TECHNICAL REPORT

ISO/IEC TR 29162

First edition
2012-07-15

Information technology — Guidelines for using data structures in AIDC media

Technologies de l'information — Directives pour l'usage des structures de données dans des medias d'AIDC

Reference number
ISO/IEC TR 29162:2012(E)

Contents

Page

Foreword

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

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

In exceptional circumstances, when the joint technical committee has collected data of a different kind from that which is normally published as an International Standard ("state of the art", for example), it may decide to publish a Technical Report. A Technical Report is entirely informative in nature and shall be subject to review every five years in the same manner as an International Standard.

Attention is drawn to the possibility that some of the elements of this Technical Report may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC TR 29162 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 31, *Automatic identification and data capture techniques*.

Introduction

Radio frequency identification (RFID) is one of the AIDC media widely used in the market place. Linear bar codes and two-dimensional symbols have long utilized AIDC media. The international standard for AIDC syntax is ISO/IEC 15434. ISO/IEC 15961 and ISO/IEC 15962 were developed as encoding rules for RFID.

Users have long utilized linear bar codes and two-dimensional symbols for item identification and numerous RFID technologies have recently been developed. Users who want to utilize RFID transponders should consider compatibility with linear bar codes and two-dimensional symbols already in the system. Because of the growing diversity and complexity of AIDC media in the market place, especially in RFID, it is not easy for users to understand how to read and write their data to each application of AIDC media.

This Technical Report explains common data structures used in both optically readable media (linear bar codes and two-dimensional symbols) and radio-frequency identification. It primarily addresses the use of ASC MH10 Data Identifiers to provide the semantics, ISO/IEC 15434 to provide the syntax, and ISO techniques of unique item identification with ISO/IEC 15961 Application Family Identifiers (AFIs) and encoding rules for RFID using ISO/IEC 15962.

Those interested in applications using Air Transport Association (ATA) SPEC 2000, Text Element Identifiers, are encouraged to contact the ATA for specific guidance.

Those interested in applications using GS1 Application Identifiers and EPC, specifically for material found in the EPC Tag Data Standard (TDS), are encouraged to contact GS1 for specific guidance.

Information technology — Guidelines for using data structures in AIDC media

1 Scope

This Technical Report provides guidance on the use of AIDC media (e.g. linear bar codes, two-dimensional symbols, RFID transponders) in the supply chain.

2 Normative references

The following referenced documents are indispensable for the application of this Technical Report. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 19762 (all parts), *Information technology — Automatic identification and data capture (AIDC) techniques — Harmonized vocabulary*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 19762 (all parts) apply.

4 Abbreviated terms

For the purposes of this document, the abbreviated terms given in ISO/IEC 19762 (all parts) and the following apply.

AFI Application Family Identifier

AI Application Identification

AIDC Automatic Identification and Data Capture

CIN Company Identification Number

DI Data Identifier

DSFID Data Storage Format Identifier

ECI Extended Channel Interpretations

EPC Electronic Product Code

IAC Issuing Agency Code

IATA International Air Transport Association

IEP Inter-sector Electronic Purse

ISBT International Association of Blood Transfusion services

OID Object Identifier

PC Protocol Control (bits)

RFID Radio Frequency Identification

SN Serial Number

TEI Text Element Identifier

TID Tag identification

UII Unique Item Identifier

UML Unified Modeling Language

UPU Universal Postal Union

VIN Vehicle Identification Number

XPC Extended PC (bits)

5 Standards applied to data encoding for AIDC media

AIDC media in various forms are transported and/or stored, together with goods or items.

ISO/IEC 15434 was developed as a syntax for high capacity AIDC media and applied to many kinds of two-dimensional symbols.

ISO/IEC 15961 and ISO/IEC 15962 were developed for RFID air interface standards, as an encoding method only for RFID.

For the sake of simplicity, users want to use a single data standard for the various forms of AIDC media. (See Figure 1). However, because of the inherent characteristics of RFID and optical technologies, differences in data encoding arise, some of which will be described within this Technical Report.

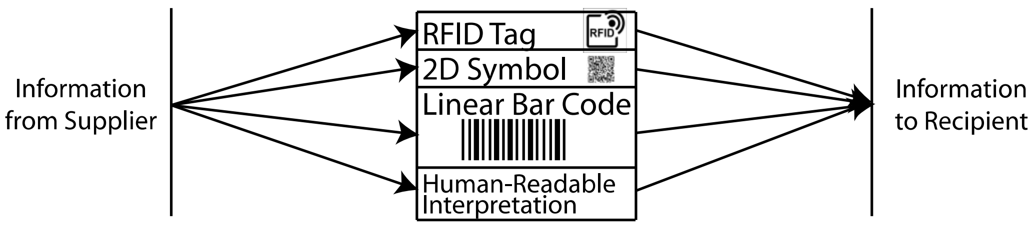


Figure 1 — Application user requirement

For example, bar codes are always scanned one at a time, but a large population of RFID tags can be inventoried nearly simultaneously. To support the RFID inventory operation, the Unique Item Identificaiton (UII) of the RFID tag is prefaced by “filtering” information (a numbering system identifier or an AFI) that has no correlation in bar code systems.

As a second example, for faster inventory operations, many RFID tag architectures transmit only the UII portion of their data during inventory, sending item attendant data only upon request. In contrast, a 2D symbol reader always obtains and transmits the full contents simultaneously (both UII and item attendant data).

Since the 1970s, linear bar code symbols have typically encoded application-specific “license plate” item information. The bar code symbol encodes an identifying primary key to a database entry that contains current information about the item. If the bar code identifier is not serialized (UPC symbols are an example), it identifies a class of item, such as a certain product of a certain size. If serialized, the “license plate” identifies a specific instance of an item; in open system applications, it is important that the identification system can guarantee that each “license plate” is uniquely distinct from all others.

Unique Item Identifiers (UIIs) can be contained in “unique identification-only” media such as a license-plate bar code symbol or an RFID tag containing only a UII. In the case of “unique identification-only,” a database or look-up to trading partner communications is required to establish additional information about the entity to which the UII is attached. Technologies such as two-dimensional symbols and data rich RF tags can contain this additional “item attendant data” within that medium.

A number of ISO/IEC specifications have been developed for encoding and decoding of linear bar code symbologies, such as ISO/IEC 15420 for EAN/UPC and ISO/IEC 15417 for Code 128, and for two-dimensional symbologies, such as ISO/IEC 15438 for PDF417 (see Bibliography for a complete list).

The remainder of this technical report describes currently available methods for encoding both UII and item attendant data in optical and RFID media. For all two-dimensional symbols, the data syntax specified in ISO/IEC 15434 (and summarized in Section 6 of this Technical Report) can be used. For most RFID data carriers, the UII is encoded separately (for efficient inventory operations), and the item attendant data should be encoded using ISO/IEC 15434 syntax. The RFID encoding options are summarized in Sections 7, 8, and 9 of this Technical Report, and additional RFID-specific guidance is provided in Sections 10, 11, and the Annexes A through D

6 ISO/IEC 15434 application for high capacity AIDC media

ISO/IEC 15434 is a transfer structure, syntax and coding of messages and data formats when using high capacity AIDC media between trading partners, specifically between suppliers and recipients and, where applicable, in support of carrier applications such as bills of lading and carrier sorting and tracking;

ISO/IEC 15434 includes encoded data:

- used in the shipping, receiving and inventory of transport units;
- contained within supporting documentation, in paper or electronic form, related to unit loads or transport packages;
- used in the sorting and tracking of transport units;
- used for the sorting and tracking of returnable transport items;
- used for the sorting and tracking of products and product packages.

To allow multiple data formats to be contained within a data stream, a two level structure of enveloping is employed. The outermost layer of the message is a Message Envelope that defines the beginning and end of the message. Within the Message Envelope are one or more Format Envelopes that contain the data (See Figure 2). Multiple formats in a single message should be employed only through trading partner agreements.

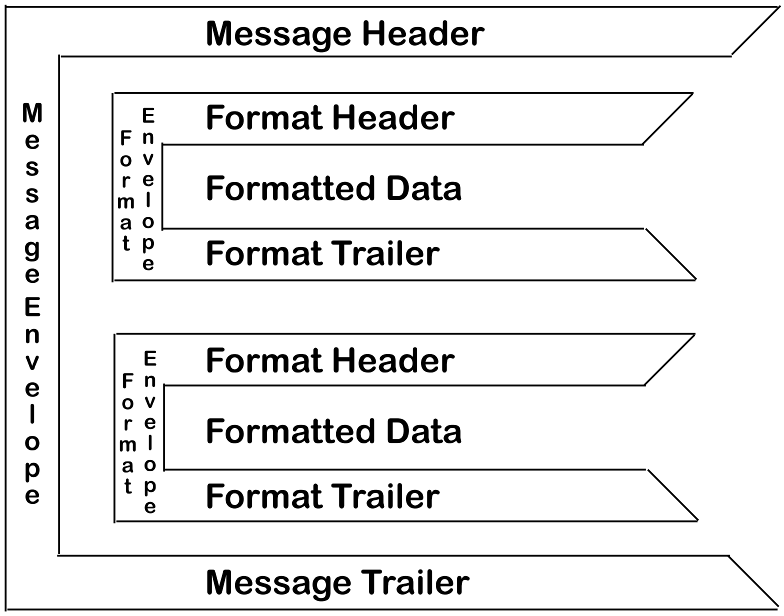


Figure 2 — Envelope structure of ISO/IEC 15434

6.1 Assigned formats in ISO/IEC 15434

Header data and format trailer for each format are defined in Table 1.

Table 1 — ISO/IEC 15434 header data and trailers

Format Indicator	Variable Header Data	Format Trailer	Format Description
00			Reserved for future use
01	$^{G}_{S}\mathit{vv}$	$^{R}_{S}$	Transportation
02			Complete EDI message / transaction
03	$\mathit{vvvrrr}^{F}_{S}{}^{G}_{S}{}^{U}_{S}$	$^{R}_{S}$	Structured data using ANSI ASC X12 Segments
04	$\mathit{vvvrrr}^{F}_{S}{}^{G}_{S}{}^{U}_{S}$	$^{R}_{S}$	Structured data using UN/EDIFACT Segments
05	$^{G}_{S}$	$^{R}_{S}$	Data using GS1 Application Identifiers
06	$^{G}_{S}$	$^{R}_{S}$	Data using ASC MH 10 Data Identifiers
07		$^{R}_{S}$	Free form text
08	*vvvvrrnn*		Structured data using CII Syntax Rules
09	$^{G}_{S}$ *ttt...t* $^{G}_{S}$ *ccc...c* $^{G}_{S}$ *nnn...n* $^{G}_{S}$	$^{R}_{S}$	Binary data (file type) (compression technique) (number of bytes)
10-11			Reserved for future use
12	$^{G}_{S}$	$^{R}_{S}$	Structured data following Text Element Identifier rules
13-99			Reserved for future use

Users should refer to ISO/IEC 15434 for the use of information objects as defined in the EDI standard directories, GS1 AI directory *(GS1 General Specification)* or ANSI DI directory *(ANS MH10.8.2).*

6.2 System data elements for compatibility across all AIDC media

As bar code technology began to proliferate in the 1980s, it became apparent that the need existed to encode more than simple product identity. Lot/batch and serial numbers, purchase order numbers, destination postal codes, country of origin and a unique license plate for the entity might all need to be encoded on a single label. Schemes in various industries evolved until the cross-industry exchange of product forced standardization of tags, or prefixes, to identify the information encoded in the bar code. This gave rise to the standardization of Data Identifiers (DIs) and Application Identifiers (AIs), referred to as the semantics of an AIDC data structure, managed by ASC MH10 (DIs) and GS1 (AIs) as defined in ISO/IEC 15418.

Over time, applications were developed for encoding the information on a shipping label into a single symbol, permitting the information to be read with a single scan. The ability to encode multiple data fields into a symbol created the requirement to know whether DIs or AIs were being read, where the various structures ended and others began, and when one would know that no more data followed. This gave rise to the standardization of data structures into messages, referred to as the syntax of an AIDC message, and was codified in ANS MH10.8.3 and later in ISO/IEC 15434.

ISO/IEC 24729-1, *Information technology — Radio frequency identification for item management — Part 1: RFID-enabled labels* provides a method for enoding the information resident in the RF tag into an optical symbol, thereby ensuring a backup source of data if the RFID tag should fail.

6.3 Data Carrier Identifiers for RFID and other AIDC media

Various applications need to identify the type of data carrier, and readers and interrogators are able to identify the means by which the data was entered: RFID, bar code, or key entry. They are able to preface the data with a data carrier identifier, following the rules of ISO/IEC 15424, *Information technology -- Automatic identification and data capture techniques -- Data Carrier Identifiers (including Symbology Identifiers).*

As an example, if an RF tag is unreadable, it may be possible to access a “back-up” technology, e.g. a linear bar code or two-dimensional symbol. If there is no “back-up” symbol or if it is unreadable, it may be necessary to key enter the data. Studies for key-entry of data have shown an error rate of approximately 1 in every 300 characters entered, compared to automated techniques with an error rate of 1 in every 1 000 000 characters entered, or better. If an RF tag or optically readable media fails, it is important to notify the supplier.

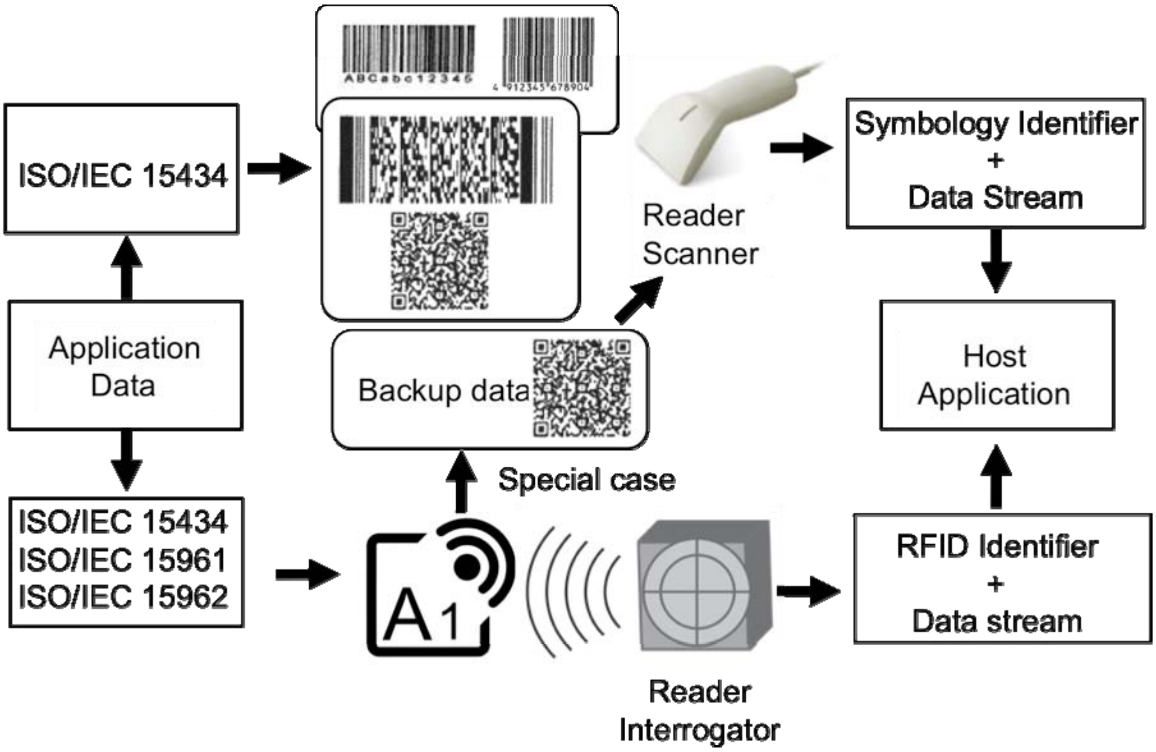


Figure 3 — Operation of data carrier identifiers

7 RFID encoding of UII

Figure 5 shows the memory layout of ISO/IEC 18000-63, Type C and ISO/IEC 18000-3, Mode 3 ASK RF tags. UIIs are encoded in Memory Bank '01' ($MB01_2$), as shown in Figure 6. The ISO/IEC 15459 series deals specifically with Unique Item Identifiers (UIIs), including the means to identify physical objects according to ISO TC 122's 1736x relevant documents, and EPC.

The AFI (Application Family Identifier) is encoded in $MB01_2$ in the event trading partners do not use EPC structures. The following subsections detail the steps involved.

7.1 Extant numbering systems for RFID

There are several existing systems to uniquely identify physical objects in an RFID context. These include:

- ISO/IEC 15459-1:2006, *Information technology — Unique identifiers — Part 1: Unique identifiers for transport units*
- ISO/IEC 15459-2:2006, *Information technology — Unique identifiers — Part 2: Registration procedures*
- ISO/IEC 15459-3:2006, *Information technology — Unique identifiers — Part 3: Common rules for unique identifiers*
- ISO/IEC 15459-4:2006, *Information technology — Unique identifiers — Part 4: Unique identifiers for individual items*
- ISO/IEC 15459-5:2007, *Information technology — Unique identifiers — Part 5: Unique identifier for returnable transport items (RTIs)*
- ISO/IEC 15459-6:2007, *Information technology — Unique identifiers — Part 6: Unique identifier for product groupings*
- ISO/IEC 15963:2004, *Information technology — Radio frequency identification for item management — Unique identification for RF tags*

— ISO/IEC 7816-5:2004, *Identification cards — Integrated circuit cards — Part 5: Registration of application providers,*
— ISO/IEC 7816-6:2004, *Identification cards — Integrated circuit cards — Part 6: Inter-industry data elements for interchange*
— *EPCglobal Tag Data Standards, Version 1.5*
— ITU X.668 | ISO/IEC 9834-9, *Information technology – Open Systems Interconnection – Procedures for the operation of OSI Registration Authorities: Registration of object identifier arcs for applications and services using tag-based identification*
— ITU X.660, *Information technology – Open Systems Interconnection – Procedures for the operation of OSI Registration Authorities: General procedures and top arcs of the International Object Identifier tree*

7.2 Tag type and UII data storage area

In the early RF tags, RFID memory consisted of a conventional memory structure, incorporating a system area and a user memory area, as shown in Figure 4. However, in ISO/IEC 18000-63 type C RFID, the structure changed, and the memory structure and the kinds of data that could be written in each memory area are defined in the ISO/IEC 18000-63 standard.

From the standpoint of storing UII data, the early RFID had only one user memory area and that is where UII data was stored. It is recommended that UII data should be the first element in these user area data elements.

In 18000-63 RFID, UII data is written in the UII are. If users intend to deal with two or more UIIs for one item, the second UII is considered as user data. Because the UII data writing area is dependent on both the memory type and the intention of the user, the system user should pay careful attention to this point.

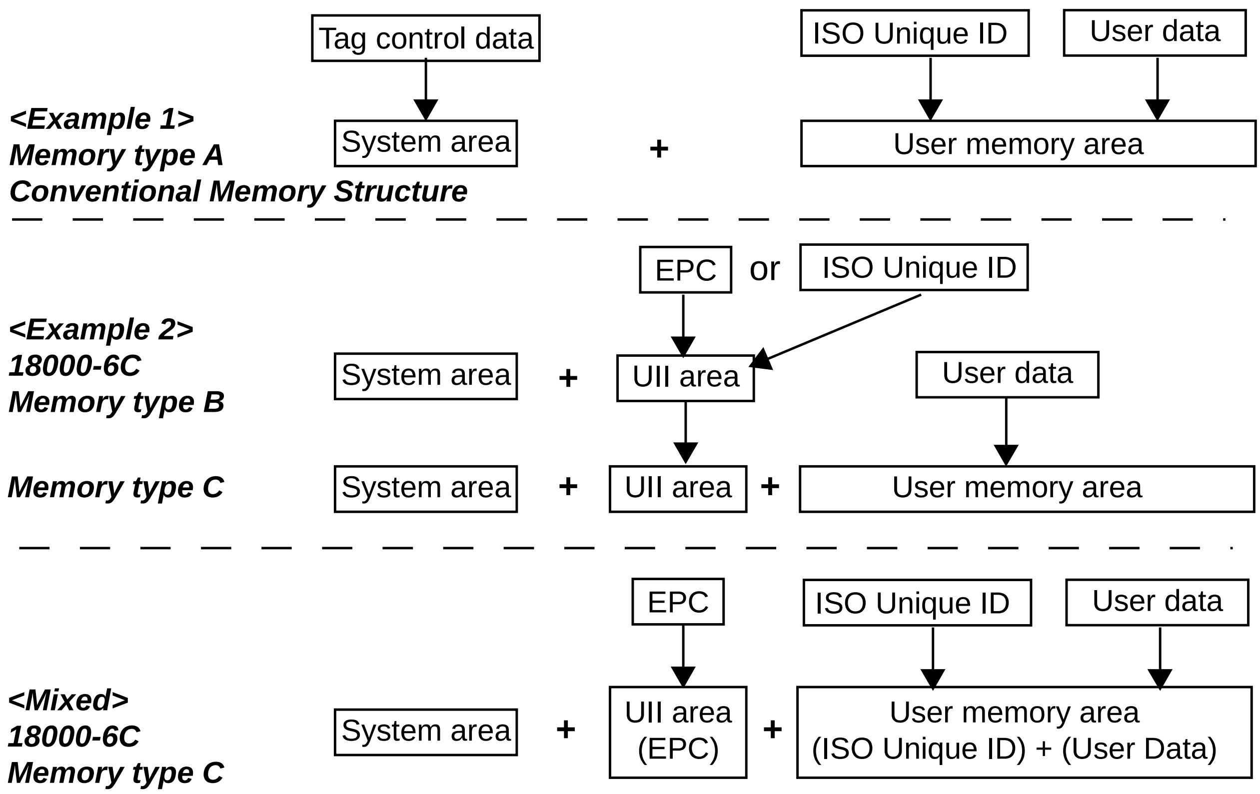


Figure 4 — RFID memory type and stored data to each memory area

7.3 ISO/IEC 18000-63, Type C and 18000-3m3 ASK and EPCglobal memory architecture

The first data element on a compliant tag for product and product packaging shall be the unique identification described in ISO/IEC 15459-4. The length and nature of this unique identification is defined in this data element. For an ISO/IEC 18000-63 Type C and ISO/IEC 18000-3 Mode 3 ASK compliant tag, the "unique identification" data element is segregated from any additional (User Data) by the memory architecture. The unique identification data element shall be stored in UII memory (Bank 01) with any additional data being stored in User memory (Bank 11). For the purposes of this standard a unique identity of products can be up to 35 alphanumeric characters in length, including the Data Identifier (an3+an..32). With trading partner agreement the combined length of the Data Identifier and data can be up to 50 alphanumeric characters in length. This architecture is shown in Figures 5 and 6 below.

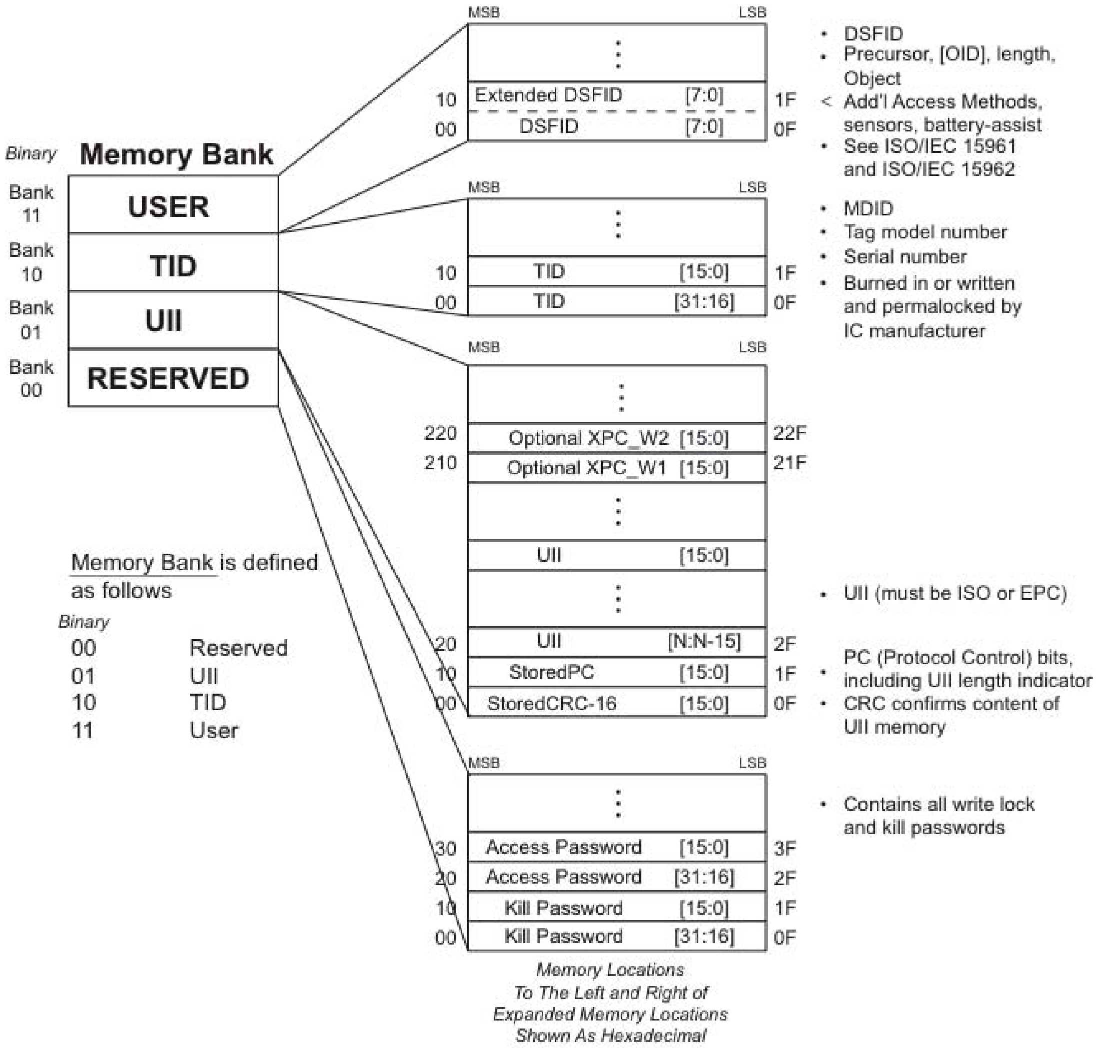
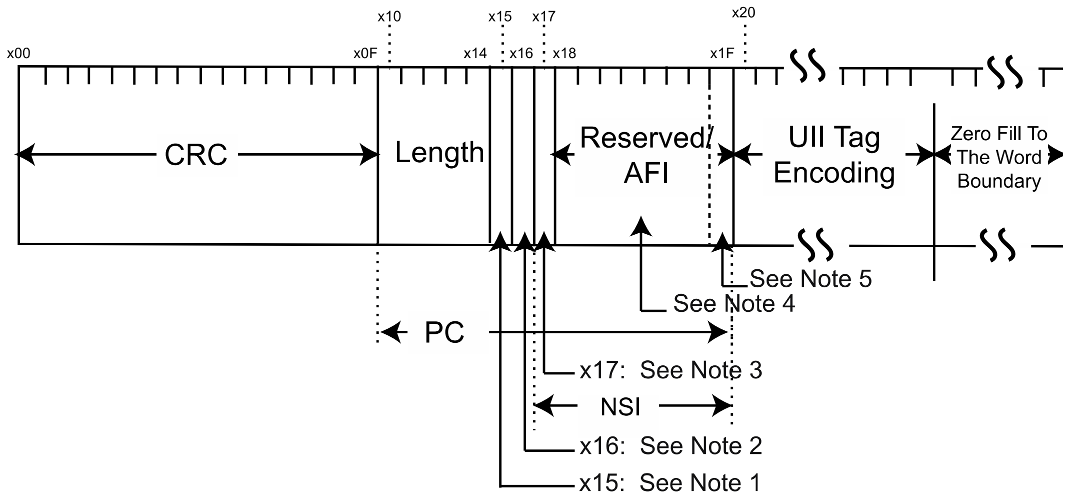


Figure 5 — 18000-63, Type C & 3m3 ASK, Memory structure



Note 1: User Memory (MB11) in Use Indicator (UMI)

Note 2: XPC Indicator

Note 3: “0=Binary / 1=AFI+ISO/IEC 15459”

Note 4: AFI for ISO / TDS-defined for EPC / 29161 defined for ISO binary

Note 5: Last bit of AFI for ISO / Haz Mat for EPC

Figure 6 — 18000-63, Type C & 3m3 ASK, Memory Bank (MB) “01_2”

7.4 Unique Identifier of a physical object (UII)

The UII can follow one of two formats designated by the state of Bit 0x17 of Memory Bank “01_2” (EPC=0/ISO=1). If Bit 0x17 is a “1” what follows at Bit 0x18 is an AFI (Application Family Identifier). The AFI formats are shared between JTC 1/SC 31(Automatic identification and data capture techniques) and JTC 1/SC 17 (Integrated Circuit Cards). The values assigned by JTC 1/SC 17 at the time of publication of this document are as shown in Table 2. The values under the control of JTC 1/SC 31 are shown in Table 3.

Table 2 — AFI values

AFI most significant nibble	AFI least significant nibble	Meaning	Examples / note
0	0	All families and subfamilies	No applicative pre-selection
X	0	All families and subfamilies	Wide applicative pre-selection
X	Y	Only the Yth subfamily of family X	
0	Y	Proprietary sub-family Y only	
1	0, Y	Transport	Mass transit, Bus, Airline, …
2	0, Y	Financial	IEP, Banking, Retail, …
3	0, Y	Identification	Access control, …
4	0, Y	Telecommunication	Public telephony, …
5	0, Y	Medical	
6	0, Y	Multimedia	Internet services, …
7	0, Y	Gaming	
8	0, Y	Data storage	Portable files, …
9	0, Y	Managed by JTC 1/SC 31	Managed by JTC 1/SC 31
A	0, Y	Managed by JTC 1/SC 31	Managed by JTC 1/SC 31
B	0, Y	UPU	Managed by JTC 1/SC 31
C	0, Y	IATA	Managed by JTC 1/SC 31
D	0, Y	RFU	Managed by JTC 1/SC 17
E	'0', Y = 1, Y = 2, Other Y values are RFU	Machine Readable Travel Documents (MRTDs)	Y=1 ePassport Y=2 eVisa
F	0, Y	RFU	Managed by JTC 1/SC 17

Table 3 — AFI values assigned by JTC 1/SC 31

Organization: function	AFI byte	Data Format binary	OID for UII	Data Format for additional memory	Root-OID for other data
Reserved for Future Use	90 – 9F				
ISO 17367: *Supply chain applications of RFID – Product tagging*	A1	6-bit ASCII	1 0 15459 4	6-bit ASCII	1 0 15418 1 ("1" suffix is to identify DIs, "0" for AIs)
ISO 17365: *Supply chain applications of RFID – Transport unit*	A2	6-bit ASCII	1 0 15459 1	6-bit ASCII	1 0 15418 1
ISO 17364: *Supply chain applications of RFID – Returnable transport item*	A3	6-bit ASCII	1 0 15459 5	6-bit ASCII	1 0 15418 1
ISO 17367: *Supply chain applications of RFID – Product tagging*, containing hazardous materials	A4	6-bit ASCII	1 0 15459 4	6-bit ASCII	1 0 15418 1
ISO 17366: *Supply chain applications of RFID – Product packaging*	A5	6-bit ASCII	1 0 15459 4	6-bit ASCII	1 0 15418 1
ISO 17366: *Supply chain applications of RFID – Product packaging*, containing hazardous materials	A6	6-bit ASCII	1 0 15459 4	6-bit ASCII	1 0 15418 1
ISO 17365: *Supply chain applications of RFID – Transport unit, containing hazardous materials*	A7	6-bit ASCII	1 0 15459 1	6-bit ASCII	1 0 15418 1
ISO 17364: *Supply chain applications of RFID – Returnable transport item*, containing hazardous materials	A8	6-bit ASCII	1 0 15459 5	6-bit ASCII	1 0 15418 1
ISO 17363: *Supply chain applications of RFID – Freight containers*	A9	6-bit ASCII	1 0 10891 0	6-bit ASCII	1 0 15418 1
ISO 17363: *Supply chain applications of RFID – Freight containers*, containing hazardous materials	AA	6-bit ASCII	1 0 10891 0	6-bit ASCII	1 0 15418 1
ISO/IEC 29174 for Mobile Item Identifier (MII) and additional data	AB	7	2 27 1	3	1 0 15434
Reserved for Future Use	AC - BF				
ISBT item level blood products	BB	30 encoded in DSFID = 0x3E	Not applicable	30 encoded in DSFID = 0x3E	Not applicable
IATA RP1740C baggage handling [separate UII memory]	C1	xxx01100	1 0 15961 12 1	xxx01100	1 0 15961 12
EDItEUR: library items	C2	xxx00110	1 0 15961 6 1		1 0 15961 6
Reserved for Future Use	C3 - CF				
Request by German Government agency for VIN to ISO 3779	TBD				
Request for Oil and Gas exploration sector, initiated by Norwegian Standards	TBD				

Note: At the time of publication this is the most current listing of AFI values. For an current version of the Data Constructs Register of which AFIs are a component please download the most current version from http://isotc.iso.org/livelink/livelink?func=ll&objId=8913563&objAction=browse&viewType=1

7.5 Data construct

AFIs A1 through AA note that the construction of the data shall be defined in the applicable standard, namely ISO 17367 through ISO 17363. Therein the AFI defines the data construct. The construct for GS1 data structures are not encoded using AFIs; they are encoded according to the rules of GS1 EPCglobal when Bit 0x17 of $MB01_2$ is set to a “0”. When Bit 0x17 (EPC=0/ISO=1) of $MB01_2$ is set to a “1” the semantics, and encoding complies with the rules of this clause.

7.5.1 Data semantics

When Bit 0x17 of $MB01_2$ (as shown in Figure 6) is set to a “0” the semantics will be as defined in the rules for GS1 EPCglobal. When Bit 17 of $MB01_2$ is set to a “1”, each data field should be prefaced by an ISO/IEC 15434 message format indicator.

7.5.2 Data syntax

Tags that encode identity only are considered to have no syntax. When Bit 0x17 of $MB01_2$ (as shown in Figure 6) is set to a “0” the syntax will be as defined in the rules for GS1 EPCglobal. When Bit 0x17 of $MB01_2$ is set to a “1” the syntax will the same as those for optically readable media.

7.5.2.1 Syntax for the Unique Item Identifier

The Unique Item Identifier (UII), contained in $MB01_2$, is interpreted as a single data element, similar to a single linear bar code. Therefore there is no message structure except for the Data Identifier and the data. The last character in this field is the $^{E}O_{T}$ character defined in Tables C.1 and D.1. The encoding of $^{E}O_{T}$ removes the necessity to either identify the length of the field or to zero- or blank-fill unused positions in a single variable length field.

The Unique Item Identifier (UII) shall be unique among all other items and shall be comprised of:

1. Unique item identifier compliant to ISO/IEC 15459 “25x”
   a. Data Identifier (DI) an…3
   b. Issuing Agency Code (IAC) an…3
   c. Company Identifying Number (CIN), and (length determined by Issuing Agency)
   d. Serial Number (SN)[1]

2. Vehicle Identification Number (an17) utilizing the DI “I”

When Bit 0x17 of $MB01_2$ (EPC=0/ISO=1) is set to a “1” and Bit 0x15 User Memory (No=0/Yes=1) is set to a “0”, the contents of $MB01_2$ will be read as a linear symbol.

7.6 Encoding of Memory Bank “01” Unique Item Identifier

Bit 0x17 is the switch between ISO formats and EPC formats. When Bit 0x17 is set to a “0”, the UII encoding is per the EPC Tag Data Standard, Version 1.5 or higher. When Bit 0x17 is set to a “1”, the UII encoding is per ISO/IEC 15459 preceded by an ISO/IEC 15961 Application Family Identifier (AFI). The specific AFIs defined for the ISO 1736x series of International Standards are shown in Table 4.

---

[1] The combined length of IAC, CIN, and SN shall be 50 characters or less. ISO/IEC 15459 states that “The composition of the serial number may include the product code, but in no case shall the UII exceed 50 characters plus the Data Identifier. For efficient use within various AIDC data carrier systems, it is recommended that the number of characters to be coded not exceed 20 characters, and number of characters should be kept as short as possible regardless of the permissible maximum of 50 characters.” The ISO 1736x series of standards state, “A unique identifier can be up to 35 alphanumeric characters in length, excluding the Data Identifier (an3+an..35). With the mutual agreement of the trading partners this length can be extended to 50 characters (an3+an..50).”

Table 4 — 1736x Application Family Identifiers (AFIs)

AFI	Assigned Organisation or Function
0xA1	ISO 17367 product tagging
0xA2	ISO 17365 transport unit
0xA3	ISO 17364 returnable transport unit
0xA4	ISO 17367 product tagging, but for hazardous materials
0xA5	ISO 17366 product packaging
0xA6	ISO 17366 product packaging, but for hazardous materials
0xA7	ISO 17365 transport unit, but containing hazardous materials
0xA8	ISO 17364 returnable transport unit, but containing hazardous materials
0xA9	ISO 17363 freight containers
0xAA	ISO 17363 freight containers, but containing hazardous materials

A linear bar code symbol encoding the data providing unique item identification is comprised of the Data Identifier (DI), Issuing Agency Code (IAC), Company Identification (CIN), and Serial Number (SN). Such a unique item identification linear bar code would be represented in Code 128 as shown in Figure 7.

- DI = 25S
- IAC = UN (DUNS)
- CIN = 043325711
- SN = MH8031200000000001

Figure 7 — Code 128 encoding "25SUN043325711MH8031200000000001"

Adding the AFI to the structure for RFID purposes we have

- AFI = 0xA1
- DI = 25S
- IAC = UN (DUNS)
- CIN = 043325711
- SN = MH8031200000000001

Looking at a completed data structure using the encoding defined above and using DUNS as the Issuing Agency Code (IAC), we find that $MB01_2$, when encoding a Product, this data structure is 25SUN043325711MH8031200000000001 and is represented in MB01 as shown in Table 5.

Table 5 — MB01 structure of AFI and UII (DUNS) using six-bit encoding

AFI = 0xA1			2	5	S	U	N	0	4	3	3	2	5	7	1
1010 0001			110010	110101	010011	010101	001110	110000	110100	110011	110011	110010	110101	110111	110001
1	M	H	8	0	3	1	2	0	0	0	0	0	0	0	0
110001	001101	001000	111000	110000	110011	110001	110010	110000	110000	110000	110000	110000	110000	110000	110000
0	0	1													
110000	110000	110001													

Alternately, looking at a completed data structure using the encoding defined above, using ODETTE as the Issuing Agency Code (IAC), we find that $MB01_2$ when encoding a Product having an:

- AFI = 0xA1
- DI = 25S
- IAC = OD (ODETTE)
- CIN = CIN1
- SN = 0000000RTIA1B2C3DOSN12345 (This example shows the SN composed of Object Type and Object Serial Number)

. . . we have an $MB01_2$ structure as shown in Table 6.

Table 6 — MB01 structure of AFI and UII (ODETTE) using six-bit encoding

AFI = 0xA1			2	5	S	O	D	C	I	N	1	0	0	0	0
1010 0001			110010	110101	010011	001111	000100	000011	001001	001110	110001	110000	110000	110000	110000
0	0	0	R	T	I	A	1	B	2	C	3	D	O	S	N
110000	110000	110000	010010	010100	001001	000001	110001	000010	110010	000011	110011	000100	001111	010011	001110
1	2	3	4	5											
110001	110010	110011	110100	110101											

In both cases, once the AFI is stripped from the message, the output of the RFID reader is identical to that of the linear bar code.

7.6.1 Conclusion

Using the ISO/IEC 15434 Direct Encoding, MB01 of an ISO/IEC 18000-63, Type C or an ISO/IEC 18000-3, Mode 3 ASK RF tag will provide the same output as optically readable media. Using the Harmonized Six-Bit approach, the encoding of MB11 of an ISO/IEC 18000-63, Type C or an ISO/IEC 18000-3, Mode 3 ASK RF tag will provide the same output as Figures 8 and 9, below. This method also has the benefit of simplifying the data encoding process. When encoded in a 2D symbol, the output would be identical:

[)><RS>06<GS>25SUN043325711MH8031200000000001<GS>1T110780<GS>Q21<GS>4LUS<RS><EOT>

Figure 8 — QR Code encoding the contents of MB01 and MB11

[)><RS>06<GS>25SUN043325711MH8031200000000001<GS>1T110780<GS>Q21<GS>4LUS<RS><EOT>

Figure 9 — Data Matrix encoding the contents of MB01 and MB11

[)><RS>06<GS>25SUN043325711MH8031200000000001<GS>1T110780<GS>Q21<GS>4LUS<RS><EOT>

For 18000-63, Type C and 18000-3, Mode 3 ASK air interfaces an interrogator will simply read the number of words specified in 0x10 to 0x14 of $MB01_2$, read those bytes, and then pass the payload for post-read processing, which might be done in a host processor or a module of the interrogator. From that read the Unique Item Identifier (UII) will be extracted.

8 RFID encoding of user data

The Access-Method, (shown in Table 7) as defined by the application, is the most significant determinant of how data is encoded on the RFID tag. The value of the Access-Method should be stored on the RFID tag, or may be defined by the air interface services, if this can be done unambiguously. The Access-Method is defined as an integer value in the application command and encoded as a compound bit value in the DSFID and the SFF (Special Features Flag) byte on the RFID tag

Table 7 — ISO/IEC 15962 Table 7

15961 integer code	15962 DSFID bit code	15962 SFF bit code	Name	Description
0	00	00	**No-Directory**	This structure supports the contiguous abutting of all the Data-Sets
1	01	00	**Directory**	The data is encoded exactly as for **No-Directory** but the RFID tag supports an additional directory, which is first read to point to the address of the relevant object identifier.
2	10	00	**Packed-Objects**	This is an integrated compaction and encoding scheme that formats data in an indexed structure as defined by the Application administrator (see ISO/IEC 15961-2)
3	11	00	**Tag-Data-Profile**	This is an integrated compaction and encoding scheme for a fixed set of data elements, each of a defined length

8.1 No directory

The **Access-Method** = **No-Directory** is designed to achieve a combination of flexibility and efficiency for the bytes that are encoded on the RFID tag. In particular:

- Data Objects are compacted efficiently using a defined set of compaction techniques that reduce the encoding of data objects on the RFID tag across the air interface;
- Data formatting minimises the encoding of the Object-Identifiers on the RFID tag and on the air interface, but still provides complete flexibility for identifying specific data.

8.2 Directory

The **Access-Method** = **Directory** has a two part structure in the Logical Memory:

- The lower addressed blocks are identical to the No-Directory structure;
- The higher blocks contain the directory.

8.3 Packed Object

The **Packed-Objects** encoding scheme was developed after the first edition of ISO/IEC 15962 was published. It uses a rules-based table, defined by the application administrator and registered under the rules of ISO/IEC 15961-2. This encoding scheme specifies common compaction schemes which are significantly more efficient than those defined for the **No-Directory** structure, and a compaction scheme can be specified for each **Relative-OID** value in the table. In addition, the Packed-Objects encoding scheme may specify the use of the same compaction schemes as of the **No-Directory Access-Method**. This enables a simpler implementation, but still with encoding efficiencies over the basic **No-Directory Access-Method**.

8.4 Tag Data Profile

The **Access-Method** = **Tag-Data-Profile** is designed to support applications that are able to define all the encoded data as mandatory and of a fixed or maximum length. It is possible to apply encoding rules that achieve an efficient encoding of the bytes on the RFID tag. In particular:

— The Tag-Data-Profile table, if accessed by the interrogator, provides a rapid access to any data on the RFID tag.

— Data Objects are compacted efficiently using a defined set of compaction techniques that reduce the encoding of data objects on the RFID tag across the air interface.

— The Precursor and other syntax on the No-Directory Access-Method are retained to enable interrogators with no access to the Tag-Data-Profile to decode the data.

The encoding consists of some meta-data that uniquely identifies the **Tag-Data-Profile** plus encoded data sets that are the same as if encoded using **the No-Directory Access-Method**, except that all the data has a predetermined length and compaction.

9 RFID ISO/IEC 15434 direct encoding of user data

There is one combination of the ISO/IEC 15962 Access Method and Data Format used for the direct encoding of an ISO/IEC 15434 message into user memory (as in the case of 18000-63, Type C). Annex C and Annex D detail the methods of encodation and formatting. Annex C gives examples of all supported ISO/IEC 15434 data formats whereas Annex D explains specific implementations of Data Identifiers for use with ISO TC 122 documents (ISO 17364, ISO 17365, ISO 17366, and ISO 17367).

10 Storing data in various types of RF tags

For RFID transponders that do not have memory bank separation, there are four ways to store user data in the memory of the transponder.

— Employ ISO/IEC 15434. The first information object should be the UII. UII is interpreted as a linear bar code symbol following the rules of ISO/IEC 15459. Utilize ISO 1736x to specify encoding. Differences in encoding are an attribute of the Application Family Identifier (AFI). The revision of 1736x specifies 6-bit ASCII encoding as shown in Annex D.
— Employ ISO/IEC 15961 and ISO/IEC 15962 structure. With root OID the reference should be ISO/IEC 15434. The user data can be constructed based on the rules of ISO/IEC 15434. And it is recommended that the first information object should be the UII.
— Employ ISO/IEC 15961 and ISO/IEC 15962 structure. With root OID the reference should be ISO/IEC 15418. The user data can be constructed using ANSI DIs or GS1 AIs. And it is recommended that the first information object should be the UII.
— Employ ISO/IEC 15961 and ISO/IEC 15962 structure. The user data can be constructed using proprietary defined information objects (with Object IDs & Values). And it is recommended that the first information object should be the UII.

For RFID transponders that have memory bank separation (e.g., ISO/IEC 18000 part 6 type C and ISO/IEC 18000 part 3 mode 3: see Figure 5), there are also four ways to store user data in the memory of the RFID transponder.

— Employ ISO/IEC 15434. The UII should be stored in “MB01”. Utilize ISO 1736x to specify encoding. Differences in encoding are an attribute of the Application Family Identifier (AFI). The revision of 1736x specifies 6-bit ASCII encoding as shown in Annex D. User memory “MB11” is interpreted as a two-dimensional symbol using the ISO/IEC 1534 syntax

- Employ ISO/IEC 15961 and ISO/IEC 15962 structure. With root OID the reference should be ISO/IEC 15434. The user data can be constructed based on the rules of ISO/IEC 15434. And the UII should be stored in the UII memory bank.
- Employ ISO/IEC 15961 and ISO/IEC 15962 structure. With root OID the reference should be ISO/IEC 15418. The user data can be constructed using ANSI DIs or GS1 AIs. And the UII should be stored in the UII memory bank.
- Employ ISO/IEC 15961 and ISO/IEC 15962 structure. The user data can be constructed using proprietary defined information objects (with Object IDs & Values). And UII should be stored in UII memory bank.

11 Methods to store UII data in RFID memory and other AIDC media

As defined in the ISO/IEC 5459 series, the UII is constructed by unique issuing agency code, unique code allocated by IAC to unique item identifier issuer and unique code allocated to the entity by unique item identifier issuer.

For individual items, the code is basically constructed from the product code and the product serial number.

A unique item identifier is a unique sequential serial number, but it is not the only way to serialize a part. In some cases this part is constructed of meaningful characters and a unique serial number. In more complicated cases, it has highly integrated structures, requiring the creation of a long UII and requires a large memory size. However, UII issuers can construct the UII in conjunction with a unique serial number and numerous kinds of necessary application data. People who have to deal with items may easily recognize the items characteristics by using this kind of UII, and it is very helpful for workers.

Users who are considering UII should pay attention to the above case (also, see Figure 10). They have an opportunity to include application data in UII for better operation.

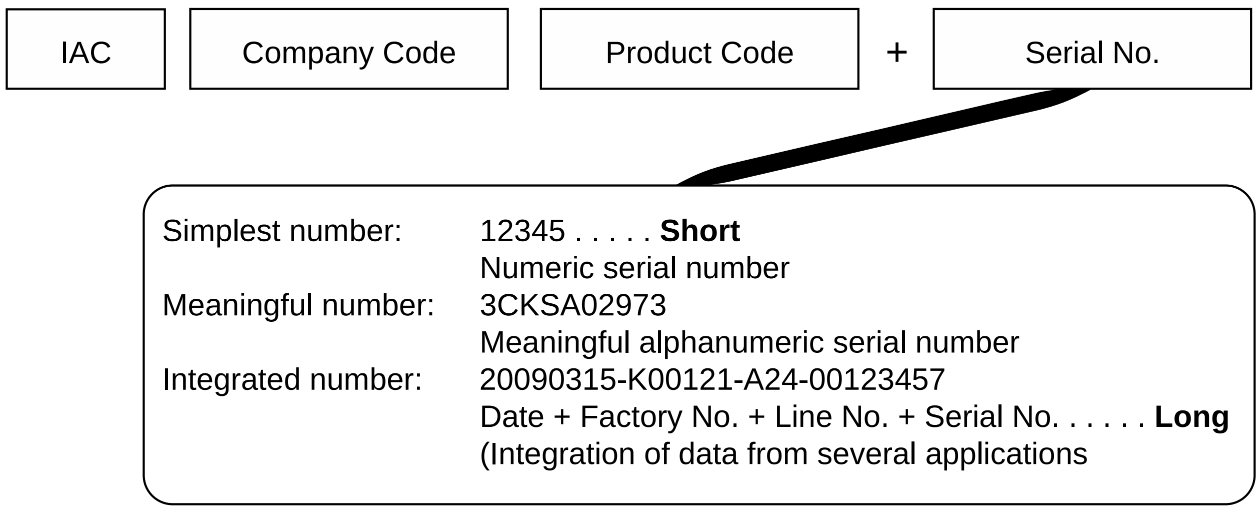
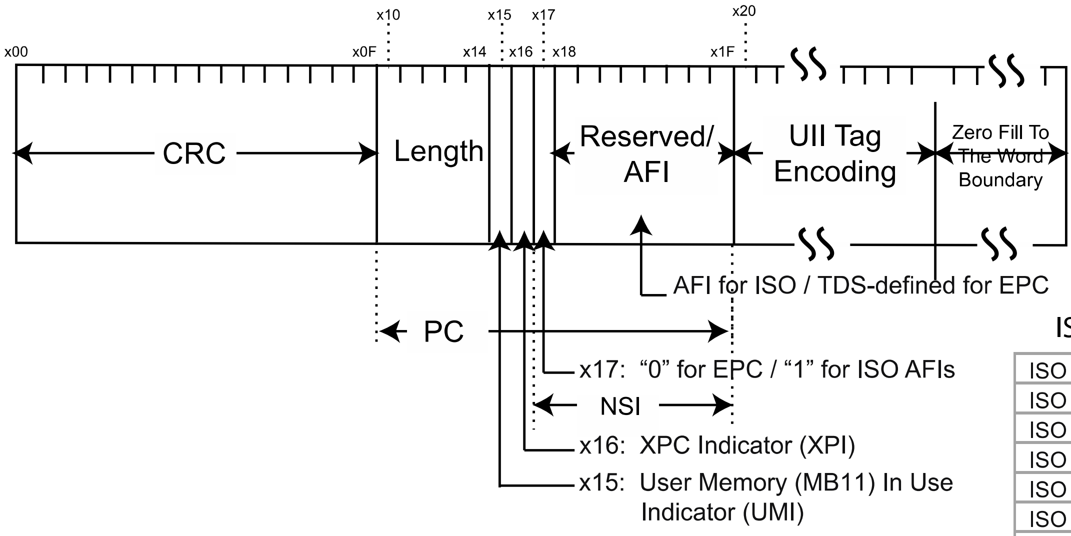


Figure 10 — UII examples

Annex A
(informative)

18000-63, Type C and 18000-3m3 ASK Memory Bank “01”



ISO 1736x AFIs

ISO 17367	A1
ISO 17365	A2
ISO 17364	A3
ISO 17367 HzMt	A4
ISO 17366	A5
ISO 17366 HzMt	A6
ISO 17365 HzMt	A7
ISO 17364 HzMt	A8
ISO 17363	A9
ISO 17363 HzMt	AA

Figure A.1 — UII bank data example

Annex B
(informative)

The Concept of Unique Identity

I have six honest serving men
They taught me all I knew
*There names are **What**, and **Where** and **When**;*
*and **Why** and **How** and **Who**.*

- Rudyard Kipling (1865-1936)

With apologies to the celebrated English poet and author, the supply chain submits that a seventh honest man is missing, whose name is ***Which***. And, as in Macbeth, there are three "***Whiches***": ***Which*** one, ***Which*** group, and ***Which*** consignment. For these seven "questions" underpin the very heart of the supply chain, traceability, tracking, and chain of custody.

The Internet of Things embodies the means where one thing can communicate with another thing via the Internet. Here a thing is an "entity" as shown in Figure 3 below where an object communicates to another object or a person, legal or real. For that "entity" to communicate it must possess an identifier of "which" it is.

There is no shortage of claimants to provide that identifier. Each is understandable due to its origins and the perspective from which it comes. The Internet is a network and groups such as the International Telecommunications Union (ITU) and the Internet Engineering Task Force (IETF) view this identifier as a mechanism to facilitate network routing. ITU X.668 | ISO 9834-9 and ITU X.660 attempt to fill this need from a network perspective. Ultimately, it is accepted that the identification of an entity must resolve to an IP address for contacting it, whether its domain name "hangs" from an OID root using an OID resolver, or from a more general DNS node (which may end up as the same thing).

However, not everything is viewed from the perspective of the network, nor necessarily should it so be viewed, for the network is a transport mechanism and the entities themselves have historic identifiers, which have their genesis from supply chain applications and identification. In 2001 David L. Brock authored the original paper on this topic: *The Electronic Product Code (EPC) – A Naming Scheme for Physical Objects.* Here, the primacy of established coding structures in the supply chain was acknowledged and a technique developed for translating an established product code into a unique digital identification and then to Uniform Resource Name (URN) for network usage.

Some have proposed a "master resolver" that would accept all existing and future schemes and to then provide commonality among the various identifiers. To provide the key to this master resolver one must identify the identifier. This key would either require the modification of all existing identifiers or add to that which already exists. What is certain is that to burden established schemes would require a compelling case for change. Alternately, and most likely the path of least resistance would be to establish an existing identifier scheme and to then establish a maintenance structure that would provide for the broadest set of applications.

At the present time there is no acceptable solution to resolving these disparate identities. In the not too distant future, I may have a telephone conversation with colleagues in Daejon or Moscow, where each of us will speak in our native language and the other will be able to understand the other without a need to be competent in the other's language. Such a network will provide this universal translation by being "content aware" and so too will such a master resolver be able to be content aware of the unique identity scheme without burdening the identity scheme with additional overhead. The solution to the master resolver is a network that will be content aware.

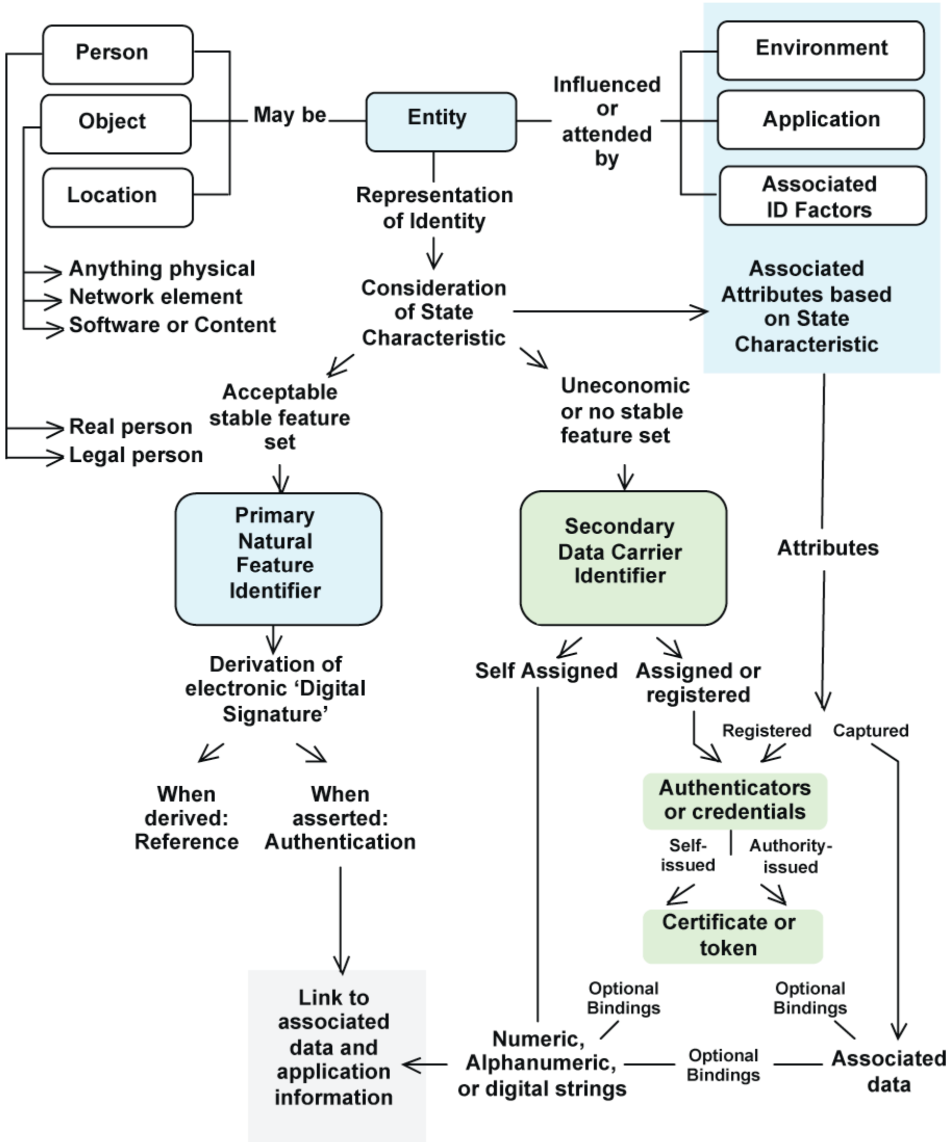


Figure B.1 — Ontology of identity

Annex C
(informative)

ISO/IEC 15434 Direct Encoding and Transmission

C.1 General

There is one Access Method and Format used in conjunction with encoding ISO/IEC 15434 data onto the user memory (as in the case of 18000-63, Type C). This annex details methods of encodation and formatting. Annex C gives examples of all supported ISO/IEC 15434 data formats whereas Annex D explains specific implementations of Data Identifiers for use with ISO TC 122 documents (ISO 17364, 17365, 17366, and 17367).

The DSFID Access Method 0 coupled with ISO/IEC 15961 Data Format 3 gives ISO/IEC 15962 ("15962") the capability for direct encodation of all ISO/IEC 15434 ("15434") formats (e.g., TEIs, DIs, binary). Special 6-bit tables (See Table C.1 and Table D.1) are defined that are different from the 6-bit table used in the rest of ISO/IEC 15962, as it has specific mappings to particular characters used in 15434. The Precursor byte allows the system to specify which 15434 format and character set table the message requires.

Direct 15434 encoding via ISO/ICE 15962 processes an ISO/IEC 15434 message in the same way as bar code applications. By default, the RFID interrogator transmits the data exactly as specified in ISO/IEC 15434. Therefore this method allows ISO/IEC 15434 data to be encoded in RFID tags and used in existing bar code applications without modification.

Users of this annex should be aware that it describes a subset of the full ISO/IEC 15962 encoding and transmission process. When additional features need to be utilized (e.g. when there are more than 16 383 bytes in a message) and or there are questions regarding the encode/decode rules, ISO/IEC 15962 and particularly Annex D therein, should be consulted.

C.2 DSFID

Data encoding starts with the DSFID (Data Storage Format Identifier) that encodes the access method and Data-Format. When using direct 15434 encoding, the DSFID is 0x03.

C.3 Precursor byte

Encoding continues with the Precursor. The Precursor encodes the extension bit in the most significant position, the compaction type (next three bits) and the 15434 Format envelope (four least significant bits). For example, the Precursor byte for most Data Identifier (DI) applications is "0 100 0110" or 0x46 (i.e. extension bit is always "0", a compaction type 4 which indicates use of the special 6-bit table defined in this annex and a 15434 format envelope "06"). Annex C.6 describes other Precursor values.

NOTE: Some DI messages (e.g. DIxx for AIs) may require the 7-bit character set.

C.4 Data byte-count indicator

Some air interface protocols allow for optimization in noisy environments by varying the number of bytes sent in each transmission. Therefore it is useful to know at the beginning the number of bytes in tag memory that contain data. For many ISO/IEC 15434 data encoding applications (e.g. DIs, TEIs), the number of bytes needed to encode the data will be a number less than 127 and therefore handled in one byte. For larger messages, two bytes are used where the first byte begins with "1" and the second byte begins with "0" as in Annex D.2 of ISO/IEC 15962. The number of bytes is encoded in the 14 remaining bits (e.g. 200 bytes is encoded as "1**0000001** 0**1001000**") (only the rightmost 7 bits are used in this binary counting method, and are bold in the previous example).

For example, if a message contains 51 6-bit characters, it will be encoded in 39 bytes (i.e. the last bit of the last data character is in the 39th byte and in this case there are six un-encoded bits which require padding). Therefore the data byte-count indicator is 0x27.

C.5 Encoding and decoding

C.5.1 Encode process

1. Starting with a valid ISO/IEC 15434 message, determine “nn” (where “nn” is the ISO/IEC 15434 format). from the initial bytes “[ ) > RS nn” Determine the Precursor byte from the “nn” and the choice of code set best suited for the 15434 message (e.g. the ISO/IEC 15434 TEI format “nn” is 12 and the data can be encoded with the special 6-bit code table so the precursor byte is “0 100 1100” or 0x4C”). Strip the 15434 header (usually “[ ) > RS nn GS”) from the front and trailer (usually “RS EOT”) from the end.

2. Convert every data character into its code value. For the 6-bit code, use Table C.1.

Table C.1 — 6-bit encoding for ISO/IEC 15962

Space	100000	0	110000	@	000000	P	010000
<EOT>	100001	1	110001	A	000001	Q	010001
Reserved	100010	2	110010	B	000010	R	010010
<FS>	100011	3	110011	C	000011	S	010011
<US>	100100	4	110100	D	000100	T	010100
Reserved	100101	5	110101	E	000101	U	010101
Reserved	100110	6	110110	F	000110	V	010110
'	100111	7	110111	G	000111	W	010111
(	101000	8	111000	H	001000	X	011000
)	101001	9	111001	I	001001	Y	011001
*	101010	:	111010	J	001010	Z	011010
+	101011	;	111011	K	001011	[	011011
,	101100	<	111100	L	001100	\	011100
-	101101	=	111101	M	001101	]	011101
.	101110	>	111110	N	001110	<GS>	011110
/	101111	?	111111	O	001111	<RS>	011111

NOTE 1 Table C.1 is six-bit encoding created through the removal of the two high-order bits from the ISO 646 8-bit ASCII character set except for the five characters <EOT> <FS> <US> <GS> and <RS> which are included to minimize the bit count when using the 15434 envelope.

NOTE 2 Use of these characters should be avoided as they are not supported by some application standards.

3. When encoding multiple Format Envelopes that use the same Format Indicator as was encoded by the Precursor (e.g. to represent a message containing several “records” from the same data format in order to describe the subassemblies of a complex part) reduce each internal 15434 sequence “RS nn GS” indicating a new “record” to a single <RS> character (encoded as “011111” from Table C.1).

When encoding multiple Format Envelopes that use a different Format Indicator as was encoded by the Precursor, the complete internal 15434 sequence "RS nn GS", with "nn" indicating the new format shall be encoded.

4. Encode an <EOT> character after the last encoded data character.

5. For ISO/IEC 15434 messages based on Table C.1, lay out the 6-bit characters as bits and then group them into 8-bit bytes. For 15434 Formats that call out other compaction schemes lay out their n-bit characters as bits and then group them into 8-bit bytes.

6. For ISO/IEC 15434 messages based on Table C.1, add the first 2 or 4 bits of an <EOT> character (i.e. “10” or “1000”) or the entire <EOT> character (i.e. “100001” from the 6-bit character set) to fill un-encoded bits in the last byte, if any.

7. Determine the byte number that contains the last bit of the last specified data character, convert the decimal count into binary and encode explicitly as the data byte-count indicator.

8. Encode the DSFID, Precursor, data byte-count indicator and data into memory.

NOTE: If the 15434 Format is greater than "14", see ISO/IEC15962 for encoding rules.

Note: Because only one 15434 message is allowed to be encoded in a single RFID data carrier, there is no need to encode a zero byte as a terminator after the last data byte. If the 15434 message is to be locked, no additional encoding is required.

C.5.2 Decode process

1. Examine the DSFID byte and verify that it is equivalent to 0x03.

2. Take the next 8-bits (Precursor Byte) and verify the first bit is “0” then take the next three bits and compare to Table 9 to determine the code set and convert the last four bits into two decimal digits for transmission as the 15434 Format.

   NOTE: If the 15434 Format is greater than "14", see ISO/IEC 15962 for decoding rules.

3. Process the next 8 bits and convert to decimal to determine the number of bytes containing data.

4. Group the remaining bits into character bit-sets from the compaction type code value (Table C.3) indicated in the Precursor byte and continue until the assigned number of bytes has been parsed (e.g. if the code bits are “100” then analyze the data bits in groups of six).

5. Assign data characters according to the code table indicated and delete all padding bits from the end.

6. Delete the encoded <EOT> from the end.

7. For any encoded <RS> character that is not immediately followed by 2 digits and a <GS> character, expand the <RS> to “RS nn GS”, where “nn” is the Format Indicator encoded in the Precursor

8. Add the ISO/IEC 15434 message and format header (usually “[ ) > RS nn GS”) to the beginning of the transmission (e.g. “nn” = 12 for TEIs) and the 15434 format and message trailer (usually “RS EOT”) at the end. There are different format headers and format trailers for 15434 messages and implementations of this decode process shall refer to the latest edition of ISO/IEC 15434 to ensure that all formats are correctly supported.

9. Transmit the entire 15434 compliant message. Optionally, the receiver may wrap the 15434 message in an OID format as a single data object. When using this option, the complete OID of the message is {1 0 15434 nn}, where “nn” is the two-digit representation of the 15434 Format Indicator encoded in the Precursor.

C.6 Encoding and decoding example using Data Identifiers

C.6.1 Translation and encoding procedure from 15434 data to Access-Method 0 Format 3

- To prepare a typical Data Identifier (DI) input message in a ISO/IEC 15434 format for encoding using ISO/IEC 15962 Access Method 0 and Format 3, the following steps are performed.
- Verify that the input message is a valid 15434 message
- The DSFID indicating Access Method 0 and Format 3 is encoded

- The Precursor byte with the chosen 15434 Format envelope and compaction type is encoded (e.g. review the data and choose the character set with the smallest number of bits per character that can encode all the data)
- The leading message envelope characters “**[ ) > RS 06 GS**” and the trailing “**RS EOT**” are discarded.
- The data is encoded into 6-bit codewords from Table C.1
- Add an <EOT> character.
- Add part or all of an <EOT> to act as pad bits to fill the last data byte, if necessary.
- Encode the DSFID, Precursor, data byte-count indicator, data, <EOT> and pad bits (if any) into memory.

C.6.2 Decoding and translation procedure from Access Method 0 Format 3 to 15434 data

The system will see this information as 15434 6-bit DI data by reading the DSFID and Precursor byte.

- The system strips off any pad bits from the end of the last byte,
- The system strips of the encoded 6-bit <EOT> character.
- The encoded bytes are parsed into 6-bit codes and then into data according to Table C.1.
- The system adds “[ ) > RS 06 GS” to the beginning of the transmission and “RS EOT” at the end
- The system transmits the entire 15434 compliant message

Optionally, the receiver may wrap the entire 15434 message in an OID format as a single data object.

C.6.3 Data encode and transmission example

Starting data:

[)> $^{R}{}_{S}$ 06 $^{G}{}_{S}$ 25SUN043325711MH8031200000000001 $^{G}{}_{S}$ 1T110780 $^{G}{}_{S}$ Q21 $^{G}{}_{S}$ 4LUS $^{R}{}_{S}$ $^{E}{}_{OT}$

The data on the tag from the above message is as follows (with DIs in **bold** font):

*25S**UN043325711MH8031200000000001** $^{G}{}_{S}$ 1T**110780** $^{G}{}_{S}$ Q**21** $^{G}{}_{S}$ 4L**US** $^{E}{}_{OT}$*

Where:

- UII = **25S**UN043325711MH8031200000000001
- LOT = **1T**110780
- QTY = **Q**21
- CoO = **4L**US

Data to bit conversion:

There are 51 6-bit characters (50 plus <EOT>) which translates to 39 data-bytes. There is a need to fill six trailing bits for byte alignment so in this case an entire <EOT> character is encoded. See Table C.2.

Table C.2 — Six-bit values assigned by system

DSFID = 0x03	**Precursor = 0x46**	**Data byte-count = 0x27**	**2**	**5**	**S**	**U**	**N**	**0**	**4**	**3**	**3**
00000011	01000110	00100111	110010	110101	010011	010101	001110	110000	110100	110011	110011
2	**5**	**7**	**1**	**1**	**M**	**H**	**8**	**0**	**3**	**1**	**2**
110010	110101	110111	110001	110001	001101	001000	111000	110000	110011	110001	110010
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**<GS>**
110000	110000	110000	110000	110000	110000	110000	110000	110000	110000	110001	011110
1	**T**	**1**	**1**	**0**	**7**	**8**	**0**	**<GS>**	**Q**	**2**	**1**
110001	010100	110001	110001	110000	110111	111000	110000	011110	010001	110010	110001
<GS>	**4**	**L**	**U**	**S**	**<EOT>**	**pad**					
011110	110100	001100	010101	010011	100001	100001					

C.6.3.1 Complete contents of tag memory

Using the Access Method 0 Format 3 encoding, including a DSFID, 15434 Precursor byte,39 bytes of data (compressing 51 6-bit characters including the <EOT>) and six pad bits, the final tag encodation in hexadecimal is as follows.

03 46 27 CB 54 D5 3B 0D 33 CF 2D 77 C7 13 48 E3 0C F1 CB 0C 30 C3 0C 30 C3 0C 31 7B 15 31 C7 0D F8 C1 E4 72 C5 ED 0C 55 38 61

C.6.3.2 Transmitted data

The header characters and the “RS EOT” are reinserted into the message. The following data string is transmitted from the reader.

[)> ${}^{R}{}_{S}$ 06 ${}^{G}{}_{S}$ 25SUN043325711MH8031200000000001 ${}^{G}{}_{S}$ 1T110780 ${}^{G}{}_{S}$ Q21 ${}^{G}{}_{S}$ 4LUS ${}^{R}{}_{S}$ ${}^{E}{}_{OT}$

C.7 Additional code values and other precursor features

This section contains basic information relating to encoding and compaction methods supported in this Technical Report.

C.7.1 Expansion bit

The most significant Precursor bit indicates there is one precursor byte (if set to 0) or whether there is an additional precursor byte (if set to 1). The additional byte supports expansion for additional features or padding for locking the data just written. Padding is never required for direct 15434 encoding because multiple data sets may not be encoded and no expansion bit features are defined (as of this edition) so the Expansion bit is set to ‘0’.

C.7.2 Compaction type

The next three Precursor bits define the encoding method according to Table C.3. For most 15434 data encoding applications (e.g. DIs, TEIs), the 6-bit code table (Code Value 4) will be the most efficient (i.e. the next three bits are “100”).

It may not be possible to encode all data sets using the special 6-bit code table (e.g. some ISBT data). This section reproduces the entire Code Value table from Annex D of ISO/IEC 15962 with the modification defined in this annex, namely Code Value 4 is defined by Table C.3.

Table C.3 — Compaction type codes

Code Value		Name	Description
Decimal	Binary		
0	000	application defined	as presented by the application
1	001	Integer	Not used in direct 15434 encoding
2	010	4-bit code	Not used in direct 15434 encoding
3	011	5 bit code	Not used in direct 15434 encoding
4	100	6 bit code	Modified alphanumeric as shown in Table 7
5	101	7 bit code	US ASCII
6	110	octet string	unaltered 8-bit
7	111	UTF-8 string	External compaction of ISO/IEC 10646

NOTE: the Integer, 4-bit code and 5-bit code compaction schemes (i.e. Code Values 1 to 3) cannot be used because they do not support the <EOT> character.

C.7.3 ISO/IEC 15434 envelope bits

The least significant four Precursor bits define the 15434 formats 0 to 14.

Note: as of the 2006 publication of ISO/IEC 15434, the values “00”, “10”, “11”, “13” and “14” to “99” are not assigned. If Format values greater than 14 are assigned in the future, see Annex D from ISO/IEC 15962 for encoding and decoding procedures.

Annex D
(informative)

ISO/IEC 15434 Direct DI Encoding and Transmission for ISO TC122 Standards

D.1 ISO/IEC 15434 direct DI encoding and transmission for ISO TC122 standards

This annex applies to the encoding of ISO/IEC Format "06" messages when used in ISO/TC 122 applications (e.g. ISO 17364, ISO 17365, ISO 17366, and ISO 17364). This annex uses a subset of the encoding rules defined in Annex C, and only provides support for Data Identifiers (DIs) and a narrower character set for the ISO/IEC 15434 messages. This Annex is reprinted from Annex U of ISO/IEC 15962

D.2 DSFID

Data encoding starts with the DSFID (Data Storage Format Identifier) that encodes the access method and Data-Format. When using direct 15434 encoding, the DSFID is 0x03. See Figure D.1 for how this byte fits into the sequence of the first three encoded bytes.

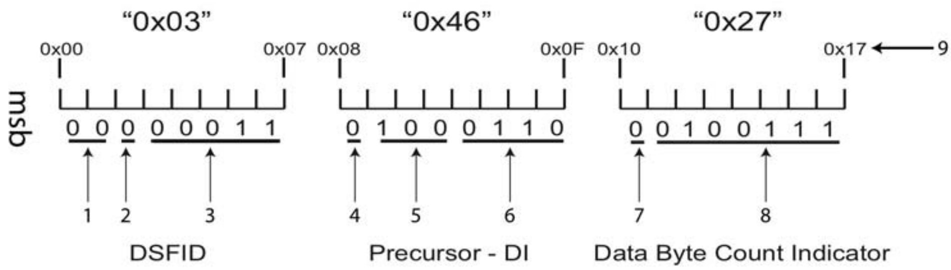
D.3 Precursor byte

Data encoding continues with the Precursor and it encodes the extension bit in the most significant position, the compaction type (next three bits) and the ISO/IEC 15434 Format envelope (four least significant bits). For TC 122 applications the preferred Precursor is byte “0 100 0110” or 0x46 (i.e. extension bit is usually “0”, a compaction type 4 which indicates use of the special 6-bit table defined in this annex and a ISO/IEC 15434 format envelope “06”). See Figure D.1 for how this byte fits into the sequence of the first three encoded bytes.

D.4 Data byte-count indicator

Some air interface protocols allow for optimization in noisy environments by varying the number of bytes sent in each transmission. Therefore it is useful to know at the beginning the number of bytes in tag memory that contain data. For many ISO/IEC15434 DI data encoding applications, the number of bytes needed to encode the data will be a number less than 127 and therefore handled in one byte. For larger messages, two bytes are used where the first byte begins with “1” and the second byte begins with “0” as is fully described in ISO/IEC 15962. The number of bytes is encoded in the 14 remaining bits (e.g. 200 bytes is encoded as "1**0000001** 0**1001000**"). (Only the rightmost 7 bits are used in this binary counting method, and are bold in the previous example)

For example, if a message contains 51 6-bit characters, it will be encoded in 39 bytes (i.e. the last bit of the last character is in the 39th byte and in this case there are six un-encoded bits which require padding). Therefore the data byte-count indicator is 0x27. See Figure D.1 for how this byte fits into the sequence of the first three encoded bytes.



Note 1 Access Method: (#0 as listed in Table 7 – ISO/IEC 15962)

Note 2 Extended Syntax – turns on additional byte of DSFID byte (turned off in this instance)

Note 3 Data Format 03 (ISO/IEC 15434)

Note 4 Extension Bit – not specified in this example

Note 5 Compaction bits (indicating 6-bit table)

Note 6 Format Envelope (specifically DI "06")

Note 7 Byte Count Indicator switch (set to "0" to signify final byte of byte count)

Note 8 Bit values for Byte Count Indicator (variable based on length of data)

Note 9 Physical memory addresses (0x00, 0x07, 0x08, 0x0F, 0x10, and 0x17)

Figure D.1 — Initial three-byte sequence encoded in a segmented memory tag compliant with this Annex – Data Format "03"

D.5 Encoding and decoding

D.5.1 Encode process

1. Starting with a valid ISO/IEC 15434 DI message, strip "[ ) > RS 06 GS" from the front and "RS EOT" from the end.

2. Convert every data character into its code value, using Table D.1

Table D.1 — 6-bit encoding for the 15434 direct encoding method for ISO TC122 applications

Space	100000	0	110000	@	000000	P	010000
<EOT>	100001	1	110001	A	000001	Q	010001
Reserved	100010	2	110010	B	000010	R	010010
<FS>	100011	3	110011	C	000011	S	010011
<US>	100100	4	110100	D	000100	T	010100
Reserved	100101	5	110101	E	000101	U	010101
Reserved	100110	6	110110	F	000110	V	010110
'	100111	7	110111	G	000111	W	010111
(	101000	8	111000	H	001000	X	011000
)	101001	9	111001	I	001001	Y	011001
*	101010	:	111010	J	001010	Z	011010
+	101011	;	111011	K	001011	[	011011
,	101100	<	111100	L	001100	\	011100
-	101101	=	111101	M	001101	]	011101
.	101110	>	111110	N	001110	<GS>	011110
/	101111	?	111111	O	001111	<RS>	011111

Note: This table is perfect subset of Table C.1 but with the following characters and associated bit patterns excluded. These characters are not currently supported by ISO TC 122 applications, and therefore reserved:

— " 100010

— <FS> 100011

- <US> 100100
- % 100101
- & 100110
- ‘ 100111

3. When encoding multiple “06” Format Envelopes (e.g. to represent a message containing several “records” from the same data format in order to describe the subassemblies of a complex part) reduce each internal ISO/IEC 15434 sequence “RS 06 GS” indicating a new “record” to a single <RS> character (encoded as “011111” from Table D.1).

4. Encode an <EOT> pattern after the last encoded data character.

5. Lay out the 6-bit characters as bits and then group them into 8-bit bytes.

6. Add the first 2 or 4 bits of an <EOT> character (i.e. “10” or “1000”) or the entire <EOT> character (i.e. “100001” from the 6-bit character set) to fill un-encoded bits in the last byte, if any.

7. Determine the byte number that contains the last bit of the <EOT> character, convert the decimal count into binary and encode explicitly as the Data byte-count indicator.

8. Encode the DSFID, Precursor, data byte-count indicator, data, and <EOT> padding bytes (if any) into memory.

NOTE: Because only one ISO/IEC 15434 message is allowed to be encoded in a single RFID data carrier, there is no need to encode a zero byte as a terminator after the last data byte.

D.5.2 Decode process

1. Examine the DSFID and Precursor bytes and verify that they are equivalent to “0x03 0x46.

2. Take the next 8 bits and convert to decimal to determine the number of bytes containing data.

3. Starting with the next bit, group the following bits into character bit-sets from the 6-bit code table and continue until the number of bytes containing data has been parsed.

4. Assign data characters according to Table 10 and delete all complete and incomplete <EOT> characters from the end.

5. For any encoded <RS> character that is not immediately followed by “06” and a <GS> character, expand the <RS> to “RS 06 GS”.

6. Add “[ ) > RS 06 GS” to the beginning of the transmission and “RS EOT” at the end

7. Transmit the entire ISO/IEC 15434 compliant message. Optionally, the receiver may wrap the ISO/IEC 15434 message in an OID format as a single data object. When using this option, the complete OID of the message is {1 0 15434 06}.

D.6 Encoding and decoding example

D.6.1 Translation and encoding procedure from ISO/IEC15434 data to Access Method 0 Data Format 3

To prepare a typical DI input message in ISO/IEC 15434 format for encoding using ISO/IEC 15962 Access Method 0 Data-Format 3, the following steps are performed.

- Verify that the input message is a valid ISO/IEC 15434 DI message

- The DSFID indicating Access Method 0 and Data Format 3 is encoded
- The leading message envelope characters "**[ ) > RS 06 GS**" and the trailing "**RS EOT**" are discarded.
- The data is encoded into 6-bit codewords from Table D.1
- Add an <EOT> character
- Add part or all of an <EOT> to fill the last data byte, if necessary.
- Encode the DSFID, Precursor, data byte-count indicator, data, <EOT> and padding into memory.

D.6.2 Decoding and translation procedure from Access Method 0 Data-Format 3 to 15434 data

The system will see this information as ISO/IEC 15434 6-bit DI data by reading the DSFID byte.

- The system discards the DSFID, Precursor and data byte-count indicator at the beginning.
- The encoded bytes are parsed into 6-bit codes, discarding any pad bits and the encoded <EOT> character, and then into data according to Table D.1.
- The system adds "[ ) > RS 06 GS" to the beginning of the transmission and "RS EOT" at the end
- The system transmits the entire ISO/IEC 15434 compliant message

Optionally, the receiver may wrap the entire ISO/IEC 15434 message in an OID format as a single data object.

D.6.3 Data encode and decode example

The following example encodes ISO/IEC 15434 DI data in a TC 122 application with a mandatory <EOT> requirement. It is identical to the example in Annex C.

Starting data:

[)> $^{R}{}_{S}$ 06 $^{G}{}_{S}$ 25SUN043325711MH8031200000000001 $^{G}{}_{S}$ 1T110780 $^{G}{}_{S}$ Q21 $^{G}{}_{S}$ 4LUS $^{R}{}_{S}$ $^{E}O_{T}$

The data on the tag from the above message is as follows (with DIs in **bold** font):

25SUN043325711MH8031200000000001 $^{G}{}_{S}$ **1T**110780 $^{G}{}_{S}$ **Q**21 $^{G}{}_{S}$ **4L**US $^{E}O_{T}$

Where:

- UII = **25S**UN043325711MH8031200000000001
- LOT = **1T**110780
- QTY = **Q**21
- CoO = **4L**US

Data to bit conversion:

There are 51 6-bit characters (50 plus <EOT>) which translates to 39 data-bytes. There is a need to fill six trailing bits for byte alignment so in this case an entire <EOT> character is encoded. See Table D.2.

Table D.2 — Six-bit values assigned by system

DSFID = 0x03	Precursor = 0x46	Data byte-count = 0x27	2	5	S	U	N	0	4	3	3
00000011	01000110	00100111	110010	110101	010011	010101	001110	110000	110100	110011	110011
2	**5**	**7**	**1**	**1**	**M**	**H**	**8**	**0**	**3**	**1**	**2**
110010	110101	110111	110001	110001	001101	001000	111000	110000	110011	110001	110010
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**<GS>**
110000	110000	110000	110000	110000	110000	110000	110000	110000	110000	110001	011110
1	**T**	**1**	**1**	**0**	**7**	**8**	**0**	**<GS>**	**Q**	**2**	**1**
110001	010100	110001	110001	110000	110111	111000	110000	011110	010001	110010	110001
<GS>	**4**	**L**	**U**	**S**	**<EOT>**	**pad**					
011110	110100	001100	010101	010011	100001	100001					

D.6.3.1 Complete contents of tag memory

Using the Access Method 0 Format 3 encoding, including a DSFID, Precursor byte, and 39 bytes of data (compressing 51 6-bit characters, including the <EOT> and six pad bits) the final tag encodation in hexadecimal is as follows.

03 46 27 CB 54 D5 3B 0D 33 CF 2D 77 C7 13 48 E3 0C F1 CB 0C 30 C3 0C 30 C3 0C 31 7B 15 31 C7 0D F8 C1 E4 72 C5 ED 0C 55 38 61

D.6.3.2 Transmitted data

The header characters and the “RS EOT” are reinserted into the message. The following data string is transmitted from the reader.

[)> $^{R}{}_{S}$ 06 $^{G}{}_{S}$ 25SUN043325711MH8031200000000001 $^{G}{}_{S}$ 1T110780 $^{G}{}_{S}$ Q21 $^{G}{}_{S}$ 4LUS $^{R}{}_{S}$ $^{E}O_{T}$

Bibliography

[1] ISO/IEC 646, *Information technology — ISO 7-bit coded character set for information interchange*

[2] ISO/IEC 7816-5:2004, *Identification cards — Integrated circuit cards — Part 5: Registration of application providers*

[3] ISO/IEC 7816-6:2004, *Identification cards — Integrated circuit cards — Part 6: Interindustry data elements for interchange*

[4] ISO 9735 (all parts), *Electronic data interchange for administration, commerce and transport (EDIFACT) — Application level syntax rules*

[5] ISO/IEC 9834-1, *Information technology — Procedures for the operation of object identifier registration authorities: General procedures and top arcs of the international object identifier tree*

[6] ISO/IEC 9834-9, *Information technology — Open Systems Interconnection — Procedures for the operation of OSI Registration Authorities: Registration of object identifier arcs for applications and services using tag-based identification*

[7] ISO/IEC 15417, *Information technology — Automatic identification and data capture techniques — Code 128 bar code symbology specification*

[8] ISO/IEC 15418, *Information technology — Automatic identification and data capture techniques — GS1 Application Identifiers and ASC MH10 Data Identifiers and maintenance*

[9] ISO/IEC 15424, *Information technology — Automatic identification and data capture techniques — Data Carrier Identifiers (including Symbology Identifiers)*

[10] ISO/IEC 15434, *Information technology — Automatic identification and data capture techniques — Syntax for high-capacity ADC media*

[11] ISO/IEC 15459 (all parts) *Information technology — Unique identifiers*

[12] ISO/IEC 15961, *Information technology — Radio frequency identification (RFID) for item management — Data protocol: application interface*

[13] ISO/IEC 15962, *Information technology -- Radio frequency identification (RFID) for item management — Data protocol: data encoding rules and logical memory functions*

[14] ISO/IEC 15963, *Information technology — Radio frequency identification for item management — Unique identification for RF tags*

[15] ISO/IEC 16022, *Information technology — Automatic identification and data capture techniques — Data Matrix bar code symbology specification*

[16] ISO 17364, *Supply chain applications of RFID — Returnable transport items (RTIs)*

[17] ISO 17365, *Supply chain applications of RFID — Transport units*

[18] ISO 17366, *Supply chain applications of RFID — Product packaging*

[19] ISO 17367, *Supply chain applications of RFID — Product tagging*

[20] ISO/IEC 18000 (all parts), *Information technology — Radio frequency identification for item management*

[21] ISO/IEC 18004, *Information technology -- Automatic identification and data capture techniques – QR Code 2500 bar code symbology specification*

[22] ISO/IEC/IEEE 21451-7, *Information technology — Smart transducer interface for sensors and actuators — Part 7: Transducer to radio frequency identification (RFID) systems communication protocols and Transducer Electronic Data Sheet (TEDS) formats*

[23] ISO/IEC 24724, *Information technology -- Automatic identification and data capture techniques – GS1 Data Bar bar code symbology specification*

[24] ISO/IEC 24729-1, *Information technology — Radio frequency identification for item management — Implementation guidelines – Part 1: RFID-enabled labels and packaging supporting ISO/IEC 18000-6C*

[25] ISO/IEC 24729-3, *Information technology — Radio frequency identification for item management — Implementation guidelines – Part 3: Implementation and operation of UHF RFID Interrogator/systems in logistics application*

[26] ISO/IEC 24791-1, *Information technology — Radio frequency identification (RFID) for item management — Software system infrastructure — Part 1: Architecture*

[27] ISO/IEC 24791-2, *Information technology — Radio frequency identification (RFID) for item management — Software system infrastructure — Part 2: Data management*

[28] ISO/IEC 24791-3, *Information technology — Radio frequency identification (RFID) for item management — Software system infrastructure — Part 3: Device management*

[29] ISO/IEC 24791-5 [2], *Information technology — Radio frequency identification (RFID) for item management — Software system infrastructure — Part 5: Device interface*

[30] Air Transport Association (ATA) *SPEC 2000, International Specification, Integrated Data Processing Materiels Management,* Chapter 9

[31] ANS MH10.8.2, *Data Identifier and Application Identifier Standard*

[32] GS1, *GS1 General Specifications*

[33] GS1 EPCglobal, *GS1 EPC Tag Data Standards, Version 1.5*

2 To be published.

ICS 35.040
Price based on 32 pages