verification of data authenticity.

*Note: An alternative solution to "URL first" for combined WEB access with unique identification by ASC DIs is the use of the DI "34L - Pointer to Process" solution "UID first" where the URL will be converted after scanning if additional WEB access is appropriate.*

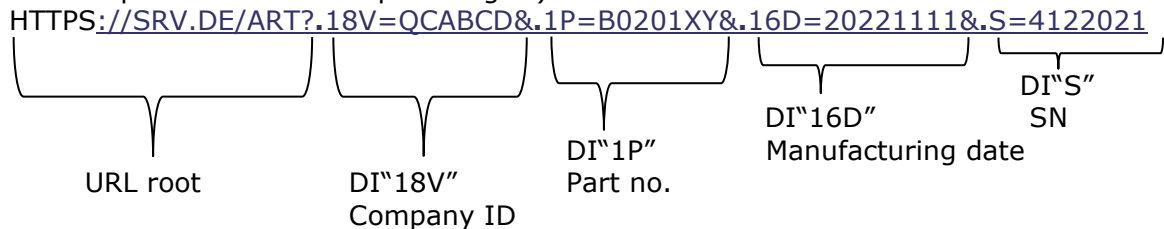
## 6. Annex Example of an AutoID URL QR code of a battery

The example in fig. 5 shows a backup battery applied with an AutoID URL encoded in QR Code and carrying URL root, unique company ID, product part number and attributes manufacturing date and serial number.



Fig. 5 Backup battery applied with AutoID URL in QR Code

Data sequence of the example in Fig. 5)



Note: The example contains an ISO/IEC 15459 ASC DI, this enables unique identification of the responsible party both by the WEB domain specifically for WEB servers and by the ASC DI (here "18V" with IAC "QC" and CIN "ABCD" registered by EDCi) for local ERP use.

## 7. Annex Encoding and Decoding Flows

### 7.1 Example Encoding Flow

An example for encoding flow is:

1. Ensure that all data elements are composed of URL conforming characters and data elements headed by the appropriate ASC Data Identifiers.
2. Take the URL root as first data element and add a question mark prior to the first SIDI and value, if applicable.
3. In case of additional data elements, put the Ampersand as separator between the next data element consisting of SIDI plus value.
4. Repeat step 3 until no further data elements are to be encoded.

### 7.2 Example Decoding Flow for local use (ERP)

An example for decoding flow for local use is:

- a) Read entire string of characters encoded in the symbol
- b) Check whether the first character of the encoded data string is a WEB domain and structure of the full URL is compliant to the AutoID URL.
- c) If embedded item information shall be displayed or sent to ERP data bases, parse each SIDI (consisting of DOT and ASC DI) with its value out of the AutoID URL and process them according to the application, e.g. product ID with expiration date, LOT, SN, etc. using interpretation of the Data Identifiers (first 1...4 characters after the Dot) with its data element definitions of ASC MH10.8.2.
- d) If information related to the AutoID URL shall be accessed through the WEB, send the full URL string to the WEB

### 7.3 Decoding example of the AutoID URL fig. 5 for local use and feeding the data elements in the fields of an ERP application

The functionality of scanning and decoding the raw data of the AutoID URL example and extracting the data elements for handover to an ERP application is shown by Fig. 6.

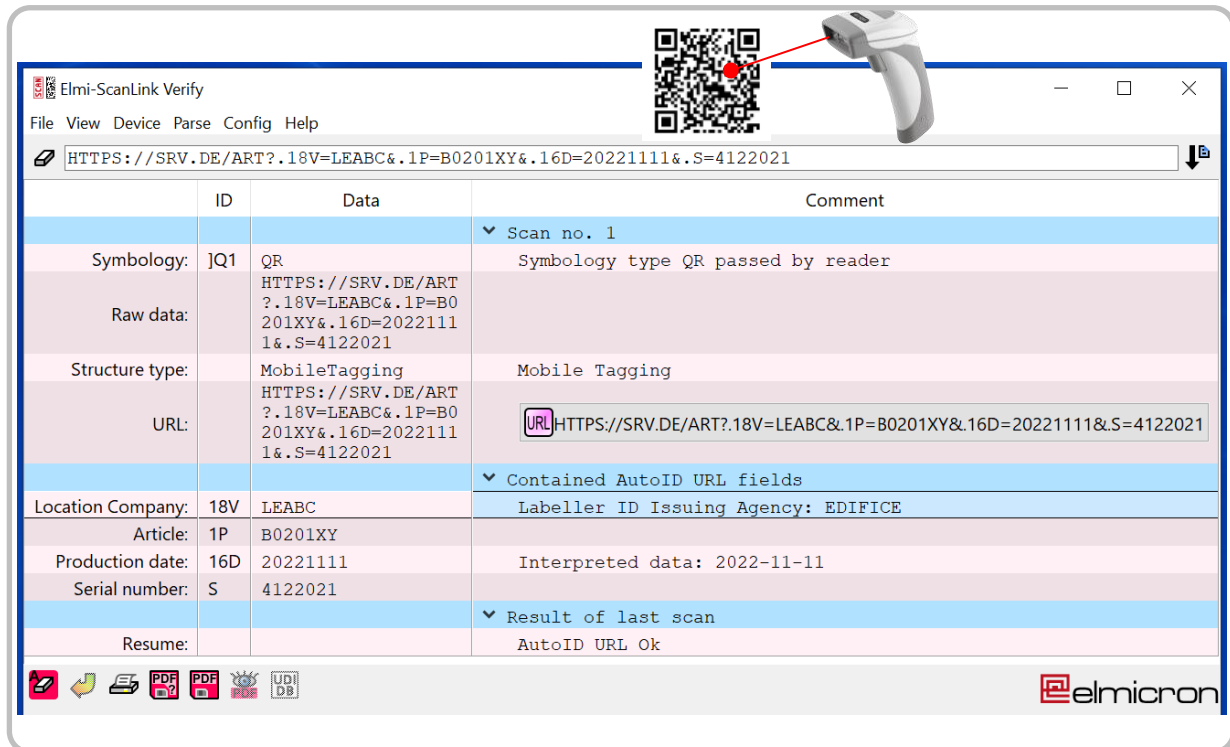


Fig. 6 Decoding screen of AutoID URL fig. 5 displaying the data elements LABELER, EXPIRY DATE, LOT and SN ready for processing by an ERP application.

Note: Fig. 6 entry field of the screen begins with "JQ1", which is the symbology ID for QR Code as carrier.

### 7.4 Example Decoding Flow WEB server use

If an AutoID URL is sent to the web-server, the parsing may be executed within a user script authored in PHP (= Hypertext Preprocessor). A manufacturer is able to know the structure of its codes, but, a check for certain DI's is appropriate.

In this example, the manufacturer uses DI "25P" for the product number and "S" for the serial number.

Here are the corresponding encoding steps:

#### 1. Detect (my) AutoID URL

In the first step, the URL query string is checked for parameter ".25P" and ".S". This may be done by the following single PHP line:

```
>if (is_string($_GET["_25P"]) && is_string($_GET["_S"])) {
```

Remark, that the "." is replaced by "\_" within PHP.

## Example Decoding Flow WEB server use

- continued -

### 2. Security check of the data

Each web server should take caution of any data coming from the outside to avoid SQL injection (DB lookup) or script injection (data presentation) attacks.

In our case, both issues may arise.

In this example, we check for alphanumerical data:

```
$ProductCode = $_GET["_25P"];  
$Serial = $_GET["_S"];  
if ( ctype_alnum($ProductCode) && ctype_alnum($Serial) ) {
```

### 3. Data Base lookup of the scanned product data

Now, the given parameter may be checked for status within the company's data base. For this aim, first the product code and serial number is saved in own variables:

Now, the data base lookup may be done:

```
$State = DBLookup($ProductCode,$Serial);
```

The resulting state is saved in the variable "State".

### 4. Show on Web site

The resulting status may be shown on the resulting web site:

```
<table class="analyse"><caption>Item information</caption>  
<tr><td width=50%>Product Code</td><td><?php echo $ProductCode;?></td></tr>  
<tr><td>Serial number</td><td><?php echo $Serial;?></td></tr>  
<tr><td>Status</td><td><?php echo $State;?></td></tr>  
</table>
```

As a result, the state will pop up.

The practical result may be seen by invoking the following URL:

<HTTPS://WWW.E-D-C.INFO/AUTOID?.25P=QCELMIBATT01X&.S=4221117>

Scan response of the example can be checked at ANNEX AutoID URL DEMONSTRATOR chapter 7.1.

## 8. Annex AutoID URL demonstrator

For showing an example of a life application of AutoID URLs a demonstrator has been prepared in the cloud. The following AutoID URL examples will lead to the individual WEB pages dedicated to product and serial number of a product. Scanning the AutoID URL examples by smartphone or other devices with browsers will open the dedicated WEB page. The WEB page examples are displayed as well for illustration if no WEB is available.

### 8.1 Example of an Auto ID URL applied to a battery and its digital twin in a WEB server

Example AutoID URL at fig. 7 will lead to the WEB page of the dedicated item (battery). Scan the code or key in the AutoID URL string into a browser.




Fig. 7 Demonstration example of an AutoID URL encoded in ISO/IEC 18004 QR for opening the related WEB page by URL root, product number and serial number.

Scan the AutoID URL fig. 8. or key in the entire string in a browser and as response the WEB page will be opened with product and serial number and additional information as shown in fig. 8.

### AutoID URL Demo

*Item information*

Product Code	QCELMIBATT01X
Description	6V Power Battery
Image	
Serial number	4221117
Status	Charged
Location	On stock 4

See the [Information page](#) for a sample codes and PHP source code of this page.  
Service by [EURODATA Council Institute e.V.](#)

Fig. 8 WEB page processed by the WEB server in response to the scanned AutoID URL fig.8.

The WEB page fig. 9 displays the encoded product code and serial number of the scanned battery and in addition to it the server is adding an image of the product, the status "charged" and the stock position where the battery is registered for.

### 8.2 Example "Motor" with an AutoID URL with digital twin at WEB server

Example AutoID URL of fig. 9 will lead to the WEB page of the dedicated item (motor). Scan the code or key in the entire string into a browser.



Fig. 9 Demonstration example of an AutoID URL encoded in ISO/IEC16922 Data Matrix for opening the related WEB page displaying the data of a motor and prepared information.

Scanning the AutoID URL of fig. 9 by smartphone or keying the string in a browser will get the response shown at fig. 10.

## AutoID URL Demo

*Item information*

Product Code	QCELMIAQ7B4
Description	Pump engine
Image	
Serial number	400D02
Status	Active
Location	Filling Station 2

See the [Information page](#) for a sample codes and PHP source code of this page.

Service by [EURODATA Council Institute e.V.](#)

Fig. 10 WEB response after scan of the AutoID URL of fig. 9, displaying data and status "active" and location where the motor is registered and expected to be.



### 8.3 Example AutoID URL for an "Instrument"

Example of fig. 11 will lead to the WEB page of the dedicated item (medical instrument).

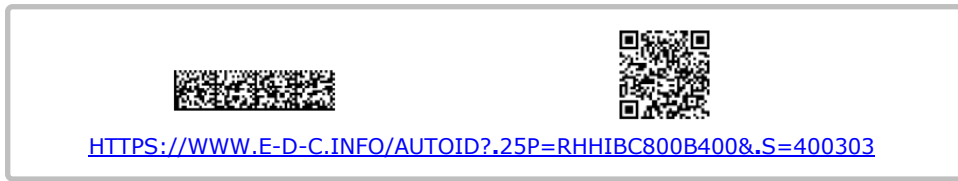


Fig. 11 AutoID URL on a medical instrument encoded in ISO/IEC ISO/IEC 21471 DMRE and ISO/IEC 18004 QR

Scanning AutoID URL of fig. 11 by smartphone or keying the string into a browser will get the WEB response shown at fig. 12.

**AutoID URL Demo**

*Item information*

Product Code	RHHIBC800B400
Description	Forcipes E4
Image	
Serial number	400303
Status	Repair
Location	Repair Dept. 7

See the [Information page](#) for a sample codes and PHP source code of this page.  
Service by [EURODATA Council Institute e.V.](#)

Fig. 12 WEB response screen after scan of the AutoID URL of fig. 11 displaying product reference and serial number and also description, status ("repair") and location where the instrument should be.

Note: AutoID URL Demonstrator has been supplied by [www.elmicron.de](http://www.elmicron.de)