INTERNATIONAL STANDARD

ISO 28560-2

Fourth edition
2023-06

Information and documentation — RFID in libraries —

Part 2: Encoding of RFID data elements based on rules from ISO/IEC 15962

Information et documentation — RFID dans les bibliothèques —

Partie 2: Encodage des éléments de données RFID fondé sur les règles de l'ISO/CEI 15962

Reference number
ISO 28560-2:2023(E)

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

ISO draws attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO takes no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents. ISO shall not be held responsible for identifying any or all such patent rights.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 46, *Information and documentation*, Subcommittee SC 4, *Technical interoperability*.

This fourth edition cancels and replaces the third edition (ISO 28560-2:2018), of which it constitutes a minor revision.

The changes are as follows:

— updates have been made including harmonization of the data format value.

A list of all parts in the ISO 28560 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user’s national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

Libraries are implementing radio frequency identification (RFID) as item identification to replace bar codes. RFID streamlines applications like user self-service, security, and materials handling. This standard data model for encoding information on RFID tags increases the cost-effectiveness of the technology within libraries, particularly through greater interoperability of RFID tags and equipment, and enhance support for resource sharing between libraries.

This document deals with the encoding of data elements in a flexible manner using encoding rules that are specified in ISO/IEC 15962. ISO 28560-1 defines the set of mandatory and optional data elements.

ISO 28560-3 and this document are mutually exclusive with respect to an RFID tag being applied to a loan item. In other words, the RFID tag is encoded according to the rules of this document, or to the rules of ISO 28560-3, or to some proprietary rules. Depending on the technologies being used, and other features of tags that are claiming conformance with this document, the reading system can achieve a degree of interoperability.

This document provides essential standards-based information about RFID in libraries. Ongoing advice needs to be provided because of the evolving nature of RFID technology, and the opportunities to migrate between different types of legacy system and encoding rules of this document.

Information and documentation — RFID in libraries —

Part 2:
Encoding of RFID data elements based on rules from ISO/IEC 15962

1 Scope

This document specifies a data model and encoding rules for the use of radio frequency identification (RFID) tags for items appropriate for the needs of all types of libraries (including national, academic, public, corporate, special, and school libraries). The rules for encoding a subset of data elements taken from the total set of data elements defined in ISO 28560-1 are based on ISO/IEC 15962, which uses an object identifier structure to identify data elements.

This document defines the technical characteristics required to encode the data elements defined in ISO 28560-1 according to ISO/IEC 15962. These subsets of data elements can be different on different items in the same library. The encoding rules also enable the optional data to be organized on the RFID tag in any sequence. In addition, the encoding rules provide for flexible encoding of variable length and variable format data.

This document provides essential standards-based information about RFID in libraries. A source of additional information about implementation issues is provided in Annex A.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 15961-1, *Information technology — Radio frequency identification (RFID) for item management: Data protocol — Part 1: Application interface*

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

ISO/IEC 18000-3, *Information technology — Radio frequency identification for item management — Part 3: Parameters for air interface communications at 13,56 MHz*

ISO/IEC 18046-3, *Information technology — Radio frequency identification device performance test methods — Part 3: Test methods for tag performance*

ISO/IEC 18047-3, *Information technology — Radio frequency identification device conformance test methods — Part 3: Test methods for air interface communications at 13,56 MHz*

ISO 28560-1, *Information and documentation — RFID in libraries — Part 1: Data elements and general guidelines for implementation*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org/

3.1
access method
component of the *DSFID* (3.8) that is responsible for declaring the ISO/IEC 15962 compaction and encoding rules on an RFID tag

3.2
air interface protocol
rules of communication between an RFID interrogator and the RFID tag of a particular type, covering: frequency, modulation, bit encoding, and command sets

3.3
application command
instruction issued from the application to the ISO/IEC 15962 data protocol processor in order to initiate an action or operation with the RFID tag(s) via the interrogator

3.4
AFI
application family identifier
mechanism used in the data protocol and the *air interface protocol* (3.2) to select a class of RFID tags relevant to an application, or aspect of an application, and to ignore further communications with other classes of RFID tags with different identifiers

3.5
arc
specific branch of an object identifier tree, with new arcs added as required to define a particular object

Note 1 to entry: The top three arcs of all object identifiers are compliant with ISO/IEC 9834-1, ensuring uniqueness.

3.6
data format
mechanism used in the data protocol to identify how *object identifiers* (3.11) are encoded on the RFID tag, and (where possible) identify a particular data dictionary for the set of relevant object identifiers for that application

Note 1 to entry: The data format declares the *Root-OID* (3.13) in an efficient manner, so that a complete *object identifier* (3.11) can be reconstructed for external communications.

3.7
data protocol process
implementation of the processes defined in ISO/IEC 15962, including data compaction, formatting, support of the command/response unit, and an interface to the tag driver

3.8
DSFID
data storage format identifier
code that consists of, at least, the *access method* (3.1) and *data format* (3.6)

3.9
digital vandalism
unauthorized modification of data on an RFID tag that either renders it unusable or falsely represents another identifier

3.10
metadata
type of data or information about data

Note 1 to entry: In the context of this document, *metadata* (3.10) can be the *Relative-OID* (3.12) in relation to the data, the precursor in relation to the compacted and encoded bytes, or the *AFI* (3.4) and *DSFID* (3.8) in relation to the data.

3.11
object identifier
value (distinguishable from all other such values), which is associated with an object

3.12
Relative-OID
particular *object identifier* (3.11) that constitutes the remaining *arcs* (3.5) after the *Root-OID* (3.13)

3.13
Root-OID
particular *object identifier* (3.11) that constitutes the first, second, and subsequent common *arcs* (3.5) of a set of object identifiers (hence the common root)

3.14
tag driver
implementation of the process to transfer data between the data protocol processor and the RFID tag

4 Applicability and relationship with other systems

Figure 1 gives an overview of the relationship of this document with other systems. This document defines a set of technical features while addressing a number of operational issues. This document interfaces with four other activities, but with a clearly defined overlap. These other activities are:

— the circulation of library materials;

— the data requirements of publishers, printers, and other suppliers;

— the interlibrary loan processes; and

— the details of borrowers, including membership cards.

Figure 1 also shows that there is a direct relationship with supply chain activities, and internally within the library with RFID circulation devices and the library management system including interfaces such as SIP2 and NCIP.

As the use of RFID in libraries moves\ towards a more standardized approach as defined in this document, the characteristics and architecture systems change compared to those already established.

To achieve interoperability with equipment and software, the required features include:

— the air interface protocol, which defines the way readers and tags communicate with one another;

— the data protocol, which defines the encoding rules that convert application-based data to the encoded bytes on the RFID tag; the data protocol also defines metadata features in the RFID tag to protect the integrity of RFID for library systems in relation to other RFID applications;

— the set of data elements that form the dictionary from which individual libraries can choose those that are most appropriate for their operation.

4.3 By adopting this document, libraries will have increased flexibility with a number of features as follows.

— Beyond the minimum of mandatory data elements defined in this document, libraries are able to choose from the optional data elements those that are more appropriate to its application, even varying these for different types of item.

— Libraries should be able to rank the optional data elements into an appropriate order for encoding on the RFID tag to support fast transactions across the air interface.

— Libraries have a greater choice of interoperable RFID equipment, and should be able to select RFID tags with an appropriate size of memory.

— Some degree of choice in the types of security system becomes a library responsibility.

— Libraries with an installed base of RFID data capture is offered options on how to migrate to the more open standard solution.

— The library community, as a whole and through developments of this document, is provided with future options to cope with changes within the RFID equipment as the technology develops. This includes ensuring that new open systems applications do not corrupt the established base of RFID systems in libraries.

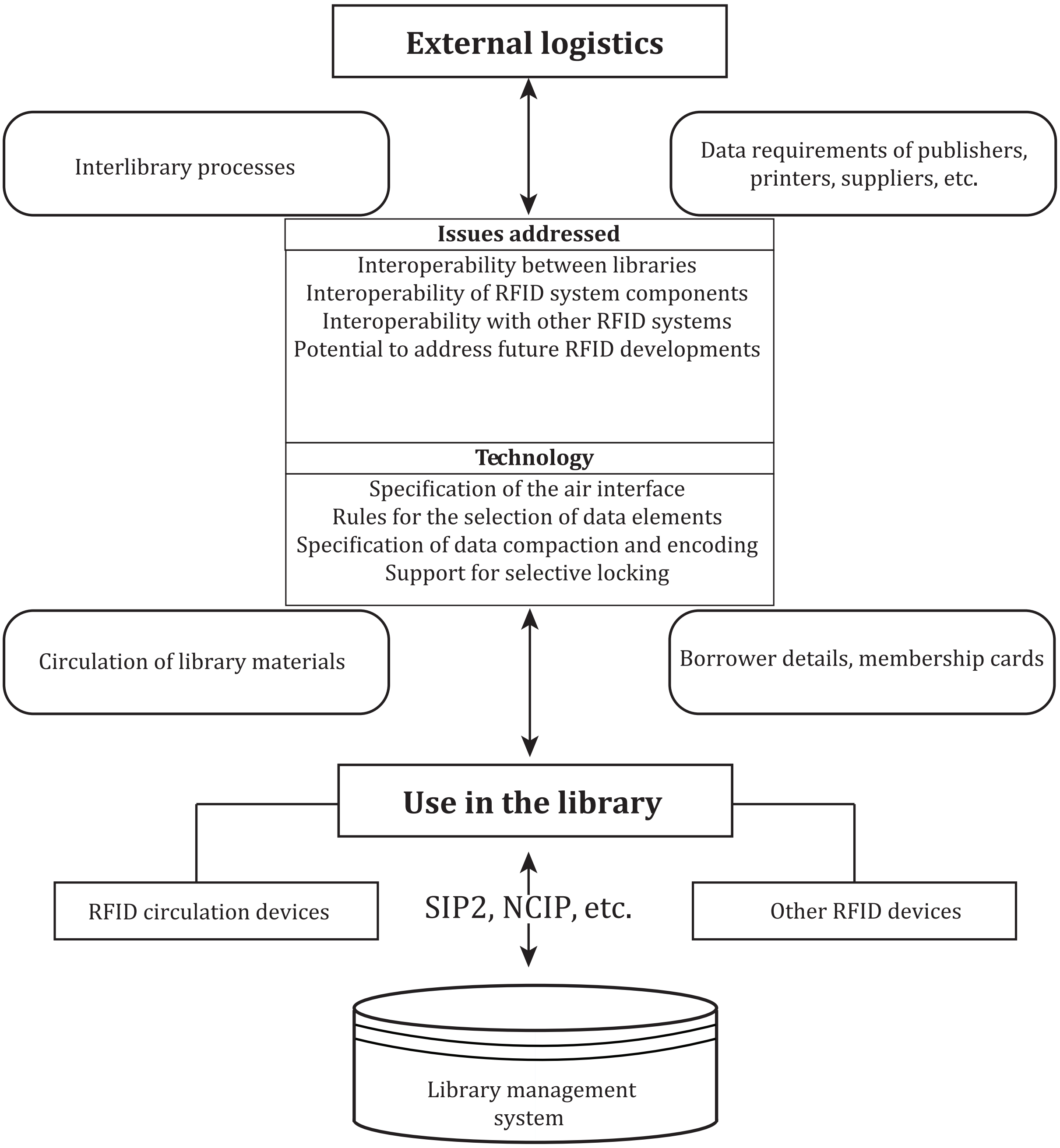

Figure 1 — Relationship of this document with other systems

5 Requirements

5.1 Data elements

The data elements shall be in accordance with ISO 28560-1.

NOTE There is a degree of flexibility in using locally defined codes that enable enhancements and variations to be implemented while still complying with the basic set of data elements.

5.2 RFID air interface

5.2.1 General

The air interface for compliant tags is specified in ISO/IEC 18000-3, especially the specification for Mode 1.

For migration purposes, additional non-compliant air interfaces used in legacy systems can be supported during a transition period, which is permitted to remain in place for years as necessary.

5.2.2 Air interface conformance

The air interface conformance shall be tested in accordance with the procedures of ISO/IEC 18047-3.

5.2.3 Tag performance

Where there are requirements for testing tag performance, these shall be done in accordance with ISO/IEC 18046-3.

5.3 Data protocol

ISO/IEC 15961-1 specifies the application commands that shall be used to define the communication requirements between the application and the RFID tag. The relevant commands are specified in Annex B.

The process rules of ISO/IEC 15962 shall be used to encode and decode data from the RFID tag. In particular, the following constraints shall apply.

— The only encoding rules shall be based on the No-directory access method. No alternative access method shall be supported until this document is revised.

— Both the hard-coded and software-encoded DSFID shall be supported, depending on the capabilities of the RFID tag.

5.4 RFID readers

In order to achieve interoperability, RFID readers shall be based on open architecture RFID standards. Particular standards are specified in this document. This means that any one manufacturer's reading/writing equipment shall be able to read or write to any other manufacturer's RFID tags, and that any manufacturer's RFID tags shall be able to be read and/or programmed by any other manufacturer's reader/writer.

6 Data elements

6.1 General

The set of data elements that comprises the data dictionary for this document is fully described in ISO 28560-1 and repeated in outline in Table 1. Only one data element is mandatory, the primary item identifier. All others are optional, but can be selected to meet the requirements of individual libraries, and/or for particular items.

Table 1 shows the Relative-OID value, the format for input data, and advice about locking the data element as an encoded data set on the RFID tag. A maximum length of 255 characters should apply to all data elements that have a variable length display format.

Table 1 — List of data elements

N[a]	Name of the data element	Status	Display format	Lock
1	Primary item identifier	Mandatory	Variable length alphanumeric Character set = ISO/IEC 646 International Reference Version (IRV)	Should be locked
2[c]	Content parameter	Optional	Bit mapped code (see 6.3)	Optional
3[b,c]	Owner institution (ISIL)	Optional	Variable length field (maximum of 16 characters) based on ISO 15511	Optional
4	Set information	Optional	{Total in set/part number} structure (maximum ≤ 255)	Optional
5[c]	Type of usage	Optional	Single octet (coded list)	Optional
6	Shelf location	Optional	Variable length alphanumeric Character set = ISO/IEC 646 IRV	Optional
7	ONIX media format	Optional	Two uppercase alphabetic characters	Optional
8	MARC media format	Optional	Two lowercase alphabetic characters	Optional
9	Supplier identifier	Optional	Variable length alphanumeric Character set = ISO/IEC 646 IRV	Optional
10	Order number	Optional	Variable length alphanumeric Character set = ISO/IEC 646 IRV	Optional
11[b,c]	ILL borrowing institution (ISIL)	Optional	Variable length field (maximum of 16 characters) based on ISO 15511	Not locked
12	ILL borrowing transaction number	Optional	Variable length alphanumeric Character set = ISO/IEC 646 IRV	Not locked
13	GS1 product identifier	Optional	Fixed length 13 numeric digit field	Optional
14	Alternative unique item identifier	Reserved for future use	—	—
15	Local data A	Optional	Variable length alphanumeric Character set = ISO/IEC 646 IRV, ISO/IEC 8859-1, or UTF-8	Optional
16	Local data B	Optional	Variable length alphanumeric Character set = ISO/IEC 646 IRV, ISO/IEC 8859-1, or UTF-8	Optional
17	Title	Optional	Variable length alphanumeric Character set = ISO/IEC 646 IRV, ISO/IEC 8859-1, or UTF-8	Optional
18	Product identifier local	Optional	Variable length Alphanumeric Character set = ISO/IEC 646 IRV	Optional
19[c]	Media format (other)	Optional	Single octet (coded list)	Optional
20[c]	Supply chain stage	Optional	Single octet (coded list)	Optional
21	Supplier invoice number	Optional	Variable length alphanumeric Character set = ISO/IEC 646 IRV	Optional
22	Alternative item identifier	Optional	Variable length alphanumeric Character set = ISO/IEC 646 IRV	Optional
23	Alternative owner institution	Optional	Variable length alphanumeric Character set = ISO/IEC 646 IRV	Optional
24	Subsidiary of an owner institution	Optional	Variable length alphanumeric Character set = ISO/IEC 646 IRV	Optional
25	Alternative ILL borrowing institution	Optional	Variable length alphanumeric Character set = ISO/IEC 646 IRV	Not locked

Table 1 *(continued)*

N[a]	Name of the data element	Status	Display format	Lock
26	Local data C	Optional	Variable length alphanumeric Character set = ISO/IEC 646 IRV, ISO/IEC 8859-1, or UTF-8	Optional
27	Not defined	Reserved for future use	—	—
28	Not defined	Reserved for future use	—	—
29	Not defined	Reserved for future use	—	—
30	Not defined	Reserved for future use	—	—
31	Not defined	Reserved for future use	—	—

[a] This column specifies the data element number (*N*) or the Relative-OID value, i.e. the number identifying the data element, as defined in ISO 28560-1.

[b] The ISIL, as used for Relative-OID values 3 and 11, is presented and displayed according to the characters defined in ISO 15511. A special encoding scheme, as defined in 6.4, is used to compact efficiently the complex ISIL character string.

[c] These data elements require the ISO/IEC 15962 application-defined compaction to preserve the integrity of some pre-processing that has been applied to them.

6.2 Primary item identifier

The primary item identifier is a mandatory data element defined in ISO 28560-1.

This is the only mandatory data element that is required to be encoded to be compliant with this document. The format is variable length, and the alphanumeric characters can be any from ISO/IEC 646 International Reference Version (also known as US-ASCII). Although the encoding rules support any length of primary item identifier, shorter codes and all-numeric codes encode more efficiently, requiring less memory and enabling faster transactions across the air interface. Although locking the primary item identifier is optional, under normal circumstances, this data element should be locked to prevent various forms of digital vandalism. The primary item identifier shall be encoded as the first data element on the RFID tag to allow for faster transactions across the air interface by invoking a Read-First-Object(s) argument in the read command (see B.5).

6.3 Content parameter

The content parameter is an optional data element used to declare the Relative-OID values that are encoded on the RFID tag, and for the purposes of this document is used as an OID index. It should be used if additional data elements are encoded on the RFID tag. If used, it can be an aid to faster reading, because it indicates the presence or absence of a particular data element. If the desired data element is encoded on the tag, then additional reading is required, whereas if the OID index indicates that it is not on the tag, the wasted transaction time can be eliminated.

The index, itself, consists of a bit sequence, where each bit position is associated with a particular Relative-OID. If the bit position is set “1”, then the Relative-OID and associated data object is encoded on the RFID tag. As Relative-OID 1 is mandatory and Relative-OID 2 is this particular data element, the bit map begins at Relative-OID 3. An example is shown in Figure 2.

Relative-OID	3	4	5	6	7	8	9	10	11							
Bit 1 = encoded	1	0	0	0	0	1	0	0	1	n/a	n/a	n/a	n/a	n/a	n/a	n/a

Key

n/a padded bits to indicate not encoded or not applicable (rounded to 8-bit boundaries)

Figure 2 — Example of OID index bit map

In the example in Figure 2, the OID index indicates that Relative-OID values 3, 8 and 11 are encoded. Irrespective of whether the data dictionary includes other Relative-OID values, the bit map can be truncated at this last Relative-OID that is encoded. It is also necessary to round up the bit map to 8-bit boundaries for encoding on the RFID tag.

If this data element is encoded on the RFID tag, it should be in the second position so that the data capture system can be set up to read the primary item identifier and the OID index in a single read process. The OID index should only be locked if the information on the RFID tag is certain to remain unchanged. This data element provides no information about the sequence of the encoded data elements, nor their size. In the example in Figure 2, the encoding sequence could be Relative-OID value 8 followed by 11, followed by 3.

6.4 Owner institution (ISIL)

The owner institution data element represents the ISIL code as specified in ISO 15511. For this document, the ISIL code is introduced into the RFID encoding process in a structure defined in accordance with the rules of ISO 15511. This means that the hyphen (present in every ISIL code following the two-character country code) is presented in the application commands.

To achieve efficient encoding, the ISIL shall be pre-encoded to rules defined in Annex C. This annex also applies to the ILL borrowing institution (6.12). In addition to providing details of the encoding scheme, Annex C also provides advice about interfacing with ISO/IEC 15962 encoders and decoders.

The use of these codes assumes (for example) an external interlibrary loans (ILL) system capable of tracking the item based on the unique combination of its primary item identifier and owner institution. This element is optional where items are not included in an ILL scheme but required when items are required to be issued on ILL using RFID. While it might be deemed necessary to lock this data element, this is left optional as some libraries can choose to leave the data element unlocked so that it can be changed, if necessary, as a result of library mergers or transfer of collections, etc. Other applications can also make use of the ISIL.

6.5 Set information

The set information is presented in two components, which are

a) the total number of parts, and

b) the ordinal part number, with a maximum of 255 parts.

ISO 28560-1 defines various examples of encoding, particularly where not all the parts of the set carry an RFID tag.

If the total number of parts is 9 or less, then the user data can be presented as a two-digit code to reduce the encoding requirement. If the total number of parts is between 10 and 99, then the user data are presented as a four-digit code, with the lowest ordinal values shown as 00 to 09. If the total number of parts is between 100 and 255, then the user data are presented as a six-digit code. If the ordinal value is less than 100, it is prefixed by leading zeros to create a three-digit number.

6.6 Type of usage

The type of usage data element is defined in ISO 28560-1, together with the supporting coded list of values for this data element. The code in ISO 28560-1 is presented as an alphanumeric code, but is actually a single-byte hexadecimal code and is encoded in this manner.

6.7 Shelf location

The shelf location is a variable length field that is used to identify the location code of a shelving system of the owning institution.

6.8 ONIX media format

The ONIX media format data element represents an ONIX media descriptor of two uppercase alphabetic characters. A reference source for the code list is provided in ISO 28560-1.

6.9 MARC media format

The MARC media format data element represents a MARC category of material descriptor of two lowercase alphabetic characters. A reference source for the code list is provided in ISO 28560-1.

6.10 Supplier identifier

The supplier identifier is a variable length field that can be used for a locally designated identification number relating to the supplier of the library material. It can be left permanently written to the tag or it can be used only temporarily during an acquisitions process.

6.11 Order number

The order number is a variable length field that can be used for a locally designated order number meaningful to the library and to the supplier of the library material. It can be left permanently written to the tag or it can be used only temporarily during an acquisitions process.

6.12 ILL borrowing institution

The ILL borrowing institution is represented by the ISIL code in accordance with ISO 15511. The data are presented according to the rules defined in 6.4 (for the owner institution). This data element shall not be locked.

6.13 ILL transaction number

The ILL transaction number is assigned by the lending institute to identify an interlibrary loan transaction. The structure of the number is locally defined. The data element shall not be locked.

6.14 GS1 product identifier

The GS1 product identifier data element is used to store the GTIN-13 code, commonly seen on retail products in a bar code format on books and other media products. A more detailed definition is provided in ISO 28560-1. The GTIN-13 code is always presented as a 13-digit code (i.e. with leading zeros, if necessary) for input into the ISO/IEC 15962 encoding process.

NOTE 1 Since January 2007, the ISBN has formally changed from being a 10-digit code (sometimes with an X check character) into a 13-digit code, as represented in the GTIN-13 code.

NOTE 2 The GTIN-13 code is more popularly understood in the United States as the UPC code, and in other parts of the world as the EAN-13 code.

6.15 Alternative unique item identifier

This data element is reserved for possibly encoding in different tag architectures.

6.16 Local data

The local data elements (A, B and C) are each variable length fields that can be used for any locally defined purpose and as such there is no external application of this data object. Table 2 identifies the parameters for the local data elements.

Table 2 — Local data element parameters

Data element	Relative-OID	Category	Format	Lock
Local data A	15	Optional	Variable length alphanumeric field Character set = ISO/IEC 646 IRV, ISO/IEC 8859-1, or UTF-8	Optional
Local data B	16	Optional	Variable length alphanumeric field Character set = ISO/IEC 646 IRV, ISO/IEC 8859-1, or UTF-8	Optional
Local data C	26	Optional	Variable length alphanumeric field Character set = ISO/IEC 646 IRV, ISO/IEC 8859-1, or UTF-8	Optional

The rules defined in 6.17 for encoding characters from different character sets shall apply to these data elements.

6.17 Title

The title data element is a variable length field used to identify the title or name of the item. The format can be UTF-8 to allow for titles to be encoded in a language other than those based on the extended Latin alphabet. The following advice is intended to assist with encoding efficiency.

— If possible, a title should be defined using the ISO/IEC 646 IRV (US ASCII) character set. It is also recommended that all uppercase characters be used as this encodes more efficiently.

— If it is not possible to use the ISO/IEC 646 IRV (US ASCII) character set, then consideration should be given to using ISO/IEC 8859-1, which is the default encoding set for ISO/IEC 15962. The default character set of ISO/IEC 15962 is defined in ISO/IEC 8859-1 (as specified in 7.4.4). As this clause on compaction clearly defines, if any graphical character of the data element is outside the range of ISO/IEC 8859-1, then UTF-8 needs to be declared as the compaction scheme.

— UTF-8 should only be declared for titles that cannot be defined using the ISO/IEC 8859-1 character set.

— For all defined titles, in situations where tag memory is small, a locally defined and administered limit can be placed on the length of this field. The length should be the shortest that is practical to satisfactorily identify the item from a small set of items (e.g. one item from six as a library borrower exits the security gates and triggers an alarm due to a processing error).

Different program languages, and therefore software, support the storage of these characters is one of two ways.

— As 16-bit code points compliant with ISO/IEC 10646 (also known as Unicode). If the LMS/ILS or the ISO 28560-2 encoding supports these characters, then the encoder needs to be responsible for the transformation to UTF-8.

— If the LMS/ILS holds the characters already transformed as UTF-8 code points, then it is essential to check that ISO 28560-2 encoding process is completely compatible.

NOTE The ISO/IEC 646 IRV character set is a perfect subset of both ISO/IEC 8859-1 and of UTF-8.

The parameters for the item title data element are defined in Table 3.

Table 3 — Title data element parameters

Data element	Relative-OID	Category	Format	Lock
Title	17	Optional	Variable length alphanumeric field Character set = ISO/IEC 646 IRV, ISO/IEC 8859-1, or UTF-8	Optional

6.18 Product identifier (local)

For items that do not have a GTIN-13 code, or where one cannot be constructed independently as is possible with the ISBN, the product identifier (local) data element can be used. This enables information systems linked to specific local code structures to be supported by the RFID system.

6.19 Media format (other)

The media format (other) data element represents any media descriptor other than ONIX or MARC. It is only used if either of the two more standard codes is not supported locally. The assigned values of this data element are defined in ISO 28560-1.

6.20 Supply chain stage

The supply chain stage is a single octet that is used to identify the current stage of the supply chain in which the RFID tag resides. The code list is provided in ISO 28560-1.

For encoding purposes, the hexadecimal values in that code list shall be encoded using the application-defined compaction scheme. The code point 00_{HEX} for undefined does not apply to this document and should not be encoded.

6.21 Supplier invoice number

The supplier invoice number is a variable length field that can be used for a locally designated invoice number meaningful to the library and to the supplier of the library material. It can be left permanently written to the tag or it can be used only temporarily during an acquisitions process.

6.22 Alternative item identifier

The alternative item identifier is a variable length field that can be used for a locally designated optional identifier. The ID can be temporary and have only local meaning as during an acquisitions process or it can contain other identifiers as deemed necessary.

6.23 Alternative owner institution

The alternative owner institution is used, for example, where a library identifier scheme pre-dates the ISIL. This element is optional where items are not included in an ILL scheme but required when items are issued on ILL. While it might be deemed necessary to lock this data element, this is left optional. Some libraries can choose to leave the data element unlocked so that it can be changed, if necessary, as a result of library mergers or transfer of collections, or a future migration to the ISIL code.

6.24 Subsidiary of an owner institution

The subsidiary of an owner institution data element is used to refine the identity to a level lower than the ISIL. As such it is an internal code defined locally.

6.25 Alternative ILL borrowing institution

The alternative ILL borrowing institution is a variable length field that can be used for a locally designated optional identifier, where an ISIL cannot be used. This data element shall not be locked.

6.26 Other reserved data elements

Data elements with the Relative-OID values 27 to 31 are reserved.

7 Data encoding

7.1 Data protocol overview

The data shall be written to, and read from, the RFID tag using facilities functionally equivalent to the commands and responses defined in ISO/IEC 15961-1, though transfer encoding is not required. This allows libraries complete flexibility in selecting from the present set of optional data elements as defined in this document, and for supporting new data elements, should these be added at a future date. This flexibility can be implemented for different loan items, and changed over a period of time, depending on the requirements of the library system.

The encoded byte stream on the RFID tag shall be encoded in accordance with the rules of ISO/IEC 15962. These rules are implemented automatically through a system that has both ISO/IEC 15961-1 and ISO/IEC 15962 as part of the complete data protocol.

NOTE The adoption of this data protocol, together with other referenced RFID standards, allow libraries to migrate more easily if any developments in RFID technology are considered suitable for the library community. This is because the data protocol has been designed to be independent of RFID air interface protocols and tag architectures. As new RFID technology has been standardized, the core components of the ISO/IEC 15961 series and ISO/IEC 15962 remain constant. New features are supported by interface mechanisms (known as tag drivers) being specified in ISO/IEC 15962 and any new features being supported more generically in the commands of ISO/IEC 15961-1 and the processes defined in ISO/IEC 15962.

7.2 Data constructs

7.2.1 General

ISO/IEC 15961-2 requires that a set of RFID data constructs be registered for applications that use the data protocol. The four RFID data constructs are described in 7.2.2 to 7.2.6, together with their particular code values that have been assigned by the ISO/IEC 15961 registration authority for use for RFID for libraries.

7.2.2 AFI

The AFI is a single-byte code used as a tag selection mechanism across the air interface to minimize the extent of communication transaction time with tags that do not carry the relevant AFI code.

The AFI value $C2_{HEX}$ has been assigned under the registration of ISO/IEC 15961-2 explicitly for library use.

A library can use the AFI in one of the two ways.

— It can use a single AFI, the value $C2_{HEX}$ assigned under the registration of ISO/IEC 15961-2. This distinguishes library loan items from all other items using RFID in item management systems. This avoids the risk of an RFID reader in another domain reading the RFID tag on a loan item and confusing the encoded content with data for its own application. It also enables a library system to reject items that carry a different AFI code, possibly from another domain visited by a client. If a single AFI is used, then a library can wish to lock this. Before they do, they should give consideration to the item's use in other libraries through cooperative arrangements or interlibrary loans. The recipient library might want to use the AFI for security while the item is in their possession even though the donor library does not.

— The AFI can additionally be used as part of a “security system” where the AFI value $C2_{HEX}$ is written to tags for items that are on loan to a client. When the books are returned, an in-stock AFI (07_{HEX}, as defined in ISO/IEC 15961-3) is written to the tag. In this case, the AFI shall not be locked.

7.2.3 Data format

The data format is used as a mechanism to enable object identifiers to be encoded in a truncated or short form. The data format value 6 (06_{HEX}) has been assigned under the registration of ISO/IEC 15961-2 explicitly for library use. The data format is part of a single-byte value defined as the DSFID and defined in 7.2.6.

7.2.4 Object identifier for library applications

The object identifier structure used in the RFID data protocol ensures that each data element is unique not only within a domain such as a library system for this document, but between all domains. The object identifier can be split into two component parts. The Relative-OID, as defined in Table 1, only distinguishes between data elements within a particular domain, whereas by prefixing this with a Root-OID, the data element becomes unique within all object identifiers. The common Root-OID that has been assigned under the registration of ISO/IEC 15961-2 explicitly for library use is:

1 0 15961 6

For all object identifiers specified in this document, only the Relative-OID needs to be encoded. Software designed specifically for the library community probably only requires the Relative-OID to be provided in commands.

If a library system uses generic ISO/IEC 15962 encoding and decoding software, the full object identifier might be required in commands and responses. In such cases, the Root-OID shall be prefixed to the Relative-OID value to create the object identifier. The RFID tag encoding is still efficient, because the data format truncates the Root-OID during the encoding process and reconstructs it in the decode process. Even under this more generic process, only the Relative-OID is actually encoded on the RFID tag to distinguish between data elements.

7.2.5 Object identifier for the primary item identifier

The primary item identifier needs to have its full object identifier structure registered as part of the rules of ISO/IEC 15961-2. This is to enable this object identifier to be labelled as the unique item identifier (UII). This provides a generic distinction between the UII and all other data elements, and ensures consistency for this identification, which might be relevant with future developments of RFID technology.

The object identifier registered for the primary item identification code is:

1 0 15961 6 1

The Relative-OID “1” for the primary item identifier (see 6.2) is consistent with this registration.

7.2.6 DSFID and access method

The DSFID is a single-byte code that has two component parts relevant to this document:

— the data format, as defined in 7.2.3, which is represented in the last five bits of the DSFID;

— the access method, which is represented in the first two bits of the DSFID, and which determines how data are structured on the RFID tag; the access method that is currently defined for this document is 00 = No-directory, where the encoded bytes are concatenated in a continuous byte stream.

The use of the OID index (see 6.3) reduces the advantage of using the directory access method, so the directory access method is not supported in this document. Other access methods are included in the second edition of ISO/IEC 15962. This document shall not support any additional access method without a formal amendment. Such an amendment shall include a migration path for the introduction and support of a new access method.

Locking the DSFID results in both the access method and data format being permanently set for the RFID tag. Any decision to lock or unlock the DSFID needs to take into consideration advice provided in 8.1.4.

7.3 ISO/IEC 15961-1 commands and responses

ISO/IEC 15961-1 specifies commands, and their responses, from the application to the ISO/IEC 15962 rules and interrogator. These commands cover the writing, reading, and modifying of data. These commands and their responses are designed to operate at a higher level than the air interface commands and responses, which only deal with bytes and blocks.

The application commands enable an object identifier and associated object (data) to be defined in a way understood by an application. Additional command arguments support features to enable the application to instruct the encoder to compact the data, to lock the data, and to avoid encoding duplicate data. A list of ISO/IEC 15961-1 commands that are relevant for the ISO/IEC 18000-3 Mode 1 RFID tags are specified in Annex B.

All the arguments in the command are essential to achieve compliant encoding (e.g. instructions to lock a specific data set, or to determine the sequence of data elements). However, it has been approved that ISO/IEC 15961-1 no longer requires a detailed interface mechanism with ISO/IEC 15962, as currently incorporated in the first edition of these International Standards. This means that the explicit ASN.1 transfer encoding rules into the ISO/IEC 15962 encoder are no longer required to claim compliance. Systems providers now have a simpler and more flexible manner to implement the encoding on the RFID tag, but are still required to encode based on the relevant command arguments. The conformance requirements (see Clause 5) are in line with this approach.

7.4 ISO/IEC 15962 encoding rules

7.4.1 General

The encoding rules are designed to achieve a combination of flexibility and efficiency for the bytes that are encoded on the RFID tag. In particular:

— data are compacted efficiently using a defined set of compaction techniques that reduce the encoding on the RFID tag and across the air interface;

— data formatting minimizes the encoding of the object identifiers on the RFID tag and on the air interface, but still provides complete flexibility for identifying specific data without the recourse to rigid message structures.

The syntax associated with the encoding rules effectively creates a self-defining message structure for each RFID tag. This allows optional data from the application data dictionary to be selected. It also enables variable length data to be encoded, and for different formats of data (e.g. numeric or alphanumeric) to be encoded as efficiently as possible and intermixed in the same RFID system. The rules of ISO/IEC 15962 make it possible to correctly interpret the data on the RFID tag without any prior knowledge of what is encoded on the tag. This is an important feature that enables interoperability of devices, and allows this document to add new data elements without changes to the equipment. It also allows an individual library to vary the choices of data elements without the need for any major update.

Figure 3 illustrates the basic architecture of the data protocol. The components of ISO/IEC 15962 are discussed below.

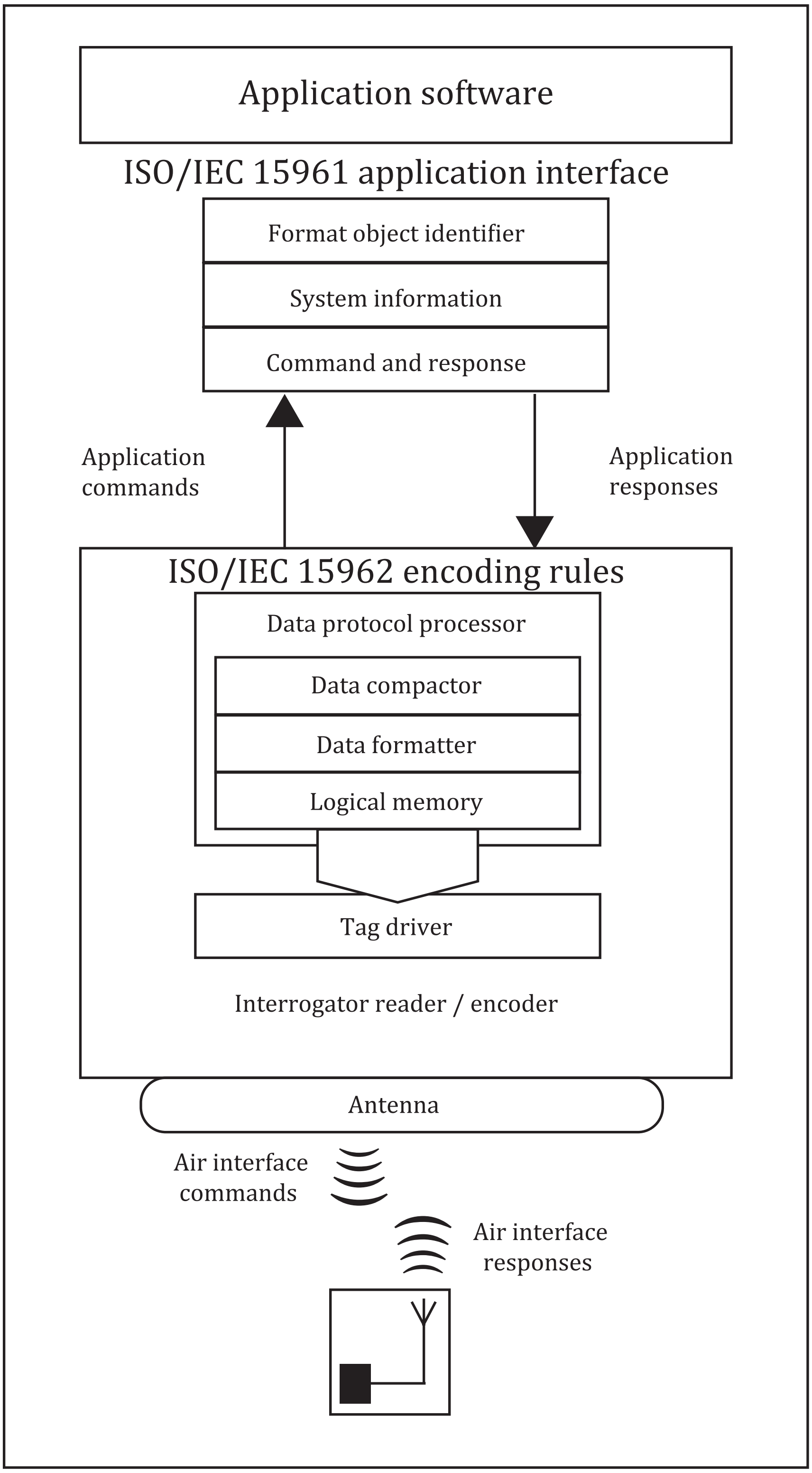

Figure 3 — ISO RFID data protocol architecture

7.4.2 Logical memory

The logical memory is the software equivalent of the structure of the memory on the RFID tag, itself. Not all tags have the same memory size and memory structure. The parameters that define block sizes and number of blocks shall be delivered from the tag through the interrogator and the tag driver to enable the encoder to create a logical memory that is appropriate for a particular tag.

This process is hidden from the application, but is necessary to cater for the fact that in a truly open system with full interoperability, RFID tags that are compliant with the specified air interface protocol (see 5.2.2) have different architectures.

7.4.3 Configuration of the RFID tag

7.4.3.1 General

ISO/IEC 15961-1 has specific commands that are used to configure the AFI and DSFID for a particular air interface protocol. The configuration of each of these components of the system information is defined in 7.4.3.2 to 7.4.3.3.

7.4.3.2 Configuration of the AFI

The ISO/IEC 15961-1 command to configure the AFI has an argument that enables the application to define that the AFI is to be locked, or to be left unlocked. As discussed in 7.2.2, the AFI shall not be locked if it is used as part of a security system where two values of the AFI are used, one for on-loan items and one for in-stock items. If other security mechanisms are used, then the AFI can be locked at the individual library's discretion. Once locked, the AFI cannot be unlocked.

7.4.3.3 Configuration of the DSFID

The DSFID, for library applications, consists of two components:

— the access method;

— the data format.

The data format is specified in 7.2.3, and the access method is defined in 7.2.6. These bit values are combined to create the appropriate DSFID byte value as shown in Table 4.

Table 4 — Relevant DSFID value

Bit sequence			DSFID byte
Access method[a]	Reserved	Data format	
00	0	00110	06

[a] 00 = No-directory, where the encoded bytes are concatenated in a continuous byte stream.

Some RFID tags do not have an explicit air interface command to write the DSFID to a designated memory location on the RFID tag. ISO/IEC 15962 uses rules in the tag driver to automatically identify whether a particular tag supports a “hard-coded” DSFID or a “software-encoded” DSFID.

NOTE This process is transparent to the application and uses similar features in ISO/IEC 15962 that are used to determine the size of the memory and the size of the block. Using these processes increases the opportunity for interoperability and choice of tags suitable for particular types of item.

7.4.4 Data compaction

Most of the data elements in Table 1 are subjected to the standard ISO/IEC 15962 compaction as described in the next paragraph. The exceptions are explicitly discussed later in this subclause.

When the command arguments are set to compact the data, ISO/IEC 15962 automatically selects the most efficient compaction scheme for each data element presented. This allows libraries to use alphanumeric or numeric code structures flexibly, with the only penalty being that more complex character sets require more encoding space on the RFID tag. It also enables shorter codes to be represented (generally) in fewer bytes.

The application-defined argument can be used to encode externally encrypted data, whose interpretation is only known to the host system. Its most common use in the support of this document is when encoding the OID index for Relative-OID value 2. Because this is a bit string, no pre-encoding is required. Another use of the application-defined argument is when encoding the ISIL for Relative-OID values 3 and 11. In this case, the ISIL shall be pre-encoded in accordance with the rules defined in Annex C, and then encoded according to ISO/IEC 15962 rules as application-defined. Similarly, the application-defined argument applies to the encoding of Relative-OID value 5 for type of usage, Relative-OID value 19 for media format (other), and Relative-OID value 20 for supply chain stage.

The UTF-8 string is used to encode characters outside the default character set of ISO/IEC 8859-1. This is mainly used for languages that use character sets other than the Latin No. One character set. This compaction scheme needs to be declared only when a UTF-8 character string is encoded for Relative-OIDs values 15, 16, 17 and 26.

The compaction schemes are identified on the RFID tag by a three-bit code which is included as part of the precursor (see 7.4.5.2). The full set of compaction schemes and their code is shown in Table 5.

Table 5 — ISO/IEC 15962 compaction schemes

Code	Name	Description
000	Application-defined	As presented by the application
001	Integer	Integer
010	Numeric	Numeric string (from “0” to “9”)
011	5-bit code	Uppercase alphabetic
100	6-bit code	Uppercase, numeric, etc.
101	7-bit code	US ASCII
110	Octet string	Unaltered 8-bit (default = ISO/IEC 8859-1)
111	UTF-8 string	External compaction to ISO/IEC 10646

7.4.5 Creating the encoded data set(s)

7.4.5.1 General

The encoding of the Relative-OID and data object on the RFID follows a particular sequential structure defined in ISO/IEC 15962. Subclauses 7.4.5.2 and 7.4.5.3 define the only rules, defined in ISO/IEC 15962 for the Relative-OID, that are relevant to this document.

NOTE ISO/IEC 15962 defines other rules, for example, for encoding full object identifiers, which are not described in this document because they are not relevant its application.

7.4.5.2 Data set for Relative-OID values 1 to 14

The structure of an encoded data set with the Relative-OID values 1 to 14 consists of the following components:

— a precursor, i.e. a single-byte that in this case encodes the compaction scheme and the Relative-OID;

— the length of the compacted data object;

— the compacted data object.

This structure is shown in Figure 4.

Figure 4 — ISO/IEC data set with Relative-OID values 1 to 14

The majority of data elements defined in this document have Relative-OID values (1 to 14). These are directly encoded in the precursor (see Table 6), and this reduces the amount of memory required for the encoding.

Table 6 — Bit position of precursor components

Precursor bit positions							
7	6	5	4	3	2	1	0
Offset	Compaction code			Object identifier			

The offset bit in the precursor is only set to “1” if an offset byte is encoded on the RFID tag. An example of the use of the offset byte is given in 7.4.5.4.

7.4.5.3 Data set for OID values 15 to 127

The precursor only provides 4 bits for encoding the object identifier. It is only capable of directly encoding Relative-OID values from 1, which encodes as 0001_2, to 14 which encodes as 1110_2. For Relative-OID values between 15 and 127, of which some are used for this document, the last four bits of the precursor are set = 1111_2. This signals that the Relative-OID shall be explicitly encoded as a separate component (a single-byte) in the data set, as shown in Figure 5.

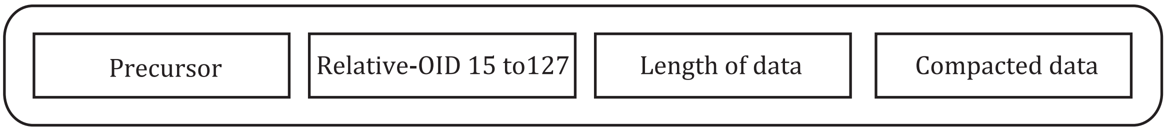

Figure 5 — ISO/IEC data set with Relative-OID values 15 to 127

The value that is encoded for the Relative-OID is the value offset by −15. This means that Relative-OID 15 is encoded as 15 − 15 = 0 = 00_{HEX}. The highest Relative-OID that can be encoded in this manner is Relative-OID 127, encoded as 127 − 15 = 112 = 70_{HEX}.

7.4.5.4 Locking a data set

Based on the requirements of the application, any one or more data elements can be locked. The lock object argument in an application command effectively calls for the entire data set to be locked. This avoids one component being permanently encoded and others changeable. The ISO/IEC 18000-3 Mode 1 air interface protocol permits locking by block. Generally, any data set that requires locking needs to be block aligned so that the precursor is the first byte of a block and the precursor of the next data set also begins on the first byte of a block. The encoding rules carry out the necessary re-alignment processes, which are defined in detail in this subclause, and insert an offset byte immediately following the precursor. The value of this offset is the number of null bytes (typically encoded as value 00_{HEX}, but the value 80_{HEX} is also acceptable) that are added after the final byte of the compacted data to end on a block boundary.

Each of the values of the pad byte, or null bytes, has the same status. The value encoded in the offset byte determines how many bytes to skip after the compacted data to find the precursor of the next encoded data set. The null byte value 80_{HEX} is preferred when modifying or deleting data to minimize the number of air interface transactions.

The decode process is expected to accept tags encoded with one or other pad byte value, and even with these intermixed on the same tag.

Although all of this processing is undertaken automatically by the software that implements the ISO/IEC 15962 encoding rules, the following descriptions are included for users to gain a better understanding of some of the factors that are taken into account.

— If the data set immediately preceding the data set to be locked is itself to remain unlocked, the encoding rules ensure that this data set ends on a block boundary. This is to ensure that the locking process does not lock the trailing bytes of the unlocked data set. This formatting process can result in the insertion of an offset byte and a change in the value of the precursor.

— If two or more contiguous data sets are to be locked, then block alignment is only required at the beginning of the first locked data set and the end of the last locked data set. From this, it should be clear that there can be encoding efficiencies and a reduction in the number of bytes to be encoded if data sets that are to be locked are grouped together.

— Once a block of memory is locked, it cannot be unlocked or erased, so the data set becomes permanently encoded on the RFID tag.

7.4.5.5 Logical memory

Irrespective of whether one data set or multiple data sets are to be encoded, or if a data set is to be added or modified, the encoded bytes are formatted in the logical memory in a structure that is compliant with the specific tag architecture. Because the block size and number of blocks differ between manufacturers and even between model versions, this formatting is an essential feature of the encoding rules to achieve interoperable RFID tags. This enables any RFID tags to be considered as candidates for encoding to this document that can claim compliance with ISO/IEC 18000-3 Mode 1, but differ between each other within the options permitted by the air interface standard.

EXAMPLE 1 RFID tags can have different sizes of memory.

EXAMPLE 2 The block size is permitted to vary within a prescribed range.

EXAMPLE 3 Some tags are able to transfer multiple blocks across the air interface in write and read transactions, others only to transfer single blocks.

Once the logical memory has been populated, single or multiple blocks are written across the air interface. Any blocks that require to be locked are marked up as such so that the interrogator calls up a subsequent series of lock block air interface commands.

When reading data from the RFID tag, the logical memory is populated block by block. Decoding of an RFID tag with a No-directory access method is done sequentially by Relative-OID, but the data object only needs decoding if its Relative-OID is selected by the application command.

There is an ISO/IEC 15961-1 command that enables the first position data set(s) to be read without attempting to read other data from the RFID tag. This command should be used to achieve a faster read transaction of the data sets that are encoded in the lower block positions. A use of this command is to read the primary item identifier and the OID index as part of an initial transaction.

8 RFID tag requirements

8.1 Air interface protocol

8.1.1 General

The air interface protocol shall be in accordance with ISO/IEC 18000-3 Mode 1, with the requirements specified in the 8.1.2 to 8.1.5.

8.1.2 Declaring memory parameters

The tag ID (which specifies a unique tag identifier) is a mandatory component for ISO/IEC 18000-3 Mode 1 RFID tags. The structure of the 64-bit code, as defined in ISO/IEC 15693-3 only specifies the first 16 bits of this code structure. ISO/IEC 18000-3 Mode 1 RFID tags that claim compliance with this document shall provide information to enable interrogators and applications to establish the following characteristics of the RFID tag:

— the block size;

— the number of blocks;

— the read size if greater than one block;

— the write size if greater than one block;

— the address of the first block in which application data can be written;

— the address of the last block in which application data can be written.

8.1.3 AFI memory

An ISO/IEC 18000-3 Mode 1 RFID tag compliant with this document shall have a specific memory location assigned to encode the AFI. This location shall be addressable with write, read, and lock commands. The address of this location can be determined by the IC manufacturer and does not need to be declared in air interface commands.

8.1.4 DSFID memory

An ISO/IEC 18000-3 Mode 1 RFID tag compliant with this document shall support the DSFID in one of the following two ways.

— The preferred manner is for the RFID tag to have a specific memory location assigned to encode the DSFID. This location shall be addressable with specific DSFID write, read, and lock commands. The address of this location can be determined by the IC manufacturer and does not need to be declared in air interface commands.

— The alternative method is to use the “software-encoded” rules for DSFID specified in ISO/IEC 15962.

RFID tags using either method are fully interoperable with respect to the use of the DSFID.

If the RFID tag has a specific memory allocation, then it could also support a *Lock DSFID* air interface command. Locking then becomes a user choice based on an assessment of the need to change the access method and data format at some future date. If, on the other hand, the DSFID is software-encoded, locking it should be taken into consideration with locking the primary item identifier, which this document recommends to be locked.

8.1.5 Required air interface commands

Table 7 identifies the mandatory and optional commands that are requirements for RFID for item management applications and therefore for this document. Interrogators and tags claiming conformance to this document shall comply with the item management requirements provided in Table 7.

Table 7 — Required commands and their codes

Command code	ISO/IEC 18000-3 Mode 1 basic type	Function	Item management requirement
01	Mandatory	Inventory	The AFI is a requirement in the command, and the DSFID is required as part of the response.
02	Mandatory	Stay quiet	No change.
20	Optional	Read single block	The interrogator shall support this command. The RFID tag shall support this command if the *Read multiple blocks* command is not supported.
21	Optional	Write single block	The interrogator shall support this command. The RFID tag shall support this command if the *Write multiple blocks* command is not supported.
22	Optional	Lock block	Required for the interrogator and for the RFID tag.
23	Optional	Read multiple blocks	The interrogator shall support this command. The RFID tag shall support this command if the *Read single block* command is not supported.
24	Optional	Write multiple blocks	The interrogator shall support this command. The RFID tag shall support this command if the *Write single block* command is not supported.
25	Optional	Select	This command shall be supported in interrogators and should be supported in tags.
26	Optional	Reset to ready	This command shall be supported in interrogators and should be supported in tags.
27	Optional	Write AFI	Required for the interrogator and for the RFID tag.
28	Optional	Lock AFI	Required for the interrogator and for the RFID tag.
29	Optional	Write DSFID	The interrogator shall support this command. The RFID tag should support this command, but if this is not possible, it shall support the “software-encoded” DSFID (see 8.1.4)
2A	Optional	Lock DSFID	The interrogator shall support this command. The RFID tag should support this command if it supports the air interface command *Write DSFID* (command code 29).
2B	Optional	Get system information	Required for the interrogator and for the RFID tag.
2C	Optional	Get multiple block security status	Required for the interrogator and for the RFID tag.

8.2 Bit and byte sequence

Bit and byte ordering rules vary based on the standards being referred to, the development of proprietary interfaces to the interrogator, the interrogator's processing of data, even the brand and model of computer and operating system used. Figure 6 illustrates some of the issues and constraints.

ISO/IEC 18000-3 Mode 1 refers to the air interface protocol defined in ISO/IEC 15693-3. This makes clear that the AFI and DSFID (as single-byte values) are transmitted in commands and responses least significant bit first. The unique chip id, or UID, is a multiple-byte field that is transmitted least significant byte first; each byte is transmitted least significant bit first.

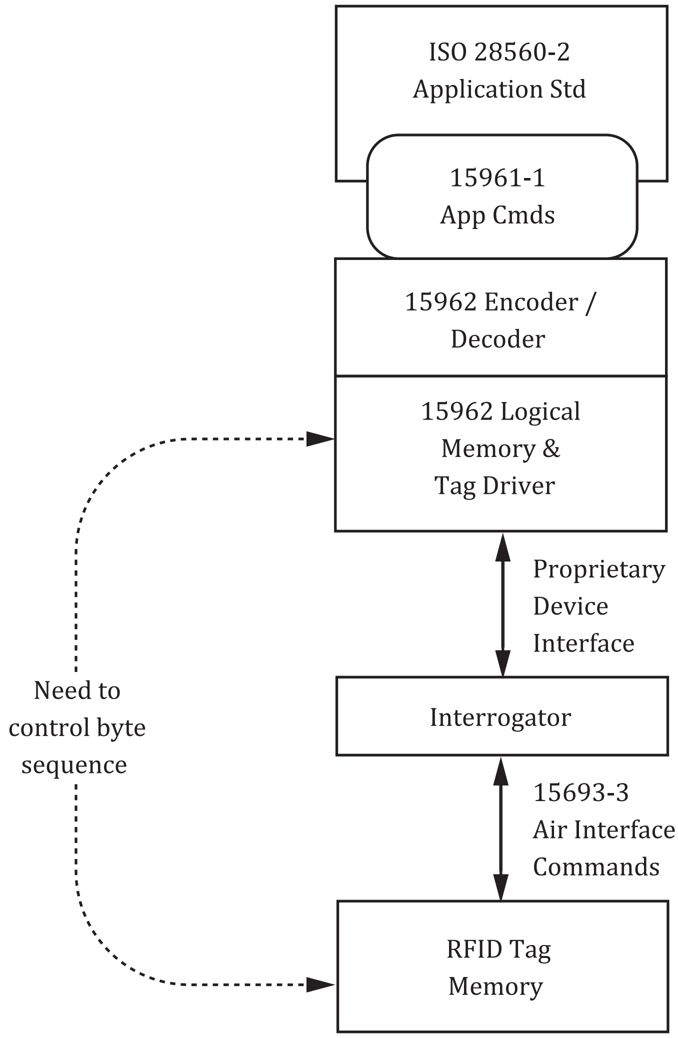

Figure 6 — Issues and constraints on bit and byte sequence

Unlike some of the more recent RFID technologies, ISO/IEC 18000-3 Mode 1 has no standardized device interface. This means that the bit and byte ordering might be addressed differently in the interrogator, or input API to the interrogator, or even between the ISO/IEC 15962 logical memory and the API for the interrogator. It is therefore important for systems designers to address these issues for the particular software, hardware, and operating system being used.

ISO/IEC 15962 defines the structure of a data set as always beginning with the precursor.

The complete encoding for an example of the primary item identifier is shown in Table D.3.

The precursor of the primary item identifier is therefore encoded in the lowest addressable block of user memory and in the lowest addressable byte. For example, in a memory with a 4-byte block, the encoding for the example in Table D.3 is shown in Figure 7.

Increasing block addresses ▼▼▼▼	Increasing byte addresses ▶▶▶▶			
	91	00	05	1C
	BE	99	1A	14

Figure 7 — Encoded bytes for example in Table D.3

The only exception to this is for those tags where the soft-coded DSFID needs to be in this position; moving the precursor of the primary item identifier to the second addressable byte.

The air interface Write Single Block command (or request) to write to the lowest block is shown in Table 8, with each byte with each byte sent to the RFID tag least significant bit first.

Table 8 — Example of a Write Single Block ISO/IEC 15693-3 air interface command

SOF	Flags	Command code	UID[a]	Block Number[b]	Data	CRC16	EOF
	8 bits	21	D5 9B 7A 13 00 01 04 E0	8 bits	91 00 05 1C	16 bits	

[a] The UID is only required under specific flag settings. The UID of the tag in this example is E0040100137A9BD5.

[b] The value of the lowest block number varies depending on the memory architecture defined by different IC manufacturers and models.

8.3 Air interface conformance

Conformance testing of the system shall be in accordance with ISO/IEC 18047-3.

8.4 Performance

Performance testing of the RFID tag shall be in accordance with ISO/IEC 18046-3.

9 Data integrity, security and privacy issues

9.1 Data integrity

ISO/IEC 15962 supports the selective locking of an individual data set, which renders the associated blocks on the RFID tag permanently locked and virtually impossible to change. This feature should be used to lock particular data objects that would render the RFID tag inoperable if changed. Any data set that is likely to be modified, or deleted, should not be locked. The locking feature should be considered for the primary item identifier (see 6.2) and possibly the ISIL code for the owner institution (see 6.4). Locking the data sets of other data elements is a local decision.

Locking any data element ensures the permanent integrity of that data throughout the lifetime of the loan item, and protects the system against accidental or deliberate changes of key data elements.

9.2 Item security

9.2.1 General

Various approaches can be used for securing library loan items against unauthorized removal. The choice of a security system is outside the scope of this document and the responsibility of solution providers to develop particular schemes for libraries to choose. However, there are some specific features of the ISO/IEC 18000-3 Mode 1 RFID tag and the implementation of the data protocol that can be incorporated into specific security systems. The individual features are discussed in the following subclauses, without any comment on their particular merits. Combinations of these can also be provided in particular systems.

Detailed advice is provided in ISO 28560-1.

9.2.2 Use of the dual AFI system

Security implementations based on a dual AFI require that value 07_{HEX} is programmed as the AFI code for library items that are checked into the collection. It also requires that AFI code $C2_{HEX}$ is encoded on items that are checked out and on loan.

The portal at the library exit interrogates for any tag with AFI code 07_{HEX}. RFID tags with this AFI code value respond with their unique tag ID, while all other tags with different AFI codes (including the authorized on-loan code value of $C2_{HEX}$) are ignored.

If this security system is used, the AFI code shall not be locked.

9.2.3 Use of the unique tag ID

Each ISO/IEC 18000-3 Mode 1 tag has a unique tag ID programmed by the integrated circuit manufacturer in a non-volatile memory. The tag ID is used as part of the anti-collision procedure to ensure that communications across the air interface is with one particular RFID tag. As such, it is the primary stage of the communication chain and is part of the response of the Inventory command. The Inventory command, itself, has an argument that enables selection using the AFI.

The database lookup system requires a database to be compiled of the tag IDs of items that are checked out of the library, usually for a given defined period of time. The security system is programmed to read all tag IDs, look them up on the database and, if they are not on the database of authorized checked-out items, the assumption is that they have been removed in an unauthorized manner.

Because the ISO/IEC 18000-3 Mode 1 tag can be used in other systems, the DSFID (see 7.2.6) should be checked to ensure that the RFID tag is compliant with this document. The DSFID is returned as part of the response to the Inventory command.

It is essential that the AFI on-loan items conform with this document to avoid interference with other RFID systems that are compliant with the AFI registration procedures of ISO/IEC 15961-2. Security implementations that are based on the unique tag ID being checked for “on loan” status shall additionally ensure that the AFI value $C2_{HEX}$ is encoded on all items that are checked out and on loan. The procedure to achieve this is not specified here and should be developed by each systems manufacturer.

9.2.4 Use of the EAS features

Electronic article surveillance (EAS) features have been added by some manufacturers to the ISO/IEC 18000-3 Mode 1 tag as a proprietary feature. As such, the operation of this feature is outside the scope of the air interface standard. It is included here, because a number of vendors offer this as a feature as part of their system. Interoperability between different EAS systems cannot be assumed.

It is essential that the AFI on-loan items comply with this document to avoid interference with other RFID systems that are compliant with the AFI registration procedures of ISO/IEC 15961-2. Security implementations that are based on EAS features being checked for “on loan” status shall additionally ensure that the AFI value $C2_{HEX}$ is encoded on all items that are checked out and on loan. The procedure to achieve this is not specified here and should be developed by each systems manufacturer.

10 Implementation and migration

Issues concerned with new implementations and migration issues from legacy RFID implementations to this document are discussed in Annex E.

Annex A
(informative)

Information about ISO 28560 RFID in libraries

A.1 Informational website

Additional information about RFID in libraries is available from http://biblstandard.dk/rfid/, hosted by the Royal Danish Library.

A.2 Types of support information

At the time of publication of this document, two relevant items are available. These are listed below.

— RFID in libraries. Links to external materials:

http://biblstandard.dk/rfid/docs/RFID-in-libraries-Links-external

— RFID in libraries. Q&A:

http://biblstandard.dk/rfid/docs/RFID-in-libraries-q-and-a

Other material can be published in future, and this will be publicized and made available from the URL in A.1.

Annex B
(normative)

Relevant ISO/IEC 15961-1 application commands

B.1 Configure-AFI

The Configure-AFI command is used to write, or overwrite, the AFI code to the RFID tag. The command supports the additional argument to lock the AFI.

A valid response from the command indicates that the command was successfully implemented, or provides an indication of failure.

B.2 Configure-DSFID

This command supports the writing of the DSFID to the RFID tag. It includes support for the optional argument to lock the DSFID. The ISO/IEC 15962 processes automatically determine whether the particular RFID tag being encoded supports the "hard-coded" version of DSFID or the "software-encoded" version.

The response indicates whether the operation was successful, or provides details of errors.

B.3 Inventory-Tags

The Inventory-Tags command is used to select a set of tags from within a population of RFID tags. The AFI is used as a selection criterion. It is also possible to use the Identify-Method argument to determine whether some, an exact number of, or all tags need to be identified. This argument is combined with the Number-Of-Tags argument. The combination of these arguments is intended to support many different business applications, where reading all the RFID tags might not be necessary.

The response from the command is a list comprising the tag IDs together with the DSFID for each RFID tag. There is also an indication of the nature of any failures to complete the command instructions.

B.4 Write-Objects

The Write-Objects command and its Add-Objects-List argument is used to write a list of one or more object identifiers and objects to an RFID tag. Each object is provided as a byte stream supported by a Compact-Parameter. Additionally, there are arguments to lock the data set and to check that the Relative-OID is not already encoded on the RFID tag.

The response from the command is a list that identifies what action was taken with respect to each data object.

B.5 Read-Objects

The Read-Objects command is used to read one or more data objects from the RFID tag. It is supported by a Read-Type argument that enables only first objects (such as the primary item identifier and OID-index) to be read, or to read one or more objects, or to read all objects. If the read type is Read-First-Objects, this has to be supported by an additional argument, i.e. Max-App-Length, which requires a total number of bytes to be defined. This value can be calculated relatively easily if the primary item identifier for the particular library is of a fixed length. An additional Check-Duplicates argument can also return information of whether there is more than one instance of a particular Relative-OID encoded on the RFID tag. This can be useful for housekeeping purposes.

The response from the command provides a list of information associated with each data object. In particular, it clearly identifies whether the data has been de-compacted, or remains application-defined, as would apply for example to the OID-index.

B.6 Read-Object-Identifiers

The Read-Object-Identifiers command reads all the object identifiers, but not the associated data objects, from an RFID tag. The command is mainly used as part of a “housekeeping” procedure, for example to check that the OID-Index is correctly encoded, or when a new loan item is brought in from another source such as an interlibrary loan.

The response from the command provides a list of all of the Relative-OIDs encoded on the tag, or details of any failure to execute the command.

B.7 Get-App-Based-System-Info

The Get-App-Based-System-Info command requires the interrogator to read the system information (the AFI and DSFID) from the RFID tag. In some applications, this is a housekeeping command ensuring that the code values are correctly encoded.

The response from a successful read returns the AFI and DSFID values.

B.8 Modify-Object

The Modify-Object command is used to change the value of the data object, and effectively overwrite the associated data set. If the data set is already locked, it cannot be modified. The command supports an argument that enables the modified data set to be locked.

The response from the command either indicates success, or reasons for failure in invoking the command including the inability to modify a data set that is already locked.

B.9 Delete-Object

The Delete-Object command enables a complete data set to be deleted from the RFID tag. This can only be achieved if the data set is not locked.

The response from the command indicates whether the action was successful, or provides reasons for failure.

B.10 Read-Logical-Memory-Map

The Read-Logical-Memory-Map command is used for diagnostic purposes. It returns the complete content of the memory on the tag in terms of encoded bytes, with no processing to identify individual object identifiers or objects.

The response is either a byte stream of all the encoding on the tag, or a reason for failing to action the command.

B.11 Erase-Memory

The Erase-Memory command instructs the interrogator to set to zero the entire encoding on a specified RFID tag. If any blocks are locked, then a generic Blocks-Locked response is returned, indicating failure to properly action the command.

The response from the command gives an indication that the command was successfully completed, or reasons for failure.

Annex C
(normative)

Pre-encoding the ISIL

C.1 General considerations

The ISIL supports a number of structures, including library identifiers under a national scheme using the two-alphabetic-character ISO 3166-1 country code, and three other formats where the prefix identifiers are one-, three-, or four-character long. A hyphen minus character separates this code from the library identifier.

The hyphen minus character is essential to distinguish the boundary point between the various types of prefix and the library identifier. The use of the standard compaction schemes of ISO/IEC 15962 results in a relatively inefficient compaction of these data.

To achieve a more efficient compaction of the ISIL, which can be up to 16 characters in length and a mixture of uppercase, lowercase, and numeric characters, a specific pre-encoding scheme is included in this document. The scheme supports the encoding of any mixture of characters that is compliant with ISO 15511 for the registration of the ISIL.

The ISIL encoding scheme is based on an encoding table, as defined in Table C.1. The characters are laid out in three columns, each of which contains a subset of the character set and some control characters. Each alphabetic and punctuation character is encoded in 5 bits, and each numeric character in 4 bits. Special control characters (see C.2) are used to switch between the sets. The encoding shall always begin in the uppercase set, a point that is explained in C.3.

Table C.1 — ISIL encoding table for this document

Uppercase set			Lowercase set			Numeric set		
Value	**Character**	**Hex**	**Value**	**Character**	**Hex**	**Value**	**Character**	**Hex**
00000	-	2D	00000	-	2D	0000	0	30
00001	A	41	00001	a	61	0001	1	31
00010	B	42	00010	b	62	0010	2	32
00011	C	43	00011	c	63	0011	3	33
00100	D	44	00100	d	64	0100	4	34
00101	E	45	00101	e	65	0101	5	35
00110	F	46	00110	f	66	0110	6	36
00111	G	47	00111	g	67	0111	7	37
01000	H	48	01000	h	68	1000	8	38
01001	I	49	01001	i	69	1001	9	39
01010	J	4A	01010	j	6A	1010	-	2D
01011	K	4B	01011	k	6B	1011	:	3A
01100	L	4C	01100	l	6C	1100	Latch upper	N/A
01101	M	4D	01101	m	6D	1101	Shift upper	N/A
01110	N	4E	01110	n	6E	1110	Latch lower	N/A
01111	O	4F	01111	o	6F	1111	Shift lower	N/A
10000	P	50	10000	p	70			
10001	Q	51	10001	q	71			

Table C.1 *(continued)*

Uppercase set			Lowercase set			Numeric set		
Value	**Character**	**Hex**	**Value**	**Character**	**Hex**	**Value**	**Character**	**Hex**
10010	R	52	10010	r	72			
10011	S	53	10011	s	73			
10100	T	54	10100	t	74			
10101	U	55	10101	u	75			
10110	V	56	10110	v	76			
10111	W	57	10111	w	77			
11000	X	58	11000	x	78			
11001	Y	59	11001	y	79			
11010	Z	5A	11010	z	7A			
11011	:	3A	11011	/	2F			
11100	Latch lower	N/A	11100	Latch upper	N/A			
11101	Shift lower	N/A	11101	Shift upper	N/A			
11110	Latch numeric	N/A	11110	Latch numeric	N/A			
11111	Shift numeric	N/A	11111	Shift numeric	N/A			

C.2 Control characters

Each character set in Table C.1 has four control characters. These are used as follows.

— The *Shift* function transfers the encoding to a different character set for a single character. The encoding automatically returns to the character set from which the *Shift* was invoked.

— The *Latch* function transfers subsequent encoding to one of the other character sets, where encoding remains in place until the end of encoding or until another *Latch* or *Shift* is invoked.

These control characters are required to be encoded on the RFID tag so that the data can be properly reconstructed during the decode process.

C.3 Encoding rules

C.3.1 Base character set

The encoding shall begin in the uppercase character set, which is considered as the base character set. This is to provide an efficient encoding scheme for the majority of ISIL codes that begin with an alphabetic country prefix. If the ISIL begins with a lowercase alphabetic character or a numeric digit, then the appropriate latch or shift character from the uppercase set shall be the first encoded bit string.

C.3.2 Encoding process

Each character is encoded in sequence, adding a bit pattern associated with a particular character to an encoded bit string. Encoding remains within the same character set until the end of the encoding process, or until a character is found that is not supported by the particular character set.

At this point, there is an advantage to “look ahead” to additional characters. If the next two characters can be supported in the same character set, then a latch to that character set should be used. If only one character can be encoded from the character set, then a shift to that character set should be used.

These rules are recommended because they are simple to implement. Using alternative analyses of the complete character string might result in a more efficient encoding, and this can be used as long as appropriate shift and latch characters are properly used.

A shift or a latch to the numeric set only requires each of these characters to be recorded with a four-bit string.

At the end of the encoding process, a concatenated bit string is produced that consists of a number of five-bit and four-bit (for any numeric digits) patterns produced for each character and any necessary control characters. If the number of bits is not perfectly divisible by eight, then additional pad bits “1” shall be added at the end. During the decode process, some of these pad bits might appear as control characters, but as they cannot point to a shift in alphabetic, numeric, or punctuation characters, they are ignored.

C.4 Declaring the ISO/IEC 15962 compaction scheme

When the byte stream is transferred to the ISO/IEC 15962 encoding process, the compaction scheme shall be specified as “application-defined”. This is to ensure that no additional compaction is applied. An application-defined compaction scheme declares itself as such during the decode process, requiring special application-defined rules (for example, as defined in this annex), to be applied to interpret the data string.

C.5 Use of generic or application-specific ISO/IEC 15962 encoders and decoders

C.5.1 General

It is likely that a library implementation could use a generic ISO/IEC 15962 encoder or decoder, or one more specifically designed to support this document. It is also possible that, depending on specific equipment configurations, the library implementation might even need to support both methods. These are discussed below.

C.5.2 Generic ISO/IEC 15962 encoder or decoder

The generic ISO/IEC 15962 encoder or decoder is intended for use in mass-market applications where application-specific details are not directly supported. Therefore, the encoding rules, and implicit decoding rules, defined in this Annex for the ISIL need to be implemented externally. The presentation to the ISO/IEC 15962 encoder is the byte stream result (see C.3.2), qualified by the compaction scheme being declared as application-defined. The generic ISO/IEC 15962 decoder outputs the uninterpreted byte string, qualified by the compaction scheme being declared as application-defined.

C.5.3 Inclusive encoder or decoder for this document

In this type of hardware or software device implementation, the interface with the application is the ISIL, itself. All of the encoding processes described in this annex are incorporated in the process, including the creation of the appropriate byte stream and declaring that the compaction is application-defined. The decoding process outputs a valid ISIL.

C.6 Encoding examples

C.6.1 EXAMPLE 1: DE-Heu1

This example contains characters from all three character sets and illustrates the use of the latch and shift control characters as follows.

a) Encoding begins in uppercase character set with the character {D}, which encodes as “00100”.

b) Encoding continues with the next three characters {E-H}, which are from the same character set and encode as: “00101”, “00000”, and “01000”.

c) The next two characters are lowercase {eu}, and cannot be encoded in the same character set as used for the first four characters. Therefore, a latch lower control character needs to be encoded as “11100”.

d) The two lowercase characters {eu} are then encoded as “00101” and “10101”.

e) The next character is from the numeric set and cannot be encoded in the same character set as currently being encoded. As there is only a single digit, a shift numeric control character is encoded as “11111”.

f) The numeric digit {1} is encoded as the four-bit code “0001”.

g) The bit pattern is concatenated to create a 44-bit string, as shown in Table C.2.

Table C.2 — Encoding of ISIL DE-Heu1

D	E	-	H	Latch lower	e	u	Shift numeric	1
00100	00101	00000	01000	11100	00101	10101	11111	0001

h) The four-bit pad string “1111” is appended to round this encoding string to one divisible by eight to enable conversion to a byte string, as shown in Table C.3.

Table C.3 — Byte string for ISIL DE-Heu1

00100001	01000000	10001110	00010110	10111111	00011111
21	40	8E	16	BF	1F

C.6.2 EXAMPLE 2: CH-000134-1

This example is an ISIL with a long numeric string and is encoded as follows.

a) Encoding begins in uppercase character set with the character {C}, which encodes as “00011”.

b) Encoding continues with the next two characters {H-}, which are from the same character set and encode as “01000” and “00000”.

c) The next six characters are numeric {000134}, and cannot be encoded in the same character set as used for the first three characters. Therefore, a latch numeric control character needs to be encoded as “11110”.

d) The six numeric digits {000134} are then encoded as a sequence of four-bit codes: “0000”, “0000”, “0000”, “0001”, “0011”, and “0100”.

e) The next character is {-}, and as this is supported by the numeric set, it can be encoded as a four-bit code “1010”.

f) The final character is the numeric digit {1} and is encoded as “0001”.

g) The bit pattern is concatenated to create a 52-bit string, as shown in Table C.4.

Table C.4 — Encoding of ISIL CH-000134–1

C	H	-	Latch numeric	0	0	0	1	3	4	-	1
00011	01000	00000	11110	0000	0000	0000	0001	0011	0100	1010	0001

h) The four-bit pad string “1111” is appended to round this encoding string to one divisible by eight to enable conversion to a byte string, as shown in Table C.5.

Table C.5 — Byte string for ISIL CH-000134–1

00011010	00000001	11100000	00000000	00010011	01001010	00011111
1A	01	E0	00	13	4A	1F

Annex D
(informative)

Encoding examples

D.1 General considerations

This annex shows the encoding of a hypothetical set of data elements compliant with this document. The example includes features that require selective locking of some data elements.

The processes described in this annex are presented in a manner to help the reader understand how input data are converted to encoded bytes on the RFID tag. This is done progressively for each data element, but it should be borne in mind that software compliant with ISO/IEC 15962 might have different processes to achieve the same end result.

D.2 Input assumptions

D.2.1 RFID tag

The RFID tag has separate memory areas for encoding the AFI and the DSFID. Therefore, the encoding of the data can begin from the first byte of user memory. The user memory is organized in blocks of 4 bytes and each block can be independently locked.

D.2.2 Input data

The data elements to be encoded are described in Table D.1.

Table D.1 — Example data elements

Data element	Sequence	Relative -OID	Lock	Format	Example data
Primary item ID	1st	1	Yes	Variable length alphanumeric	123456789012
OID index	2nd	2	No	Bit mapped code	
Set information	3rd	4	No	*n* of *m* structure (*n* and *m* numeric ≤ 255)	3rd item of set of 12
Shelf location	4th	6	No	Variable length alphanumeric	QA268.L55
Owner institution (ISIL)	5th	3	Yes	Variable length field based on ISO 15511	US-InU-Mu

D.3 Encoding the data elements

D.3.1 General

As all of the Relative-OID values are in the range 1 to 14, the Relative-OID is directly encoded in the precursor (see 7.4.5.2). This means that each data set consists of the precursor, the length of the encoded data, and the compacted data.

D.3.2 Primary item identifier

The primary item identifier is all-numeric, and because it begins with a non-zero value the compaction process automatically compacts this as an integer. The encoded byte string is:

1C BE 99 1A 14

It is encoded in 5 bytes, and the compaction code (see Table 5) is “001”. The complete data set requires this to be preceded by a byte representing the length of the object, and that to be preceded by the precursor. This makes a total of 7 bytes, but as this data set needs to be locked, it needs to be extended by 1 byte so that it can be encoded on a 4-byte boundary.

The precursor (see 7.4.5.2) consists of three component parts: the offset bit, the compaction code, and the Relative-OID. The structure of the precursor of this example is shown in Table D.2.

Table D.2 — Precursor for the primary item identifier example

Description	Precursor bit positions							
	7	6	5	4	3	2	1	0
Function	Offset	Compaction code			Object identifier			
Example	1	001			0001			

The precursor is encoded as the hexadecimal value 91. Because the offset bit is set, an offset byte needs to be inserted immediately following the precursor. The offset byte indicates the number of pad bytes that need to follow at the end of the data set. In this particular case, the offset byte itself provides the necessary padding (a 1 byte) and therefore the offset byte value 00_{HEX}.

The complete encoding of this data set is shown in Table D.3.

Table D.3 — Encoded byte string for the primary item identifier

Precursor	Offset	Length of compacted data	Compacted data
10010001 = 91	00	05	1C BE 99 1A 14

D.3.3 OID index

This index (see 6.3) is a bit map and identifies the Relative-OID values that are encoded. As the Relative-OID for the primary item identifier is mandatory and Relative-OID 2 is used for this data element, counting can begin from 3. The other Relative-OID values encoded in this example on the RFID tag are 3, 4, and 6. A “1” bit is set to indicate the presence of a Relative-OID, so that pattern for this example is:

1101

This is padded with trailing zeros to achieve byte alignment, with the resultant bit string:

11010000

This converts to hexadecimal as “D0”.

This is encoded in 1 byte, and compaction code is “000”.

As this data set is not locked, and nor is the next data set, there is no need for block alignment. The complete encoding of this data set is shown in Table D.4.

Table D.4 — Encoded byte string adding the OID index

Precursor	Offset	Length of compacted data	Compacted data
10010001 = 91	00	05	1C BE 99 1A 14
00000010 = 02		01	D0

D.3.4 Set information

In this example, this is item 3 of a set of 12. The encoding rules require the size of a set to be followed by the ordinal value of the item, so the input value is “1203”. This is an all-numeric code, and it is encoded as an integer value. The encoded byte string is:

04 B3

This is encoded in 2 bytes, and compaction code is “001”.

As this data set is not locked, and nor is the next data set, there is no need for block alignment. The complete encoding of this data set is shown in Table D.5.

Table D.5 — Encoded byte string adding the set information

Precursor	Offset	Length of compacted data	Compacted data
10010001 = 91	00	05	1C BE 99 1A 14
00000010 = 02		01	D0
00010100 = 14		02	04 B3

D.3.5 Shelf location

The example uses a Library of Congress Catalogue classification, with the value QA268.L55. To encode all the nine characters including the full stop (or dot) {.}, the six-bit code compaction scheme is automatically selected by the encoding software. The encoded byte string is:

44 1C B6 E2 E3 35 D6

This is encoded in seven bytes, and compaction code is “100”, indicating the six-bit compaction scheme.

As this data set is not locked, and ends on block boundary, there is no need for block alignment. The complete encoding of this data set is shown in Table D.6.

NOTE If the data set did not end on a block boundary, then an offset and necessary pad bytes would need to be encoded, so that the next data set can be properly locked.

Table D.6 — Encoded byte string adding the shelf location

Precursor	Offset	Length of compacted data	Compacted data
10010001 = 91	00	05	1C BE 99 1A 14
00000010 = 02		01	D0
00010100 = 14		02	04 B3
01000110 = 46		07	44 1C B6 E2 E3 35 D6

D.3.6 Owner institution (ISIL)

This example is alphanumeric, with hyphens:

US-InU-Mu

These are encoded using the scheme defined in Annex C. The encoding is shown in Table D.7.

Table D.7 — Encoding of ISIL US-InU-Mu

U	S	-	I	Shift lower	n	U	-	M	Shift lower	u
10101	10011	00000	01001	11101	01110	10101	00000	01101	11101	10101

This results in a 55-bit string, and a single-bit 1 is appended to round this encoding string to one divisible by eight to enable conversion to a byte string as shown in Table D.8.

Table D.8 — Byte string for ISIL US-InU-Mu

10101100	11000000	10011110	10111010	10100000	01101111	01101011
AC	C0	9E	BA	A0	6F	6B

This is encoded in 7 bytes, and compaction code is “000”, indicating the application-defined compaction scheme.

These 7 bytes need to be preceded by a precursor and length byte. This results in a total length of 9 bytes, but as this data set is required to be locked, the data set needs to be rounded up to 12 bytes to achieve block alignment. Therefore, the precursor needs to indicate the presence of an offset byte. In turn, the offset byte on its own is insufficient to achieve block alignment. Therefore, the offset byte needs to be encoded with the value 02 to indicate that there are two trailing pad bytes following the data set. The complete encoding of this data set is shown in Table D.9.

NOTE Even though a pad byte is required, the length of the compacted data remains unchanged. This is because the number of pad bytes is declared by the value of the offset.

Table D.9 — Encoded byte string adding the ISIL

Precursor	Offset	Length of compacted data	Compacted data	Pad bytes
10010001 = 91	00	05	1C BE 99 1A 14	
00000010 = 02		01	D0	
00010100 = 14		02	04 B3	
01000110 = 46		07	44 1C B6 E2 E3 35 D6	
10000011 = 83	02	07	AC C0 9E BA A0 6F 6B	00 00

D.4 Complete encoding

For illustration purposes, the encoding is set out in a tabular format, as shown in Tables D.10 and D.11.

Table D.10 — Encoded bytes

Block 1	**91**	**00**	**05**	**1C**	Locked
Block 2	**BE**	**99**	**1A**	**14**	Locked
Block 3	**02**	**01**	**D0**	**14**	
Block 4	**02**	**04**	**B3**	**46**	
Block 5	**07**	**44**	**1C**	**B6**	
Block 6	**E2**	**E3**	**35**	**D6**	
Block 7	**83**	**02**	**07**	**AC**	Locked
Block 8	**C0**	**9E**	**BA**	**A0**	Locked
Block 9	**6F**	**6B**	**00**	**00**	Locked

Table D.11 — Encoded bytes of the resulting data element

First 3 bytes	Data element
91 00 05 …	Primary item identifier data set
02 01 D0	OID index data set
14 02 04 …	Set information data set
46 07 44 …	Secondary item identifier data set
83 02 07 …	Owner institution data set

Annex E
(informative)

Implementation and migration

E.1 New RFID implementations

Any library that has yet to introduce RFID should give consideration to the use of this document. Depending on the time that the implementation is being considered, particularly immediately after publication, vendors might be at incomplete stages of their development and support for this document. In situations where not all features are supported but are considered essential for the library system, the institution should seek a development path and timetable from the vendor.

Once this document is more widely adopted, there will be an increasing choice, particularly of RFID devices and tags. These need to be interoperable with any that are already installed. The performance and conformance requirements (see Clause 4) should be used to evaluate any new device or tag component.

E.2 Legacy RFID implementations using ISO/IEC 18000-3 Mode 1 RFID tags

The decision to migrate from a legacy implementation to a data model based on this document depends on many economic and operational considerations that are beyond the scope of this document. Some of the factors that will influence change are if vendors are offering features in applications that are considered beneficial, or if jobbers are prepared to supply items with RFID tag labels fixed and even encoded.

The first point for these legacy systems is to confirm that the RFID tags that are being used are compatible with ISO/IEC 18000-3 Mode 1. This can also apply to products that are defined as "ISO/IEC 15693 conformant", because these are based on integrated circuits that were originally used for smart card.

Discussions should take place with the RFID tag supplier, and supplier of printer encoders and RFID readers to define the specific features of the products already installed, compared with the RFID tag requirement defined in Clause 8 and the support for commands defined in Annex A. Although the air interface protocol is the same, the detailed features differ between tags and devices from different vendors. This procedure confirms if the existing tags are conformant with ISO/IEC 18000-3 Mode 1 for use for item management.

If this is confirmed, procedures should be established to make use of the correct AFI code value for items on loan and for items that are in-stock (see 7.2.2) if a dual AFI system is used for security purposes. If the RFID tags used in the legacy system are shown to be non-conformant, the library should consider the advice in E.3.

It is almost certain even with the same air interface protocol, that the data model and the encoding rules for the present implementation are different from the data model and encoding rules as defined in this document. Figure E.1 describes a migration model.

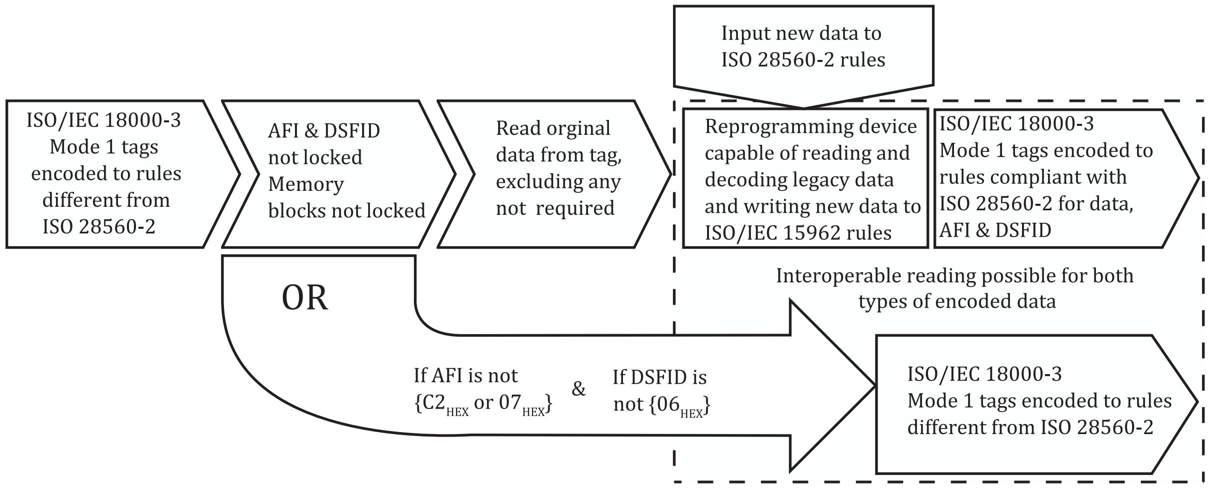

Figure E.1 — Migration model

Figure E.1 shows that there is a requirement to discriminate between a domestic legacy RFID tag and one that is compliant with this document. The DSFID is probably the easiest feature to use to discriminate between the legacy and compliant tags with this document. There is only 1 in 256 probabilities that the legacy system uses the same DSFID, and there is also the possibility that no DSFID code is used on the legacy RFID tags. There are other mechanisms that could be used and implemented in software.

The simplest plan to convert the data on the tag to be compliant with this document is to erase all the data on the RFID tag and overwrite according to the rules of ISO/IEC 15962. Take account of any requirements to exclude data on the legacy tag that is no longer required and to include new data elements not previously encoded. A range of procedures, including software programs, can be developed to ensure that the migration is achieved in an efficient manner.

This procedure to erase and overwrite is not possible if any of the data are locked on the existing tag. The option then exists to maintain both systems in parallel, at least for RFID tags with locked data until the number of legacy tags is relatively small to cost-justify a replacement RFID tag.

The changeover process can be managed in a number of ways from a gradual changeover, possibly as items are returned, to a more accelerated programme. This choice is left to the discretion of the individual library, given that however short the changeover programme, there is a requirement to maintain in parallel both the legacy system and the system compatible with this document.

A particular business operation can be in place, and discussions should take place with vendors to ensure that these functions are supported in the new system compliant with this document. There are no features in this document that should impede particular business operations.

E.3 Legacy RFID implementations using other RFID tags

If it is confirmed that the RFID tags, which have been used in a legacy system, do not conform to the requirements of ISO/IEC 18000-3 Mode 1, there are still possibilities for migration.

If the RFID tag used in the legacy system operates at 13,56 MHz, it might be possible to convert to the ISO/IEC 18000-3 Mode 1 tag, because some aspects of the air interface protocol are similar. Effectively, what is required is to introduce a system that supports multiple air interface protocols. This would probably require an upgrade and installation of components for interrogators and printer encoders, with the latter type of device probably presenting a more complex challenge.

If the vendor of the RFID equipment can support migration through the installation of upgraded components and possibly devices, then the advice provided in E.2 is generally applicable.

If the existing RFID system is operating at a different frequency (e.g. 125 kHz to 135 kHz or 2,45 GHz), then a significantly more complex challenge exists with radio frequency. RFID systems that operate at frequencies other than 13,56 MHz have significantly different operational features from those operating at 13,56 MHz. A library system using a different RFID frequency is likely to require a significant upgrade of devices to ensure that the system operates on a dual-frequency basis. Such technological developments are possible, and some are taking place for particular frequencies and particular air interface protocols. Addressing this challenge is beyond the scope of this document.

Bibliography

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

[2] ISO 3166-1, *Codes for the representation of names of countries and their subdivisions — Part 1: Country code*

[3] ISO/IEC 8859-1, *Information technology — 8-bit single-byte coded graphic character sets — Part 1: Latin alphabet No. 1*

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

[5] ISO/IEC 10646, *Information technology — Universal coded character set (UCS)*

[6] ISO 15511, *Information and documentation — International standard identifier for libraries and related organizations (ISIL)*

[7] ISO/IEC 15961-2, *Information technology — Radio frequency identification (RFID) for item management: Data protocol — Part 2: Registration of RFID data constructs*

[8] ISO/IEC 15961-3, *Information technology — Radio frequency identification (RFID) for item management: Data protocol — Part 3: RFID data constructs*

[9] ISO/IEC 15693 (all parts), *Identification cards — Contactless integrated circuit cards — Vicinity cards*

[10] SIP-2 *3M Standard Interchange Protocol*

[11] ANSI/NISO Z39 83, *NISO Circulation Interchange Protocol (NCIP)*

ICS 35.240.30; 35.040.50
Price based on 43 pages