TECHNICAL SPECIFICATION

ISO/TS 28560-4

Second edition
2023-07

Information and documentation — RFID in libraries —

Part 4: Encoding of data elements based on rules from ISO/IEC 15962 in an RFID tag with partitioned memory

Information et documentation — RFID dans les bibliothèques —

Partie 4: Encodage des éléments de données RFID fondé sur les règles de l'ISO/CEI 15962 dans une étiquette de RFID avec la mémoire divisée

Reference number
ISO/TS 28560-4:2023(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

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 second edition cancels and replaces the first edition (ISO/TS 28560-4:2014), which has been technically revised.

The main changes are as follows:

- a few updates are made and a new clause (see Clause 11) about RFID Emblem has been added;
- Annexes B, C and D have been changed to informative.

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.

A standard data model, taking into account the lessons learned from the national schemes and vendor solutions was developed with ISO 28560-1, which defines the set of mandatory and optional data elements. ISO 28560-2 and ISO 28560-3 define encoding rules for those libraries that choose to use High Frequency RFID technology operating at 13,56 MHz.

This document defines encoding rules for those libraries that choose to use UHF RFID technology operating at 860 MHz to 960 MHz, with the interrogators (readers) set to conform to local radio regulations that specify only part of this spectrum. The UHF tags can function efficiently in any of the radio regulated regions. This document uses encoding rules that are specified in ISO/IEC 15962, as does ISO 28560-2. Some of the encoding rules are different because of the nature of the different RFID technology, but a number of rules are similar if not identical.

Information and documentation — RFID in libraries —

Part 4:
Encoding of data elements based on rules from ISO/IEC 15962 in an RFID tag with partitioned memory

1 Scope

This document defines rules for ISO 28560-1 data elements to be encoded in radio frequency identification (RFID) tags with a memory structure that is partitioned into four memory banks. This primarily applies to ISO/IEC 18000-63 (previously known as ISO/IEC 18000-6 Type C) operating in the UHF frequency, but not necessarily restricted to this technology.

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 rules for encoding a unique item identifier in a specific memory bank, known as MB 01, taking into account different requirements for privacy. It also defines the rules for encoding other relevant data in a separate memory bank, known as MB 11. Each of these memory banks is addressable using different command set of the appropriate RFID technology.

As with other parts of ISO 28560, this document is appropriate for the needs of all types of libraries (including academic, public, corporate, special, and school libraries).

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 18000-63, *Information technology — Radio frequency identification for item management — Part 63: Parameters for air interface communications at 860 MHz to 960 MHz Type C*

ISO/IEC 18046-1, *Information technology — Radio frequency identification device performance test methods — Part 1: Test methods for system performance*

ISO/IEC 18046-2, *Information technology — Radio frequency identification device performance test methods — Part 2: Test methods for interrogator performance*

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

ISO/IEC 18047-6, *Information technology — Radio frequency identification device conformance test methods — Part 6: Test methods for air interface communications at 860 MHz to 960 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 terminology 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

Note 1 to entry: In this document, the term is only relevant to Memory Bank 11, containing optional data elements.

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) through the interrogator

3.4
application family identifier
AFI
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

Note 1 to entry: In this document, the term is only relevant to Memory Bank 01, containing the data elements comprising the UII.

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
component of the *DSFID* (3.8) that is a mechanism used in the data protocol to identify how *object identifiers* (3.12) 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: In this document, the term is only relevant to Memory Bank 11, containing optional data elements. The data format declares the *Root-OID* (3.14) in an efficient manner, so that a complete object identifier can be reconstructed for external communications.

3.7
data protocol processor
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
data storage format identifier
DSFID
code that consists of, at least, the *access method* (3.1) and *data format* (3.6)

Note 1 to entry: In this document, the term is only relevant to Memory Bank 11, containing optional data elements.

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

3.10
Memory Bank
MB
designated name of a *segmented memory structure* (3.15)

Note 1 to entry: In this document, the Memory Banks 00, 01, 10, and 11 are using binary notation.

3.11
metadata
type of data or information about data

Note 1 to entry: In the context of this document, metadata can be the *Relative-OID* (3.13) 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.

3.12
object identifier
value (distinguishable from all other such values), which is associated with an object

3.13
Relative-OID
particular *object identifier* (3.12) that constitutes the remaining *arcs* (3.5) after the *Root-OID* (3.14)

3.14
Root-OID
particular *object identifier* (3.12) that constitutes the first, second, and subsequent common *arcs* (3.5) of a set of object identifiers (hence, the common root)

3.15
segmented memory structure
memory storage that is separated into separate elements and requires multiple addressing elements for access

Note 1 to entry: In this document, this has the same meaning as partitioned memory.

3.16
tag driver
implementation of the process to transfer data between the data protocol processor and the RFID tag

3.17
unique item identifier
UII
encodable data that when combined with an object identifier prefix renders the combination unique within the rules of the application domain

4 Applicability and relationship with other systems

4.1 General

The use of the ISO/IEC 18000-63 air interface protocol brings with it a set of different standards which can be deployed to support RFID in libraries. Three different device architectures are discussed below from the library application to the RFID tag. It should be noted that with the development in UHF RFID, the LMS/ILS is not the only end point (or start point for encoding) in an RFID system. Therefore, under the heading of "library application", the following also needs to be considered:

— the library management system/integrated library system;

— software linked to encoding devices used by book suppliers;

— portable devices, with on-board processing, which exchange data (as necessary) with the LMS/ILS on a transaction or even batch mode basis;

— sortation systems that can operate somewhat autonomously from the LMS/ILS;

— software to support quality control devices;

— mobile phones and other user-centric portable devices.

Some of the architectures described below offer different perspectives on how to achieve interoperability. The one described in 4.2 offers more opportunities for using generic modules, while the one described in 4.3 can be called the "traditional model" replicating structures used by libraries that have implemented RFID using 13,56 MHz technology, while the one described in 4.4 can be considered a compromise between the other two architectures. There is no requirement to adopt one of these architectures; in fact, some might be more suited to some type of device (e.g. portable devices, RFID tunnel readers) or some type of operator (e.g. a book supplier compared with a circulation library). Some of the choices might depend on the interfaces supported by hardware devices and software components.

4.2 Independent standards-based components

Figure 1 shows an architecture where individual hardware and/or software modules communicate between different layers.

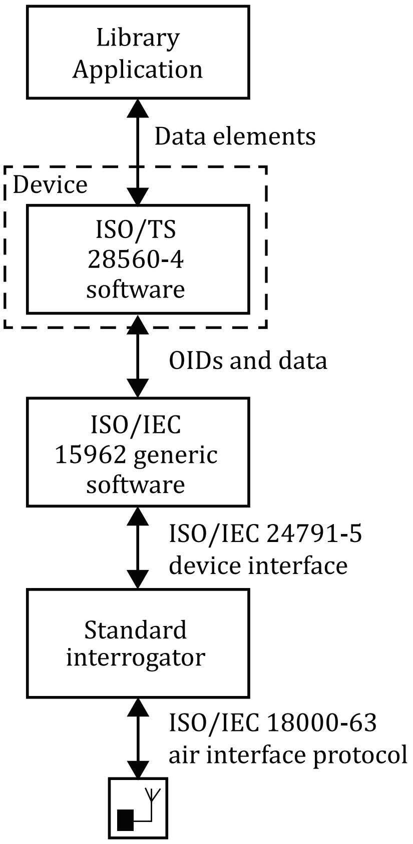
Library Application
Data elements
Device
ISO/TS 28560-4 software
OIDs and data
ISO/IEC 15962 generic software
ISO/IEC 24791-5 device interface
Standard interrogator
ISO/IEC 18000-63 air interface protocol

Figure 1 — Architecture using all standard components

Using the example of reading data from the tag (i.e. bottom up in Figure 1, the inverse applies for encoding data).

a) The ISO/IEC 18000-63 air interface protocol is used between the tag and the interrogator to transfer encoded bits within commands and responses.

b) The interrogator supports the device interface standard (ISO/IEC 24791-5) and communicates to upper layers using this protocol, typically to a stand-alone ISO/IEC 15962 software implementation.

c) In turn, after decoding the data on the tag, the ISO/IEC 15962 software simply communicates to an ISO/TS 28560-4 software implementation:

   — object identifiers and the de-compacted data, where these are encoded using standard ISO/IEC 15962 compaction rules;

   — object identifiers and the still compacted data bytes for data elements defined as application-specific.

d) The device, shown in dotted lines, is capable of processing some front-line function. A typical example is the self-check terminal, but can be any other device designed or configured for a library application. The ISO/TS 28560-4 software module can be integrated in the device or can directly interface with it, based on the implementation and design requirements. This software decodes the additional application-specific data elements and passes on all the data elements to specific software in the library device (e.g. circulation control terminals, sortation systems, portable devices) for processing.

e) The library device processes the data elements in a manner relevant to library applications, including the LMS/ILS using a communication protocol implemented by the library, such as SIP 2.0.

4.3 Integrated encoding/decoding software

Figure 2 illustrates an architecture that has a more integrated software component but uses the standard air interface protocol and the device interface protocol from the interrogator module.

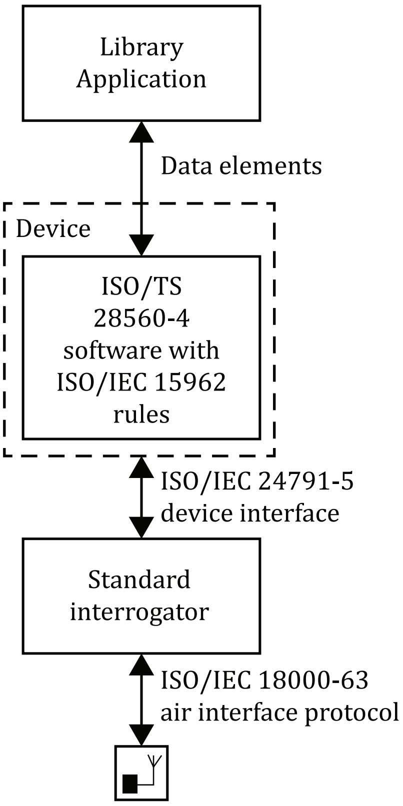
Library Application
Data elements
Device
ISO/TS 28560-4 software with ISO/IEC 15962 rules
ISO/IEC 24791-5 device interface
Standard interrogator
ISO/IEC 18000-63 air interface protocol

Figure 2 — Architecture with integrated data element software component

The software module that processes the data according to ISO/IEC 15962 rules is incorporated into a software product that also deals with all the ISO/TS 28560-4 encoding rules. This means that the interface between the software and any device with which it is associated or embedded communicates ISO 28560-1 data elements to library application modules.

4.4 Legacy-based architecture

Figure 3 shows an architecture that is not dissimilar to that, which is common for RFID systems based on 13,56 MHz technology.

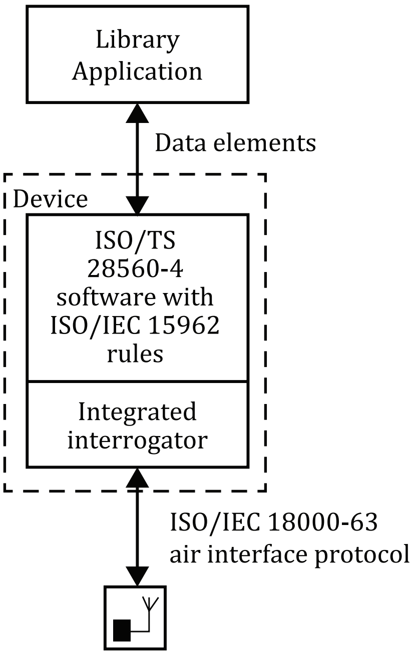

Figure 3 — Legacy-based architecture

Using the same example of reading data from the tag, as in 4.2, the ISO/IEC 18000-63 air interface protocol communicates directly with an interrogator integrated directly into the front-line library device. This either embeds, or makes calls to, a software module that supports all the requirements of ISO/TS 28560-4 including the encoding rules of ISO/IEC 15962.

The advantage of this architecture is that it might be familiar to system vendors. The disadvantage is that the application interface of the interrogator is probably based on some proprietary protocol. In turn, this means that some aspects of the ISO/IEC 28560-4 software might also require a customised interface.

5 Requirements

5.1 Data elements

The data elements shall be conformant 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 conforming with the basic set of data elements.

5.2 RFID air interface: ISO/IEC 18000-63 for UHF

5.2.1 General

The air interface for compliant RFID tags and interrogators is specified in ISO/IEC 18000-63. RFID tags have what is known as a segmented memory structure, where four different memory banks are supported and separately addressable. The memory banks are using the following binary notation:

- 00 for password;
- 01 for the unique item identifier;
- 10 for tag identification, which can include serialization;
- 11 for additional user data, which in the case of ISO/TS 28560-4, will include the optional data.

Memory is organized in a 16-bit word for commands to read and write the data, but the actual memory structure is left to the chip manufacturer to decide on how this is implemented.

There are different national and regional radio regulations for the use of RFID within the UHF frequency spectrum between 860 MHz and 960 MHz.

5.2.2 Air interface conformance

The air interface conformance shall be tested in accordance with the procedures of ISO/IEC 18047-6.

5.2.3 Tag performance

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

5.2.4 Interrogator performance

Where there are requirements to test interrogator (reader) performance, these shall be done in accordance with ISO/IEC 18046-2.

5.2.5 System performance

Where there are requirements to test system performance, these shall be done in accordance with ISO/IEC 18046-1.

5.3 RFID air interface: Other air interface protocols

Although the segmented memory structure is used for other air interface protocols, this document currently provides no specific advice and guidelines on implementing with an air interface protocol other than ISO/IEC 18000-63.

5.4 Data protocol

ISO/IEC 15961-1 specifies the application commands that are used to define the communication requirements between the application and the RFID tag. The relevant commands are described 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:

- encoding in memory bank 00 is for passwords;
- encoding in memory bank 01 shall comply with the ISO/IEC 15962 rules for a Monomorphic-UII. Encoding in MB 01 is mandatory with the rules as defined in 6.2;
- encoding in memory bank 11 shall comply with the No-directory access method, and be used for encoding the optional data elements defined in ISO 28560-1;
- no alternative access method shall be supported until this document is revised;
- no encoding is possible in memory bank 10.

Memory bank 11 is defined as optional in ISO/IEC 18000-63, and therefore, not all RFID tags include this memory bank. Increasingly, MB 11 is incorporated in RFID tag products, so should be used to support the encoding rules defined in this document.

5.5 RFID interrogators (RFID readers)

RFID interrogators shall support all memory banks so that tags with three or four memory banks and different sized memory are all interoperable.

In order to achieve interoperability, RFID interrogators shall be based on open architecture RFID standards defined by ISO/IEC JTC 1/SC 31. Particular standards are specified in this document. This means that any one of the 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 and adapted for this document as outlined 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

0	Unique item identifier (UII)	Mandatory	One of six formats
			{Primary item identifier}
			{Primary item identifier}.S
			{Primary item identifier}.{set information}
			{Owner institution}.{Primary item identifier}
			{Owner institution}.{Primary item identifier}.S
			{Owner institution}.{Primary item identifier}.{set information}
				Should be locked

1	Primary item identifier	Not explicitly used (see UII *N* 0)	Variable length alphanumeric Character set = ISO/IEC 646 International Reference Version (IRV)	Not applicable
2	Content parameter	Optional	Bit mapped code (see 6.4)	Optional
3[b]	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	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

Table 1 *(continued)*

N[a]	Name of the data element	Status	Display format	Lock
10	Order number	Optional	Variable length alphanumeric Character set = ISO/IEC 646 IRV	Optional
11[b]	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, or ISO/IEC 8859-1, or UTF-8	Optional
16	Local data B	Optional	Variable length alphanumeric Character set = ISO/IEC 646 IRV, or ISO/IEC 8859-1, or UTF-8	Optional
17	Title	Optional	Variable length alphanumeric Character set = ISO/IEC 646 IRV, or ISO/IEC 8859-1, or UTF-8	Optional
18	Product identifier local	Optional	Variable length Alphanumeric Character set = ISO/IEC 646 IRV	Optional
19	Media format (other)	Optional	Single octet (coded list)	Optional
20	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
26	Local data C	Optional	Variable length alphanumeric Character set = ISO/IEC 646 IRV, or 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	—	—

Table 1 *(continued)*

N[a]	Name of the data element	Status	Display format	Lock
[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.				

6.2 Unique item identifier (UII)

The unique item identifier (UII) is a mandatory data element to be encoded in Memory Bank 01 of an RFID tag with a segmented memory structure. The UII shall be encoded using the rules defined in ISO/IEC 15962 for a Monomorphic-UII, which declares the object identifier and encoding scheme directly from the AFI.

NOTE 1 The Relative-OID does not need to be encoded, nor is a DSFID or precursor required for MB 01.

Specifically, the encoding shall comply with the URN Code 40 encoding rules as defined in ISO/IEC 15962. This enables this document to support three components (ISIL, primary item identifier, and set information) resulting in six possible structures for the UII, one of which shall be selected for encoding in a particular RFID tag. Because of the structure of the UII, the options can be intermixed within a system. The structures are described in the following subclauses.

NOTE 2 Including the ISIL (if it exists for the library) in the UII can result in privacy issues because the tag is more uniquely identifiable. Encoding the ISIL as optional data in MB 11 is of lesser concern. Excluding the ISIL from the UII can impact aspects of transactions for ILL items.

The URN Code 40 encoding rules require a 'dot' separator to be placed between the component parts of the UII. To ensure that this is always the case, a 'dot' (also known as a ‘full stop’ or ‘period’, ISO/IEC 8859-1 code point $2E_{HEX}$) shall not be part of either the primary item identifier or the ISIL for any of the structures defined in the following subclauses.

Although beyond the scope of this document, characters other than the 'dot' need to be used if they currently appear in either the ISIL or the primary item identifier and are encoded in the UII. This is because the presence of a 'dot' in the wrong place will result in a corrupted decode.

The UII should be locked to prevent various forms of digital vandalism and to ensure proof of ownership by a particular library. The procedure for locking MB 01 is defined in Annex C.

MB 01 has a signal, encoded by air interface protocol rules, to indicate if encoding is present in MB 11 for the optional data elements (see 7.3.4).

The Relative-OID value '0' is created by the RFID decoding process from the information registered for the AFI associated with the Monomorphic-UII. A library system may retain this Relative-OID as the data element identifier, or map the constituent part(s) of the UII to data elements as defined in the following subclauses.

6.2.1 UII comprising of only the primary item identifier

The input format for this structure shall comprise a single component:

{Primary item identifier}

The format of the primary item identifier 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 will support any length of primary item identifier, shorter codes will encode more efficiently, requiring less memory and enabling faster transactions across the air interface.

After decoding, the Relative-OID may be retained as the value '0' assigned by the RFID decoding process, or mapped to Relative-OID '1'.

6.2.2 UII comprising owner institution + primary item identifier

The input format for this structure shall comprise two components with a 'dot' separator between them:

{Owner institution (ISIL)}.{Primary item identifier}

The input of the ISIL for the owner institution to the encoding process and the output from the decoding process shall be as defined in 6.5, including the hyphens. Because it always precedes the 'dot' separator if encoded in the UII, it is distinguishable from the primary item identifier, which is the second component.

The input of the primary item identifier is defined in 6.2.1.

After decoding, the Relative-OID may be:

- retained as the value '0' assigned by the RFID decoding process, retaining the 'dot' separator, or
- mapped to Relative-OID '3' for the owner institution and Relative-OID '1' for the primary item identifier. In this case, the 'dot' separator is discarded.

6.2.3 Encoding set information

Because ISO/IEC 18000-63 requires different commands to read the different memory banks, there are advantages in providing an optional structure in the UII to signal that the item is part of a set. This can save on the need to have an additional read of MB 11. This subclass recognizes that libraries use different approaches to define the primary item identifier when the item is part of a set.

6.2.3.1 Set indicator when the primary item identifier is unique between all items in the set

The input format for this structure shall append an additional component, preceded by a 'dot' separator following the structures defined in 6.2.1 and 6.2.2:

{Primary item identifier}.S

{Owner institution (ISIL)}.{Primary item identifier}.S

The final character 'S' is just an indicator, and may be used in the application to invoke secondary procedures to identify other information. This can be in MB 11 for the encoding of the data element for set information (see 6.6).

6.2.3.2 Set indicator when the primary item identifier is not unique between all items in the set

If the primary item identifier is an all-numeric code of 2, 4, or 6 digits long, then, this subclause shall not be applied, because it will be impossible for the decoding software to distinguish between the structures. If it is necessary to declare that an item is part of a set, then the rules of 6.2.3.1 shall be used.

The input format for this structure shall append an additional component, preceded by a 'dot' separator following the structures defined in 6.2.1 and 6.2.2:

{Primary item identifier}.{set information}

{Owner institution (ISIL)}.{Primary item identifier}.{set information}

The set information is constructed exactly as defined in 6.6.

NOTE 1 This option makes the UII unique, but still preserves the integrity of the primary item identifier as a common code for the interfaces to the LMS/ILS.

NOTE 2 All the rules that are defined in ISO 28560-1 for sets might not be applicable to this solution.

6.2.4 Unambiguous UII structure

The six structures of the UII, as listed below, can all be unambiguously decoded. Those that include the ISIL as the first component have a structure that is defined in ISO 15511, so that one to four alphabetic characters precede a 'hyphen' character. Those with the ISIL component cannot have a single component. Those that have a set indicator (irrespective of option) as the final component have this as a very short string. This can never be longer than six characters and is always preceded by the primary item identifier.

- {Primary item identifier}
- {Primary item identifier}.S
- {Primary item identifier}.{set information}
- {Owner institution (ISIL)}.{Primary item identifier}
- {Owner institution (ISIL)}.{Primary item identifier}.S
- {Owner institution (ISIL)}.{Primary item identifier}.{set information}

6.3 Primary item identifier

This data element as defined in ISO 28560-1 shall only be used in this document as part of the UII (see 6.2).

6.4 Content parameter

The content parameter is an optional data element that shall only be encoded in Memory Bank 11. If encoded, it should be encoded immediately after the DSFID (see 7.1.7). If the content parameter is to remain unlocked and the tag supports selective locking, the block lock size of the tags being used by the library need to be taken into account. An unlocked content parameter, preceded by a locked DSFID, and followed immediately by other locked data elements can consume a significant portion of memory in a tag with a small total memory capacity and a relatively large block size.

The content parameter is used to declare the Relative-OID values that are encoded on the RFID tag, and for the purposes of this document, it is used as an OID index. A 'rule of thumb' has evolved with reading data from Memory Bank 11 using the inherent ISO/IEC 15962 encoding schemes when applied to an ISO/IEC 18000-63 RFID tag. That is, if the encoded data is up to 256 bits or 16 words, an air interface command set to read 16 words is probably faster than invoking successive reads. If this rule is applied, and if the encoded data requires no more than 16 words, this means that the OID index might be redundant. If it is difficult to determine the memory required for the encoding of a particular set of data elements, then another metric is the number of data elements being encoded. The OID index should be used if more than five additional data elements are encoded in Memory Bank 11. However, if the Title is one of the data elements, then that number needs to be reduced, especially if the full title is being encoded. When creating the OID index, consideration should be given to the highest value Relative-OID likely to be encoded and the OID index constructed with sufficient capacity to support this Relative-OID.

If used, 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 4.

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 4 — Example of OID index bit map

In the example in Figure 4, 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 the highest value Relative-OID intended to be encoded either initially or with subsequent encoding. It is also necessary to round up the bit map to 8-bit boundaries for encoding on the RFID tag.

This data element provides no information about the sequence of the encoded data elements nor their size. In the example in Figure 4, the encoding sequence could be Relative-OID value 8, followed by 11, followed by 3.

The OID index should only be locked if the information on the RFID tag is certain to remain unchanged.

6.5 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 after the ISIL prefix) is presented in the application commands.

This data element is defined in ISO 28560-1 as optional, and for this document, there are three options for a library to consider for processing the owner institution data element, as follows:

- it can be encoded as part of the UII in Memory Bank 01 as defined in 6.2.1;
- it can be excluded from the UII and encoded in Memory Bank 11;
- it can be excluded from the encoding.

6.6 Set information

The data element for set information is optional, and for this document, there are three options for a library to consider for processing the owner institution data element, as follows:

- it can be encoded as part of the UII in Memory Bank 01 as defined in 6.2.3;
- it can be excluded from the UII and encoded in Memory Bank 11;
- it can be excluded from the encoding.

The set information is presented in two components:

- the total number of parts;
- followed by 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 nine or less, then the user data is 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 is 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 is 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.

6.7 Type of usage

The data element for type of usage is optional. If used, it shall only be encoded in Memory Bank 11. The encoding rules defined for ISO 28560-2 shall apply and are repeated below.

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.

6.8 Shelf location

The data element for shelf location is optional. If used, it shall only be encoded in Memory Bank 11. The encoding rules defined for ISO 28560-2 shall apply and are repeated below.

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

6.9 ONIX media format

The data element for ONIX media format is optional. If used, it shall only be encoded in Memory Bank 11. The encoding rules defined for ISO 28560-2 shall apply and are repeated below.

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.

6.10 MARC media format

The data element for MARC media format is optional. If used, it shall only be encoded in Memory Bank 11. The encoding rules defined for ISO 28560-2 shall apply and are repeated below.

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.

6.11 Supplier identifier

The data element for supplier identifier is optional. If used, it shall only be encoded in Memory Bank 11. The encoding rules defined for ISO 28560-2 shall apply and are repeated below.

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

6.12 Order number

The data element for order number is optional. If used, it shall only be encoded in Memory Bank 11. The encoding rules defined for ISO 28560-2 shall apply and are repeated below.

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

6.13 ILL borrowing institute

The data element for ILL borrowing institute is optional. If used, it shall only be encoded in Memory Bank 11. The encoding rules defined for ISO 28560-2 shall apply and are repeated below.

The ILL borrowing institution is represented by the ISIL code in accordance with ISO 15511. The data is presented according to the rules defined in 6.5 (for the owner institution). This data element shall not be locked.

6.14 ILL transaction number

The data element for ILL transaction number is optional. If used, it shall only be encoded in Memory Bank 11. The encoding rules defined for ISO 28560-2 shall apply and are repeated below.

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

6.15 GS1 product identifier

The data element for GS1 product identifier is optional. If used, it shall only be encoded in Memory Bank 11. The encoding rules defined for ISO 28560-2 shall apply and are repeated below.

The GS1 product identifier data element is used to encode 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.

NOTE 3 In the future, products, including books, can be supplied with what is known in the GS1 system as the SGTIN, which will be encoded in MB 01. This should be overwritten with the UII as defined in 6.2.

6.16 Alternative unique item identifier

This data element shall not be used in this document. It is reserved for possibly encoding in different tag architectures.

6.17 Local data

The data element for local data is optional. If used, it shall only be encoded in Memory Bank 11. The encoding rules defined for ISO 28560-2 shall apply and are repeated below.

The local data elements (A, B, and C) are each variable length fields that may be used for any locally defined purpose and as such there is no external application of this data object. Table 2 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, or ISO/IEC 8859-1, or UTF-8	Optional
Local data B	16	Optional	Variable length alphanumeric field Character set = ISO/IEC 646 IRV, or ISO/IEC 8859-1, or UTF-8	Optional
Local data C	26	Optional	Variable length alphanumeric field Character set = ISO/IEC 646 IRV, or ISO/IEC 8859-1, or UTF-8	Optional

6.18 Title

The data element for title is optional. If used, it shall only be encoded in Memory Bank 11. The encoding rules defined for ISO 28560-2 shall apply and are repeated below.

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

— 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 may 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).

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, or ISO/IEC 8859-1, or UTF-8	Optional

6.19 Product identifier (local)

The data element for product identifier (local) is optional. If used, it shall only be encoded in Memory Bank 11. The encoding rules defined for ISO 28560-2 shall apply and are repeated below.

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 may be used. This enables information systems linked to specific local code structures to be supported by the RFID system.

6.20 Media format (other)

The data element for media format (other) is optional. If used, it shall only be encoded in Memory Bank 11. The encoding rules defined for ISO 28560-2 shall apply and are repeated below.

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.

6.21 Supply chain stage

The data element for supply chain stage is optional. If used, it shall only be encoded in Memory Bank 11. The encoding rules defined for ISO 28560-2 shall apply and are repeated below.

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.

6.22 Supplier invoice number

The data element for supplier invoice number is optional. If used, it shall only be encoded in Memory Bank 11. The encoding rules defined for ISO 28560-2 shall apply and are repeated below.

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

6.23 Alternative item number

The data element for alternative item number is optional. If used, it shall only be encoded in Memory Bank 11. The encoding rules defined for ISO 28560-2 shall apply and are repeated below.

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

6.24 Alternative owner institution

The data element for alternative item number is optional. If used, it shall only be encoded in Memory Bank 11, and shall not be used as part of the UII (see 6.2). This means that libraries that do not have an assigned ISIL shall only encode the primary item identifier as the UII. The encoding rules defined for ISO 28560-2 shall apply and are repeated below, with some advice about future migration.

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 may be deemed necessary to lock this data element, this is left as a local library decision. Some libraries may choose to leave the data element unlocked so that it can be deleted, if necessary, as a result of library mergers or transfer of collections, or a future migration to the ISIL code. Another option is to lock the data element and treat the data element as redundant (and therefore not processed) when an ISIL is assigned.

6.25 Subsidiary of an owner library

The data element for subsidiary of an owner library is optional. If used, it shall only be encoded in Memory Bank 11. The encoding rules defined for ISO 28560-2 shall apply and are repeated below.

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.

6.26 Alternative ILL borrowing institution

The data element for alternative ILL borrowing institution is optional. If used, it shall only be encoded in Memory Bank 11. The encoding rules defined for ISO 28560-2 shall apply and are repeated below.

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

6.27 Other reserved data elements

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

7 Data encoding

7.1 Data protocol overview

7.1.1 General

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 standards specified by ISO/IEC JTC 1/SC 31, will allow libraries to migrate more easily if any developments in RFID technology were 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 ISO/IEC 15961 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. Different system architectures are described in Clause 4.

7.1.2 Data constructs

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 7.1.3 to 7.1.7, together with their particular code values that have been assigned by the ISO/IEC 15961 registration authority for use for RFID for libraries.

7.1.3 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. 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.

The AFI is encoded in MB 01 (see 7.3.5.2). For this document, the AFI declares that the UII that is also encoded in MB 01 is a Monomorphic-UII. Therefore, MB 01 does not require a DSFID to be encoded.

No other value of AFI shall be used in MB 01. This is to ensure that the rules registered for the data constructs according to ISO/IEC 15961-2 are consistently applied.

NOTE This means that one of the security procedures defined for other parts of ISO 28560 based on ISO/IEC 18000-3 Mode 1 tags, permitting switching between two AFI values, cannot be applied.

7.1.4 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 7.1.7.

For this document, the DSFID, and therefore the data format, are only encoded in MB 11.

7.1.5 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 all parts of ISO 28560, but between all domains. The object identifier may 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 will need to be encoded. There is one exception, the UII encoded in MB 01 has an implicit Relative-OID = 0, which is not encoded. This is because the UII is based on the ISO/IEC 15962 Monomorphic-UII rules.

Software designed specifically for the library community will probably require only 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 will need to 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.

7.1.6 Object identifier for the UII and its interpretation

The Relative-OID for the UII has an implicit value of '0' for any of the formats to comply with the registered AFI and Monomorphic-UII. Two options may be used during the decode process, as set out in more detail in 6.2. However, for the encode procedure, any Relative-OID assigned to a component part of the UII shall be ignored.

7.1.7 DSFID

The DSFID shall be encoded in the first byte of MB 11. It has two components relevant to this part of ISO 28560, as follows:

- the data format, as defined in 7.1.4, 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 is structured in MB 11 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.

Other access methods have been included in ISO/IEC 15962:2013 (second edition). 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 7.3.10.

7.2 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 18000-63 RFID tags are defined 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). ISO/IEC 15961-1 no longer requires a detailed interface mechanism with ISO/IEC 15962, as 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.

7.3 ISO/IEC 15962 encoding rules for this document

7.3.1 General

The memory of an ISO/IEC 18000-63 tag is divided into four memory banks as defined in 5.2.1. Three of the memory banks can be encoded, whereas MB 10 is written to by the manufacturer of the integrated circuit and thereafter is read-only.

Memory is organized in 16-bit words, and a word is the minimum unit that can be written to the tag or read from the tag. Commands are addressed in word number starting at 0_{HEX}. However, some of the structures of memory are defined as bit locations with the first bit in each memory bank identified as 00_{HEX}.

There are no standard air interface commands to determine which words are locked; ISO/IEC 18000-63 simply states “A Tag’s lock bits cannot be read directly; they can be inferred by attempting to perform other memory operations.”

Figure 5 illustrates the basic architecture of the data protocol. The components of ISO/IEC 15962 are discussed below.

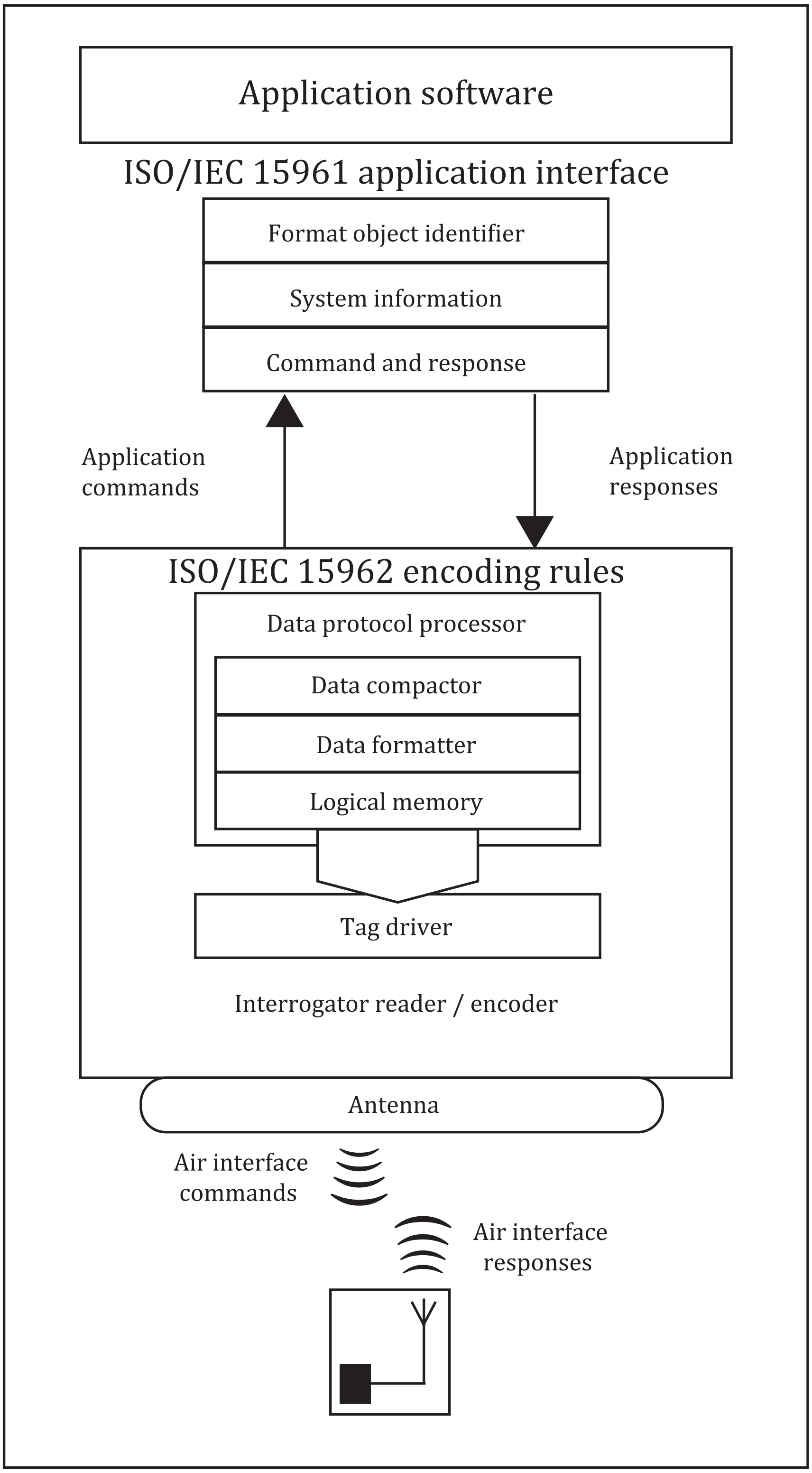

Figure 5 — ISO RFID data protocol architecture

The logical memory (see Figure 5) is the software equivalent of the structure of the memory on the RFID tag itself. RFID tags that are compliant with the specified air interface protocol will have different architectures. Although the logical memory should apply to all memory banks of an ISO/IEC 18000-63 RFID tag, it is most relevant to MB 11 for this document because of the variants in tag and interrogator products.

The following clauses identify the structure and rules as applicable for this document.

7.3.2 Structure of MB 00

This memory bank is used to store passwords. The 32-bit Kill password is stored at locations 00_{HEX} to $1F_{\text{HEX}}$. The un-programmed value of this password is a 32-bit zero string. An interrogator can use the Kill password to kill a tag and render it unresponsive thereafter.

The 32-bit Access password is encoded at location 20_{HEX} to $3F_{\text{HEX}}$. The default un-programmed value is a 32-bit zero string. A tag with a non-zero Access password requires the interrogator to issue this password before subsequent processing with the tag memory.

Each password may be locked, but as shown in 7.3.1, this is not straightforward to determine.

7.3.3 Encoding and use of MB 00

7.3.3.1 Kill password

ISO/IEC 18000-63 specifies two states for the Kill password: a zero value and a non-zero value. A zero value still enables this password to be set, so if a library leaves the password set to the zero state and is unlocked, it leaves itself at some risk to a two-stage attack. In the first stage, a non-zero Kill password can be entered into the “empty area”, and then in the second stage, this password can be used to kill the tag and render it unreadable.

To protect libraries against tags being destroyed, various options are discussed in 9.2.

Because the Kill password is not required for normal processing, loan items that have the Kill password encoded can be used for inter-library loan purposes without any adverse implications. The only time that the Kill password needs to be invoked is at the end of the useful life of the library loan item, when data on the tag should be rendered unreadable.

7.3.3.2 Access password

Although not all ISO/IEC 18000-63 tags support an Access password in terms of providing a memory area for this and support for commands, for the highest level of security tags with this feature should be used. If an Access password is supported, then it can be used to restrict unauthorized writing of any data from the RFID tag.

To protect libraries against unauthorized modification of data on tags, various options are discussed in 9.2.

7.3.4 Structure of MB 01

This memory bank contains the UII and associated syntax. The first word at memory address location 00_{HEX} to $0F_{\text{HEX}}$ contains a stored CRC-16. This is automatically generated when the tag is processed and the rules for that are beyond the scope of this document. The second word contains a protocol control word at memory locations 10_{HEX} to $1F_{\text{HEX}}$ as shown in Table 4.

Table 4 — Structure of protocol control word

Protocol control word bits 0x10 to 0x1F															
10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
					0/1	0/1	0/1								
Length indicator					1 = user memory encoded	1 = xpc exists	1 = ISO	ISO Application Family Identifier (AFI)							

Table 4 *(continued)*

Length indicator	1 = user memory encoded	1 = xpc exists	0 = GS1	GS1/EPC attribute bits

The structure is significant and relevant to this document as follows:

- A UII length field is encoded in memory locations 10_{HEX} to 14_{HEX}. The value of the length field should be calculated automatically as defined in ISO/IEC 18000-63.
- A user memory indicator (UMI) is held in location 15_{HEX}. This has the value '1' set automatically as defined in ISO/IEC 18000-63, if there is any encoding in MB 11. During a reading procedure, the value of the UMI bit is used to determine if optional data is encoded in MB 11.
- An extended protocol control indicator is stored in location 16_{HEX}. The function of this bit is beyond the scope of this document but, if used in a future revision, it would be calculated automatically as defined in ISO/IEC 18000-63.
- A numbering system identifier (NSI) is encoded in memory location 17_{HEX} to $1F_{HEX}$. This is in turn subdivided as follows:
  - Bit location 17_{HEX} shall be encoded with the value '1'.
  - Bit locations 18_{HEX} to $1F_{HEX}$ shall encode the AFI.
- The encoding of the Unique Item Identifier starts at bit location 20_{HEX}. Encoding and decoding needs to be invoked for complete 16-bit words. The value of UII length field (in memory locations 10_{HEX} to 14_{HEX}) is generated automatically as defined in ISO/IEC 18000-63.

7.3.5 Encoding in MB 01

7.3.5.1 General

MB 01 encodes the AFI and the UII. The encoding rules for these two components are defined in the following subclauses. Although shown separately, the encoding should be implemented in one action.

7.3.5.2 Encoding the AFI

The AFI is encoded as part of the protocol control word in bit locations 18_{HEX} to $1F_{HEX}$. It shall be preceded by a '1' in location 17_{HEX} to enable tags encoded to ISO rules to be distinguished from those encoded to GS1 EPC rules. The encoding is shown in Table 5.

Table 5 — Encoding the AFI in MB 01

Bit position	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
Bit value	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	0

In the absence of any more specific procedures for a creating air interface commands, the 16-bit string, as defined in Table 5, shall be used to construct the encoding of MB 01 bit positions 10_{HEX} to $1F_{HEX}$.

7.3.5.3 Encoding the UII

The Monomorphic-UII shall be encoded using the URN Code 40 encoding rules as defined in Annex D. The AFI declares the encoding scheme. In turn, this means that a DSFID does not need to be encoded in MB 01. Neither does the data need to be encoded using the data set structure (as defined in 7.3.11). The URN Code 40 encoding rules support variable length input without requiring a length to be encoded.

Any of the structures described in 6.2.4 may be encoded using these basic steps.

a) Because a Monomorphic-UII is used, replace the name within { } with the relevant value of the data element.

b) If there is a 'dot' character within the ISIL or the primary item identifier, then the encoding cannot proceed as an error has occurred, as identified in 6.2.

c) Remove the { } characters which were previously included to help with the illustrations.

d) Retain the 'dot' separators between the component parts.

e) Submit the resultant character string to the URN Code 40 encoder. The encoder takes as input a 3-character string and coverts it to a 16-bit string. The process is repeated until the encoding is completed. There are minor exceptions discussed below and in more detail in Annex D.

The resultant byte string is encoded from bit location 20_{HEX}. Because URN Code 40 encoding is always over 16-bit units, it is already aligned with the 16-bit word boundary of MB 01.

EXAMPLE Basic structure: {Owner institution (ISIL)}.{Primary item identifier}.{set information}

Step 1: {CH-000134-1}.{12345678}.{31}, indicating that the item is part of a set of three items all of which carry an RFID tag.

Steps 2 and 3: CH-000134-1.12345678.31

This 23-character string results in an encoding of 128 bits, or 8 words.

The URN Code 40 basic character set supports all the permitted characters in the ISIL except the solidus (or forward slash) '/'. If this is required as part of the ISIL, then the encoding of the preceding characters shall be completed using pad characters as necessary to align to a 16-bit boundary. The solidus has the code point $2F_{HEX}$ in the default character set. It is encoded as the double byte value $FC2F_{HEX}$ in URN Code 40. The encoding reverts to the basic character set.

A similar rule applies if the primary item identifier requires a character from the permitted character set of ISO/IEC 646 International Reference Version (IRV), but is not supported by the URN Code 40 basic character set. The lead byte of FC_{HEX} acts as a shift key to encode the character. Just like a shift key, it returns to the basic character set immediately after the character is encoded.

7.3.5.4 Rules for writing and locking MB 01

MB 01 does not support any form of selective locking, but the entire memory bank may be locked. There are no commands to determine if MB 01 is locked (see 7.3.1).

The procedure to lock MB 01 does require some precision in the sequence of commands to encode data on the tag. For example, the encoding in MB 11 needs to be started so that the UMI bit can be set automatically. Only after all of the relevant encoding has been done should any attempt be made to lock MB 01.

The structure of MB 01 and the locking functionality has some implications for this document.

— If the AFI is encoded prior to the encoding of the UII, MB 01 shall not be locked at this stage to avoid the tag being rendered useless.

— If any of the optional data elements are to be encoded in MB 11, at least the DSFID shall first be encoded to ensure that the user memory indicator (UMI) in location 15_{HEX} is correctly set.

— If no encoding of optional data elements is ever expected, then the encoding of all the data in MB 01 is completed and locked as defined in Annex C.

— As the UII is locked, then the procedure of toggling the AFI between two values for processing at the security gates cannot be invoked. The risks of corrupting the UII should be a paramount consideration.

NOTE There are other options for implementing a security procedure at the exit of a library. These are discussed in 9.2.

7.3.6 Relative-OID for the UII

As shown in Table 1, the UII has a Relative-OID value = 0. Depending on the encoding systems used, this may not be required because the structure string of constituent parts will be input. This Relative-OID could be more significant for decoding systems, particularly those that make use of more generic device architecture as shown in Figure 1.

7.3.7 Decoding and processing the Monomorphic-UII

Decoding the URN Code 40-byte string is achieved by taking each 16-bit word from the encoded Monomorphic-UII and converting it to characters.

- If the first byte has the value FA_{HEX} or lower, then the 16-bit string decodes using inverse rules to those for encoding and results in a three-character string.
- If the first byte is FC_{HEX}, then the next byte is interpreted as the ISO/IEC 646 International Reference Version (IRV) character.

Using the example in 7.3.5.3, the Monomorphic-UII is returned with the Relative-OID = 0 and character string:

CH-000134-1.12345678.31

If the UII includes the ISIL of the owner institution (which can be validated to the ISIL rules or by explicit matching to the ISIL of the library), the library may

- retain the Relative-OID = 0 and process the UII as a single entity, or
- process the ISIL component as Relative-OID = 3, and the primary item identifier as Relative-OID = 1.

If the UII includes only the primary item identifier, then it is processed as Relative-OID = 1.

If the final component of the UII comprises of two, four, or six numeric digits after the final 'dot', then this indicates that the set information (as defined in 6.6) has been encoded immediately after the primary item identifier. The set information can be a part of these UII structures:

- ISIL component, followed by the primary item identifier, followed by the set information; in which case, the library may
  - retain the Relative-OID = 0 and process the UII as a single entity, or
  - process the ISIL component as Relative-OID = 3, the primary item identifier as Relative-OID = 1, and the set information as Relative-OID = 4.
- primary item identifier, followed by the set information; in which case, the library should process the primary item identifier as Relative-OID = 1 and the set information as Relative-OID = 4.

If this final component is the letter 'S', it indicates that the item is part of a set, and that the primary item identifier is unique to the individual item. Where the set is declared with the letter 'S', it is necessary to decode the data element in MB 11 with the Relative-OID = 4 to obtain the details.

7.3.8 Use of GS1 EPC codes in MB 01

There will be circumstances where a properly structured GS1 EPC Serialised Global Trade Item Number (SGTIN) is already encoded on the RFID tag. This can be due to the fact that UHF RFID systems have been implemented prior to this document being published, or to the fact that the encoding has been carried out earlier in the supply chain.

It may be possible to use the GS1 EPC code as the UII if the library is capable of processing this through various aspects of the library's application. If not, and if MB 01 is not locked, then MB 01 with the GS1 EPC code shall be overwritten with the AFI and the UII as defined in 7.3.5.

If the tag carrying the GS1 EPC code in MB 01 has encoding capacity in MB 11, then the encoding shall be compliant with the procedures defined in 7.3.11.

7.3.9 Structure and use of MB 10

MB 10 (also known as TID memory) encodes information that identifies the manufacturer or designer of the integrated circuit and the model number. These provide some information about the tag capability.

ISO/IEC 18000-63 compliant integrated circuits that have been developed more recently can also include a serialised component in the TID.

Once the integrated circuit manufacturer has encoded the TID, it is generally locked and therefore, is in a read-only state.

7.3.10 Structure of MB 11

This memory bank contains the optional data elements and associated syntax. The first byte at memory address location 00_{HEX} to 07_{HEX} contains the encoded DSFID. The next byte beginning at memory address location 08_{HEX} can encode either

- the precursor of the first data set, or
- a pad byte if the DSFID is independently locked.

Encoding only the DSFID can be necessary when the intention to encode additional data elements is known, but for various operational reasons, those data elements are not known. In this case, it is essential to encode the DSFID so that the UMI bit in MB 01 can be set automatically (see 7.3.5.4).

As with the other memory banks, the basic *Lock* air interface protocol command only supports locking of the entire memory bank. Alternatively, if the tag and interrogator support the *BlockPermalock* command, then selective locking can be implemented. The value of the Access password determines how selective locking is implemented.

- If it is a non-zero value, then the command can be invoked only by using the correct value of the password.
- If it is zero value, then effectively the Access password is ignored and any interrogator can change the lock state.

There are no commands to determine if MB 11 is locked or selectively locked (see 7.3.1).

The structure of MB 11 and the locking functionality has some implications for this document.

- There is no discrete air interface command for encoding the DSFID. The DSFID is encoded either as part of a single encoding function for all of the data initially encoded in MB 11 or by using a write command for the first word. As the DSFID only occupies the first byte of the word, then if any data is to be encoded in the second byte, the entire word will need to be overwritten.
- If the tag supports selective locking using the *BlockPermalock* command, then the chip manufacturer determines the block size at the point of chip design. This block size can be one or more word in length.
- If the particular tag supports the *BlockPermalock* command, applying this command with only the DSFID encoded could consume a lot of memory, depending on the block size.
- If the particular tag only supports the *Lock* command, applying this command with only the DSFID encoded will render MB 11 useless for adding optional data elements.

The DSFID can be read without reading any other data encoded in MB 11 by invoking an air interface *Read* command and set this to only read the first word.

7.3.11 Encoding in MB 11

7.3.11.1 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, the following:

- data is 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 within MB 11. This allows optional data from the application data dictionary to be selected. It also enables variable length data to be encoded efficiently, 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.

The following are the requirements and recommendations for particular features.

- All interrogators shall support the encoding and decoding of MB 11.
- Any encoding and/or decoding software shall support all the data elements defined for MB 11.
- A library should use tags that support MB 11. In turn, the library may choose which data elements to encode in MB 11, and the sequence in which they are encoded.
- Not all interrogators support the air interface *BlockPermalock* command. If a library intends to make use of selective locking of data in MB 11, the interrogator shall support this feature.
- Not all tags support the air interface *BlockPermalock* command. If a library intends to make use of selective locking of data in MB 11, the tag shall support this feature.
- Where the tag supports the air interface *BlockPermalock* command, the parameters that define block size are defined by the manufacturer of the RFID chip. The block size can vary between manufacturers. Therefore, any encoding software shall declare the lock block size(s) that it supports.

  NOTE Although encoding software should support different lock block sizes, because ISO/IEC 18000-63 leaves this size undefined but under the control of the RFID chip manufacturer, any new RFID tag can have this set to a different value. The issue does not apply to decoding software, because whether a data element is locked or not is not relevant to the decoding process.

- The parameter values associated with lockable blocks in MB 11 have to 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 that the declaration of block size is essential.

7.3.11.2 Configuration of the DSFID

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

- access method;

— data format.

The data format is specified in 7.1.4 and the access method is defined in 7.1.7. These bit values are combined to create the appropriate DSFID byte value as shown in Table 6.

Table 6 — 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.

7.3.11.3 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. 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. 1 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 3-bit code which is included as part of the precursor (see 7.3.11.5). The full set of compaction schemes and their code is shown in Table 7.

Table 7 — 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

7.3.11.4 General rules for creating the encoded data set(s)

The encoding of the Relative-OID and data object on the RFID follows a particular sequential structure defined in ISO/IEC 15962. The next two subclauses define the basic rules that are relevant to this document.

NOTE ISO/IEC 15962 defines other rules, for example, encoding full object identifiers, which are not described here. If these are used to encode any data, a compliant decoder needs to be capable of decoding the object identifier and data.

7.3.11.5 Data set for Relative-OID value 1 to 14

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

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

— length of the compacted data object;

— compacted data object.

This structure is shown in Figure 6.

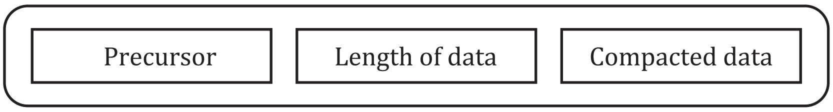

Figure 6 — ISO/IEC data set with Relative-OID values 1 to 14

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 8), and this reduces the amount of memory required for the encoding.

Table 8 — 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. Details of the use of the offset byte are given in 7.3.11.7.

7.3.11.6 Data set for OID value 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 4 bits of the precursor are set = 1111_2. This bit string signals that the Relative-OID has to be explicitly encoded as a separate component (a single byte) in the data set, as shown in Figure 7.

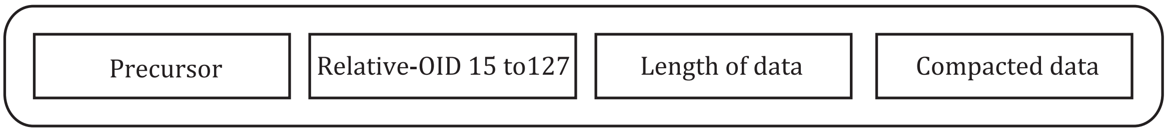

Figure 7 — ISO/IEC data set with Relative-OID values 15 to 127

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}.

7.3.11.7 Locking a data set

Based on the requirements of the application, any one or more data elements may 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 18000-63 air interface protocol permits locking by block. Account needs to be taken of the fact that selective locking can only be supported using the air interface *BlockPermalock* command, which not all RFID tags support. The block size is vendor specific as described in 7.3.10.

Generally, any data set that requires locking needs to be block aligned like the following.

— If only a single data set is to be locked, then its precursor is the first byte of the locked block. The precursor of the next **unlocked** data set also begins on the first byte of a block.

— If a contiguous sequence of data sets is to be locked, then the precursor of the first data set is the first byte of the locked block. Subsequent data sets that are contiguous are encoded immediately after the previous data set, until the end of the contiguous sequence of locked data sets. The precursor of the next **unlocked** 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 below, 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 permanently locked, it cannot be unlocked or erased, so the data set becomes permanently encoded on the RFID tag.

7.3.11.8 Logical memory for MB 11

Irrespective of whether one data set or multiple data sets are to be encoded in MB 11, 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 size of memory and the lock block size (if this is supported by the tag) 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 18000-63, but differ between each other within the options permitted by the air interface standard.

EXAMPLE 1 RFID tags can have different sizes of memory for MB 11.

EXAMPLE 2 The optional lock 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 *BlockPermalock* air interface commands.

When reading data from the RFID tag, the logical memory is populated word by word. Decoding of an RFID tag with a No-Directory access method is done in the sequence in which the Relative-OIDs are presented, 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.

8 RFID tag requirements

8.1 Air interface protocol

8.1.1 General

The air interface protocol shall be in accordance with ISO/IEC 18000-63, with the requirements specified in the following subclauses.

8.1.2 Memory parameters

The ISO/IEC 18000-63 tag has four memory banks as described in 7.3. The following parameters are relevant to the tag specification relevant to this document.

- MB 00 shall be provided with memory capacity for both the Kill password and Access password. The Access password is essential if the tag supports select locking using the *BlockPermalock* air interface command to enable selective locking, and/or to use the Access password as part of the security system.
- MB 01 is a mandatory requirement for ISO/IEC 18000-63 and shall have a minimum memory capacity to encode a UII of 128 bits. Depending on the structure of a library's UII (see 7.3.5.3), a tag with a larger capacity in MB 01 could be required.
- MB 10 is a mandatory requirement for ISO/IEC 18000-63. There is no requirement for the encoding by the IC manufacturer to be serialized, although this may be for some implementations.
- MB 11 should be provided to ensure the capability of encoding the optional data elements. The memory capacity required depends on the choice of data element to be encoded and the length of that encoding. A library that opts to have an RFID without MB 11 risks losing the full benefits of this document. This memory bank shall be capable of being selectively locked.

8.1.3 Declaring memory parameters

ISO/IEC 18000-63 defines a number of parameters that are fixed, such as the fact that the unit for reading and writing is a 16-bit word. However, many features and parameters are left to the choice of the IC manufacturer. In contrast to ISO/IEC 18000-3 Mode 1 tags (as used for ISO 28560-2), there is no air interface requirement to read a chip id as a basic part of the protocol to read the RFID tag. The ISO 18000-63 tag has, in MB 10, a code that identifies the IC manufacturer (or designer) and model. In fact, there are three permitted schemes.

Because of the variety of features that are possible among a set of ISO/IEC 18000-63 tags, a database of tag features is supported on the informational website (see Annex A). This can provide information that will help system designers support this document.

8.2 Required air interface commands

Table 9 identifies the mandatory and optional commands that are requirements for RFID for item management applications and, therefore, for this part of ISO 28560. Interrogators and tags claiming compliance with this document shall conform with the item management requirements provided in Table 9.

Table 9 — Required commands and their codes

Function	Command code (binary)	ISO/IEC 18000-63 basic types	Required for ISO/TS 28560-4
QueryRep	00	Mandatory	This is a RF level command and part of system setup.
ACK	01	Mandatory	This is a RF level command and part of system setup.
Query	1000	Mandatory	This is a RF level command and part of system setup.
QueryAdjust	1001	Mandatory	This is a RF level command and part of system setup.
Select	1010	Mandatory	This command is used to select tags by using the AFI for MB 01, and possibly the DSFID for MB 11.
Reserved	1011	N/A	
NAK	11000000	Mandatory	This is an RF level command and part of system setup.
Req_RN	11000001	Mandatory	This is an RF level command and used to communicate with a particular tag.
Read	11000010	Mandatory	This command is used to read words from the nominated memory bank, unless the memory area is read-locked (e.g. passwords).
Write	11000011	Mandatory	This command is used to write a single word to a nominated address in a nominated memory bank. It is not possible to write to a locked word, and this means that writing to MB 10 is impossible at the application level.
Kill	11011100	Mandatory	This command shall be used at the end of life of an item, possibly when sold by the library. The Kill password is delivered in two separate passes, the lead 16 bits followed by the final 16 bits. The commands have slightly different structures.
Lock	11000101	Mandatory	This command is used to lock or permalock the individual passwords, the entire MB 01, the entire MB 11.
Access	11000110	Optional	This command is used to restrict unauthorized access to encoding on the tag. The command shall be supported by interrogators and should be supported by the RFID tag.
BlockWrite	11000111	Optional	This command should be supported by interrogators and can be supported by the RFID tag.
BlockErase	11001000	Optional	This command should be supported by interrogators and can be supported by the RFID tag.

Table 9 *(continued)*

Function	Command code (binary)	ISO/IEC 18000-63 basic types	Required for ISO/TS 28560-4
BlockPermalock	11001001	Optional	This command is used to selectively lock the encoding on the tag or to read the permalock status from the tag. The command can be applied to MB 01 and MB 11. The command shall be supported by interrogators and should be supported by the RFID tag.

Although ISO/IEC 18000-63 indicates that the *Kill* command can be used to recommission a tag, this feature will be withdrawn from future versions of the air interface protocol.

NOTE No tag products are known to support this feature.

If used, the *BlockPermalock* command shall be applied selectively to the DSFID and or individual data elements in MB 11. The block size is specific to each tag model, and the parameters on the database (see 8.1.3) are required to invoke this command. Even when the tag supports the feature, it might not apply contiguously to all blocks in a memory bank, for example, a library may choose to leave a block unlocked so that the data elements can be modified.

8.3 Air interface conformance

Conformance testing of the system shall be in accordance with ISO/IEC 18047-6.

8.4 Performance

Various performance testing is possible according to the following standards:

- ISO/IEC 18046-1, for system performance;
- ISO/IEC 18046-2, for interrogator performance;
- ISO/IEC 18046-3, for tag performance.

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 all of MB 01 including the UII and AFI. It should also be considered for MB 11 for the DSFID and possibly for the ISIL code for the owner institution, if this is only encoded in MB 11. Locking the data sets of other data elements in MB 11 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.

Selective locking in MB 11 is only possible with an RFID tag that supports the *BlockPermalocking* air interface command.

9.2 Item security

9.2.1 General

Various approaches may 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 18000-63 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.

The use of the AFI, which is common for ISO/IEC 18000-3 Mode 1 tags, is deprecated as discussed in 9.2.5.

9.2.2 Use of the UII

MB 01 contains the UII, which is unique within the context of the library system. While it is possible to provide look-up through the LMS/ILS, the latency in searching the entire LMS has to be taken into account.

Another option is to create a temporary database look-up system just of the items that had been correctly checked out. To implement this option, the following steps are necessary.

a) In addition to using the UII for the LMS/ILS transactions, it is also written to the temporary security database look-up system for subsequent access by the security system.

b) The security system invokes the air interface *Select* command, which can effectively inventory all tags at the security exit.

c) Each captured UII needs to be compared with entries on the temporary security database look-up system.

   - If there is a match, then the item has been checked out and can be deleted from the check list.
   - If there is not a match, then there might be a security breach that requires further investigation.

9.2.3 Using passwords in MB 00

MB 00 contains encoding capacity for the Kill password. As the Kill password is only likely to be used at the end of life of a loan item, it can be used as part of a security system. The Access password, particularly if locked by invoking the air interface Lock command, should be used to avoid illicit reading of the Kill password. The following subclauses describe different ways that this password may be used. None requires the use of a database at the security gates.

NOTE Using a single Access password in a library and applying it to all tags might result in a single-point-of failure, because once this is known, it is possible to change the Kill password. While the most secure solution is to use unique passwords, very precise rules are required to achieve this and to avoid the rule being forgotten or even identified. It is the responsibility of solution providers to develop particular schemes.

9.2.3.1 Using a 'toggle' value

This method is an emulation of the two AFI solution used in other parts of ISO 28560. The 32-bit Kill password has one value assigned to items that are checked out and another 32-bit value assigned when the item is returned and defined as in stock. While the simplest implementation could be used to the existing single byte values, the minimum read and write unit for an ISO/IEC 18000-63 command is a 16-bit word.

While it is possible to leave one word of the Kill password as an all-zero value, this might present problems when the tag is eventually killed at the end of the life of the loan item. On this basis, a practical solution is to initially write a 32-bit Kill password with the first word defining the in-stock status and the second word some constant non-zero value for the library. The following procedures are applied.

a) On checkout, the first word of the Kill password is changed to the 'on loan' status value.

b) The security system invokes the air interface Read command for MB 00 word 00_{HEX}. This returns one of the two values.

c) This 16-bit value needs to be compared with the value of the 'on loan' status code.

- If there is a match, then the item has been correctly checked out.
- If there is not a match, then there might be a security breach that requires further investigation.

9.2.3.2 Using the date of circulation

This is similar to the method described in 9.2.3.1, except that the date of circulation is encoded as the first word of Kill password. The date should be calculated as the number of days since the start of some computer related epoch, for which a software rule might apply. The RFID system supplier may choose any start date for the epoch.

EXAMPLE If the epoch started on the 1 January 2006 and is encoded as a 16-bit integer, this date system can survive until the year 2185.

The following procedure is applied.

a) On checkout, the first word of the Kill password is changed to the current (issue) date.

b) The security system invokes the air interface Read command for MB 00 word 00$_{\text{HEX}}$. This returns a date as a 16-bit integer, counting from the start date of the epoch.

c) This needs to be compared with the current date set on the security system.

- If there is a match, then the item has been correctly checked out.
- If there is not a match, then there might be a security breach that requires further investigation.

This method does not need any change to the Kill password on return to the library. The previous issue date is simply over-written the next time the item is loaned.

9.2.3.3 Using an authentication algorithm

The first word of the Kill password can be a constant value or one that is accessible, like the date of purchase of the loan item. The second 16-bit word is an authentication code. It needs to be generated using an algorithm that takes the value of the UII and processes this using a constant encryption key. The following procedure is applied.

a) At the checkout, the 16-bit authentication code is calculated and written as the second word in the Kill password.

b) At the security gate, the second word of the Kill password is read from MB 00 and the UII read from MB 01. These are processed through what is effectively an authentication algorithm.

c) The authentication algorithm effectively produces

- a single instance that is true, in which case, the item has been correctly issued, or
- all other values are not authentic, which indicates that there might be a security breach that requires further investigation.

d) On return, the second word of the Kill password is overwritten with an all-zero 16-bit string.

9.2.3.4 Implications for data integrity, interlibrary loans, and end-of-life transactions

All three schemes retain a non-zero Kill password throughout the circulation cycle, so that the risk of the tag being killed by malicious or accidental action is reduced. This can be almost completely eliminated by the use of the Access password. However, there are some differences.

The toggle method only uses two different values of the Kill password, with these following implications:

- This type of password is easier to hack, especially if the basic rules are known.
- At the end of life, only one of two Kill passwords needs to be tried to render the tag unreadable.

The date of circulation method has multiple values of the Kill password, with these following implications:

- This type of password is difficult to hack, even if the basic rules are known. Each kill action requires knowledge of the date of the most recent circulation. For items that have been returned, this requires access to the LMS/ILS.
- At the end of life, the LMS/ILS needs to be accessed to construct the Kill password.

The authentication algorithm method has multiple values of the Kill password, with these following implications:

- This type of password is difficult to hack, even if the basic rules are known. Each kill action might require access to the LMS/ILS if the date of purchase is used. The algorithm is also essential.
- At the end of life, the algorithm is essential, and the LMS/ILS needs to be accessed to construct the Kill password if the date of purchase is used.

In each case, for ILL purposes, the entire Kill password needs to be set to a declared non-zero value that the receiving library resets to its own system. On return, this needs to be reset to the Owner library system.

9.2.4 Use of the unique tag ID

Many, but not all, ISO/IEC 18000-63 tags have a unique tag ID programmed by the integrated circuit manufacturer in MB 10, which is a non-volatile memory.

To implement the option, given that the RFID tag supports serialization in MB 10, these following steps are necessary.

a) MB 10 shall be read during the check-out procedure.

b) A transaction file shall be created for subsequent access by the security system;

c) The security system shall read MB 10 on all tags leaving the library.

   - If there is a match, then the item has been checked out and can be deleted from the check list.
   - If there is not a match, then there might be a security breach that requires further investigation.

This system is similar to reading the UII, but does require more specific air interface transactions.

9.2.5 Use of the AFI

The AFI encoded in MB 01 shall not be used as a security mechanism, because there are a number of system integrity risks associated with using a dual state AFI as is possible with ISO/IEC 18000-3 mode 1 tags, which include the following.

- Because the AFI is part of the Protocol Control word, other bits that are vital to the construction of the UII such as its length can be corrupted.
- The Protocol Control word also contains bits that signal functions of the tag, including the presence of encoding in MB 11. If this bit changes in value, then access could be lost to MB 11.
- MB 01 encodes the UII, which is critical to the library circulation functions. If this is corrupted, then the RFID is effectively useless.

For these reasons of system integrity, the AFI shall be permanently encoded.

9.2.6 Use of the EAS features

Electronic article surveillance (EAS) features have been added by some manufacturers to the ISO/IEC 18000-63 tag as a proprietary feature. As such, any proprietary EAS feature should not be used because interoperability between different EAS systems cannot be ensured.

9.3 Privacy issues

What constitutes privacy issues can be subject to national and regional legislation, combined with cultural perceptions. Therefore, this document does not define any methods for addressing privacy issues in terms of requirements. However, there is some general advice as the following:

— Where legislation and regulations are in place in the jurisdiction domain of a library, these shall be followed.

— Where standards for undertaking Privacy Impact Assessments are in place, these should be followed, and shall be followed if supported by legislation.

— Within the context of the two previous points, minimum use should be made of data elements that are specified as of concern. This does not mean that false assumptions should be made about the privacy risk of a data element. For example, in the past, the encoding of the ISBN has been of concern, whereas in some legislation the fact that the RFID tag has a unique chip ID and unique item identifier is of far greater concern.

— As more formal work is undertaken on privacy in other domains, information will be periodically provided on the informational website (see Annex A).

— The ISO/IEC 18000-63 tag (based on ISO/IEC 18000-63) has no particular features that can be considered as privacy enhancing technologies (PET), or similar techniques or tools. The next edition of ISO/IEC 18000-63 is expected to specify rules for restricted read access. If such a feature is considered suitable for this document, then a revision shall be necessary, but taking into account that not all tags and interrogators will be able to support the feature.

10 Implementation and migration

Annex E gives some encoding examples of data elements compliant with this document. Issues concerned with new implementations and migration issues from legacy RFID implementations to this document are discussed in Annex F.

11 Miscellaneous

Two-character code for the RFID Emblem defined in ISO/IEC 29160 is E* for UHF = 860 to 960 MHz.

Annex A
(informative)

Information about ISO 28560 RFID in libraries

A.1 Informational website

Additional information about RFID in libraries: http://biblstandard.dk/rfid hosted by Royal Danish Library.

A.2 Types of support information

At the time of publication of this document, two items of information have been identified. These are listed below, together with the URL of the continually updated material.

- 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 materials might be published in the future, and this will be publicized and made available from the URL in A.1.

Annex B
(informative)

Relevant ISO/IEC 15961-1 application commands

B.1 Write-Monomorphic-UII

The Write-Monomorphic-UII command is used to write the UII, including the AFI.

Part of the process is to compare the AFI input with that registered for the application on the ISO/IEC 15961-2 Data Constructs register. In the case of this document, this AFI is $C2_{HEX}$. If the AFI matches, the Object-Identifier is also checked for a match with that on the ISO/IEC 15961-2 Data Constructs register. A mismatch of either generates an appropriate error.

The Data Processor uses the explicitly defined compaction scheme associated with the AFI on the ISO/IEC 15961-2 Data Constructs register to encode the Monomorphic-UII. In the case of this document, this is URN Code 40.

The command supports two optional processes relevant to this document, which are the following:

- the lock argument, which is applied to the entire MB 01;
- the use of the **Access-Password**. If this is already encoded in MB 00, then before the UII can be encoded, the **Access-Password** in the command needs to match that already on the tag.

B.2 Write-Objects-Segmented-Memory-Tag

The Write-Objects-Segmented-Memory-Tag command is used to write data to a selected memory bank in a segmented memory tag. The command may be implemented to write initial data to the RFID tag or to add data to the tag. For this document, it applies only to MB 11.

The command supports the following optional processes relevant to this document:

- defining the DSFID to be encoded, and whether this is to be locked;
- the lock argument, which should be applied selectively to individual data elements;
- the use of the **Access-Password**. If this is already encoded in MB 00, then before any data elements can be encoded, the **Access-Password** in the command needs to match that already on the tag.

B.3 Write-Password-Segmented-Memory-Tag

The Write-Password-Segmented-Memory-Tag command does not require any encoding, other than a transfer using the device interface (e.g. as in ISO/IEC 24791-5) to the interrogator. If it is incorporated in the data processor, then the output interface will simply carry through the command arguments.

The following process arguments shall be supported:

- Password;
- Password-Type.

B.4 Inventory-ISO-UIImemory

The Inventory-ISO-UIImemory command is intended to return the contents of the UII memory from a number of segmented memory tags, given the expectation that an Object-Identifier for a non-EPC-Code is encoded. The response returns the content of the UII memory for all tags whose encoded bit string matches the arguments of the command. The relevant argument for this document is specific to the AFI.

The response returns to the UII from the tags in the read zone.

B.5 Read-Objects

The Read-Objects command is used to read one or more data objects from MB 11 on the RFID tag. It is supported by a Read-Type argument that enables only first objects (such as the OID-index) to be read, or to read one or more objects, or to read all objects. If the read type is Read-1st-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. 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-Words-Segmented-Memory-Tag

The Read-Words-Segmented-Memory-Tag command instructs the interrogator to read a contiguous sequence of words from one of the memory banks of a segmented memory RFID tag. This command can be used to extract encoded bytes from MB 01 or MB 11. It can also be useful for diagnostic purposes.

B.7 Read-Object-Identifiers

The Read-Object-Identifiers command reads all the object identifiers, but not the associated data objects from MB 11. 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.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 permalocked, it cannot be modified. If it is locked, then changing its value depends on what authorization is possible using the parameters in the BlockPermalock command. 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 Erase-Memory

The Erase-Memory command instructs the interrogator to set to zero the entire encoding in MB 01 or MB 11 on a specified RFID tag. If any blocks are locked, then a generic error 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.

Invoking the air interface BlockErase command is one way to achieve this more effectively if the interrogator and the tag support the command. This is a system implementation issue beyond the scope of this document.

B.11 Kill-Segmented-Memory-Tag

The Kill-Segmented-Memory-Tag command instructs the interrogator to apply appropriate air interface protocols to render the RFID tag unreadable in the future. The Kill-Password in the command shall match the Password encoded on the RFID tag.

For this document, this command is only relevant at the end of life of a loan item, or if the encoding on a tag has been corrupted and a new tag needs to be applied to a loan item.

Annex C
(informative)

Locking procedure for MB 01 with encoding in MB 11

The following procedure assumes that the RFID tag conforms to the recommended specification as defined in this document and has encoding capabilities in MB 11.

a) Write the DSFID in MB 11. If the input details are known, other data may be encoded at the same time.

b) Consider locking the DSFID and any data elements using the BlockPermalock air interface command, ensuring that the lock status of the first data set is taken into account. This might require the use of pad bytes. If the tag does not support selective locking using the BlockPermalock command, then locking MB 11 will make it impossible to make any future changes to the encoding.

NOTE The fact that the block size of the tag can vary between RFID chip manufacturers requires some consideration for the initial encoding to consider both the organization of the sequence of data elements and whether they are locked. As a library is likely to use the same brand and model of tag for a period of time, a rule for encoding and locking can be persistent for some time.

c) Create the bit string for the Protocol Control word starting at bit location 15 with {10111000010} to encode the UMI bit, the NSI bit and the AFI (see 7.3.4). This is the safer of two options defined in ISO/IEC 18000-63 because it supports tags that do not automatically generate the UMI bit. A tag that does will generally overwrite the lead '1' bit in the string with an automatically generated '1' bit.

d) Create the UII.

e) Append the UII (step 4) to the Protocol Control word bits (step 3) and write to MB 01. It might be necessary to pad with 5 bits {00000} to complete the structure of the Protocol Control word.

f) Lock MB 01.

WARNING — If the DSFID is not encoded before MB 01 (containing the AFI and UII), then using MB 11 becomes extremely difficult because the UMI bit in the protocol control word will indicate that it contains no encoding.

Annex D
(informative)

Monomorphic-UII and URN Code 40 encoding

D.1 Monomorphic-UII

The Monomorphic-UII encoding scheme is designed to achieve a simple encoding of a Unique Item Identifier when encoded in a memory area dedicated to this function, and where no additional data is encoded in the same memory area. All of the features can be self-declaring through the registration of the particular AFI code values under the rules of ISO/IEC 15961-2. The following conditions apply.

- The Monomorphic-UII is only capable of being encoded in a tag memory architecture that supports the separate encoding of a UII. For this document, the tag is one that conforms to ISO/IEC 18000-63.
- An AFI assigned to a particular Monomorphic-UII shall not support the encoding of any additional item-related data in the same memory area. If an item-related data is required for the application, then this shall use an entirely different AFI and Data-Format. For this document, the AFI is $C2_{HEX}$.
- The AFI fully defines all the arcs of the Object-Identifier for communications in the commands of ISO/IEC 15961 and in the ISO/IEC 24791 standards. For this document, the Object Identifier for the UII is 1,0,15961,6,0.
- Each AFI declares which of the encoding schemes, as defined in ISO/IEC 15962 is used. For this document, this is URN Code 40. As this encoding scheme is based on a sequence of 16-bit encoding units, there is no requirement to encode the length of the encoding. This is declared by the length bits in the protocol control word and 'calculated' by hardware rules for the ISO/IEC 18000-63 air interface protocol.
- The Monomorphic-UII does not require the encoding of a DSFID in MB 01. For this document, a DSFID is still required to be encoded in MB 11.

D.2 URN Code 40 encoding

This particular encoding is designed to support the encoding of a hierarchical URN with the appropriate separators between the hierarchical components. This encoding scheme provides a method to encode data compliant with the urn:oid namespace scheme and extended to cover the Unique Item Identifier, as used in the ISO RFID data protocol. Therefore, when the URN is decoded from the RFID tag, it is presented in a structure that is compatible with that required for resolving on the Internet.

NOTE Although this feature is not defined in this document, services can be developed either from reading the UII form the tag (if it contained the ISIL) or the Object Identifier can be constructed by combining the ISIL for the library with the encoded primary item identifier.

The UII shall be encoded using the URN Code 40 encoding defined in the following subclauses.

D.2.1 Basic character set

The encoding process takes a string of three data characters (from Table D.1) and compacts these into two bytes. The PAD character is used in the final string where there are fewer than three data characters.

Table D.1 — URN Code 40 character set

Graphic symbol	Name	HEX code	8-bit code	URN Code 40 (decimal)
	PAD			0
A	Capital letter A	41	01000001	1
B	Capital letter B	42	01000010	2
C	Capital letter C	43	01000011	3
D	Capital letter D	44	01000100	4
E	Capital letter E	45	01000101	5
F	Capital letter F	46	01000110	6
G	Capital letter G	47	01000111	7
H	Capital letter H	48	01001000	8
I	Capital letter I	49	01001001	9
J	Capital letter J	4A	01001010	10
K	Capital letter K	4B	01001011	11
L	Capital letter L	4C	01001100	12
M	Capital letter M	4D	01001101	13
N	Capital letter N	4E	01001110	14
O	Capital letter O	4F	01001111	15
P	Capital letter P	50	01010000	16
Q	Capital letter Q	51	01010001	17
R	Capital letter R	52	01010010	18
S	Capital letter S	53	01010011	19
T	Capital letter T	54	01010100	20
U	Capital letter U	55	01010101	21
V	Capital letter V	56	01010110	22
W	Capital letter W	57	01010111	23
X	Capital letter X	58	01011000	24
Y	Capital letter Y	59	01011001	25
Z	Capital letter Z	5A	01011011	26
–	Hyphen-Minus	2D	00101101	27
.	Full stop	2E	00101110	28
:	Colon	3A	00101110	29
0	Digit zero	30	00110000	30
1	Digit one	31	00110001	31
2	Digit two	32	00110010	32
3	Digit three	33	00110011	33
4	Digit four	34	00110100	34
5	Digit five	35	00110101	35
6	Digit six	36	00110110	36
7	Digit seven	37	00110111	37
8	Digit eight	38	00111000	38
9	Digit nine	39	00111001	39

Three URN Code 40 values (the last column in the table) are encoded as a 16-bit value (msb first). Three URN Code 40 values (C_1, C_2, C_3) shall be encoded as:

$$(1600*C_1) + (40*C_2) + C_3 + 1$$

This process produces a value in the range from 1 to 64000, which is converted to hexadecimal in the range 0001_{HEX} to $FA00_{HEX}$.

D.2.2 Extended encoding

The table-driven encoding, as discussed above, only uses the double byte values up to $FA00_{HEX}$. As the basic encoding is always of a pair of bytes, the leader byte can never have a value FB_{HEX} to FF_{HEX} in the table-driven solution. If these byte values are used as syntax to encode different encoding, then the capability of the scheme can be extended in the following ways:

— FB to signal long string integer encoding. If the number of contiguous decimal digits is equal to or greater than nine, then this scheme is as efficient as or more efficient than the table-driven encoding scheme. It may be introduced on any 2-byte boundary of the table-driven compaction. The encoding consists of three components, as shown in Table D.2.

Table D.2 — Structure for encoding long numeric strings

Lead byte	Second byte	Subsequent bytes
FB	1st hex char = number of decimal digits 9 to 24 2nd hex char = number of encoded bytes 4 to 19	Hex bytes = integer value of numeric string (msb first)

The value of the second byte is structured as follows.

— The first hex character {0 to F} represents the number of decimal digits 9 to 24 in the input string, including leading zeros. A longer string cannot be encoded.

— The second hex character {0 to F} represents the number of bytes 4 to 19 used to encode the converted integer.

The scheme supports the encoding of leading decimal zeros.

— FC to signal an ISO/IEC 646 character: The next single byte value represents any of the ISO/IEC 646 characters not directly encodable from the base table.

— FD to signal a double byte UTF-8: The next two bytes represent any UTF-8 double-byte encoding transformed from ISO/IEC 10646.

NOTE To meet the requirements of IETF *RFC2141 URN Syntax*, 2.4, all ISO/IEC 8859-1 characters in the range $7F_{HEX}$ to FF_{HEX} (note not 80_{HEX} to FF_{HEX}) are transformed to their UTF-8 code points. All the UTF-8 encoding on the tag can be mapped to the standard Internet resolver syntactical format of:

"%" < hex > < hex >

— FE to signal a triple byte UTF-8: The next three bytes represent any UTF-8 triple-byte encoding transformed from ISO/IEC 10646.

— FF (currently RFU in ISO/IEC 15962).

D.2.3 Encoding example

Variable length data has a 'dot' separator between each constituent part.

NOTE As indicated in 6.2, this means that a 'dot' (also known as a full stop or period) cannot be part of the ISIL or part of the primary item identifier.

Using the example in 7.3.5.3, the encoding can be shown as:

CH-000134-1.12345678.31

This results in a three tiered structure, where the string {CH-000134-1} is associated with the highest level namespace resolver.

The encoding process begins by taking the first three characters {CH-}. These are converted to their URN Code 40 table values and then compacted as follows:

$$1600*3 + 40*8 + 27 + 1 = 5148_{10} = 141C_{HEX}$$

The process is repeated for the next three characters {000}

$$1600*30 + 40*30 + 30 + 1 = 49231_{10} = C04F_{HEX}$$

The process continues for a total of eight cycles until the last characters {31}, but as the encoder requires three characters, the PAD is appended to the two data characters: {31} PAD. The encoding process is:

$$1600*33 + 40*31 + 0 + 1 = 54041_{10} = D319_{HEX}$$

This results in an encoded string of eight 16-bit words.

Input data:	CH-	000	134	–1.	123	456	78.	31
Encoded bytes:	141C	C04F	C70B	ADB5	ADB5	DA1D	ED4D	D319

Annex E
(informative)

Encoding examples

E.1 General considerations

This annex shows the encoding of a hypothetical set of data elements compliant with this document. The example excludes features that require selective locking of some data elements. This is because of the different block lock sizes, however, advice is provided about how to address this.

E.2 Encoding the UII

This is as defined in Annex D, with the UII preceded by the AFI.

E.3 Encoding optional data in MB 11

E.3.1 General

The processes are presented in a manner to help the reader understand how input data is 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.

E.3.2 Input assumptions for the tag

The DSFID is required to be encoded as the first byte of MB 11. On the assumption that selective locking of individual data elements is not included in this example, the next byte is part of the first data set.

E.3.3 Input data

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

Table E.1 — Example data elements

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

Although the ISIL and set information may be encoded as part of the UII, for this example, they are encoded in MB 11.

E.3.4 Encoding the data elements

E.3.4.1 General

As all of the Relative-OID values are in the range from 1 to 14; the Relative-OID is directly encoded in the precursor (see 7.3.11.5). This means that each data set consists of the precursor, the length of the encoded data, the compacted data.

E.3.4.2 OID index

The OID index (see 6.4) 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 three. 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 one byte and the compaction code is "000".

The complete encoding of this data set is shown in Table E.2.

Table E.2 — Encoded byte string for the OID index

Precursor	Offset	Length of compacted data	Compacted data
00000010 = 02		01	D0

E.3.4.3 Set information

In this example, this is item three 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 two bytes and the compaction code is "001".

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

Table E.3 — Encoded byte string adding the set information

Precursor	Offset	Length of compacted data	Compacted data
00000010 = 02		01	D0
00010100 = 14		02	04 B3

E.3.4.4 Shelf locationf

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 6-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 6-bit compaction scheme.

The complete encoding of this data set is shown in Table E.4.

Table E.4 — Encoded byte string adding the shelf location

Precursor	Offset	Length of compacted data	Compacted data
00000010 = 02		01	D0
00010100 = 14		02	04 B3
01000110 = 46		07	44 1C B6 E2 E3 35 D6

E.3.4.5 Owner institution (ISIL)

This example is alphanumeric, with hyphens:

US-InU-Mu

To encode all the nine characters (which include lower case characters), the 7-bit code compaction scheme is automatically selected by the encoding software. The encoded byte string is:

AB 4D 6C 9D D5 56 CD EB

The encoding is shown in Table E.5.

Table E.5 — Encoded byte string adding the ISIL

Precursor	Offset	Length of compacted data	Compacted data
00000010 = 02		01	D0
00010100 = 14		02	04 B3
01000110 = 46		07	44 1C B6 E2 E3 35 D6
01010011 = 53		08	AB 4D 6C 9D D5 56 CD EB

E.3.5 Complete encoding

The encoded data sets are preceded by the DSFID, which is 06_{HEX}. This results in this string of bytes presented as 16-bit words:

0602 01D0 1402 04B3 4607 441C B6E2 E335 D653 08AB 4D6C 9DD5 56CD EB

The final word is padded with the byte 00_{HEX} to align to a word boundary, resulting in an encoding of 14 words, requiring a tag with at least 224 bits of memory in MB 11.

E.3.6 Locking implications

If the DSFID or any of the data elements requires selective locking, then an offset combined with pad bytes might be required to end the encoding on a block boundary. Assume that the encoding in this example resulted in the following:

- DSFID being locked;
- content parameter (OID index) being unlocked;
- other three data elements being locked.

This would result in the DSFID being followed by pad bytes 80_{HEX} until the end of the block. The content parameter would then begin at a boundary and as its data set is only three bytes long, then this too would need to be block aligned. A better approach would be to extend the encoding capacity of the OID index to support additional data elements should they be encoded on the tag. The final three data sets would be encoded as follows.

a) The first two would be encoded without an offset.

b) The third data set might require the use of the offset to end on a block boundary.

Alternative approaches include the re-arrangement of the sequence of data elements or locking the entire group if no other data is expected to be encoded in MB 11.

Annex F
(informative)

Implementation and migration

F.1 New RFID implementations

Any library that has yet to introduce RFID should give consideration to the use of this document. Consideration needs to be given to the implications of any current installed base using ISO/IEC 18000-3 Mode 1 tags and/or ISO/IEC 18000-63 tags within the same country or locality.

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.

There is already an increasing choice of RFID devices and tags that conform to ISO/IEC 18000-63. The performance and conformance requirements (see Clause 4) should be used to evaluate any new device or tag component.

F.2 Legacy RFID implementations using ISO/IEC 18000-63 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 18000-63. 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 B. 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 18000-63 or use with this document.

It is almost certain even with the same air interface protocol, that the data model and the encoding rules for the present implementation will be different from data model and encoding rules as defined in this document. The following options exist.

- 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.
- If any of the data is locked on the existing tag, the procedure to erase and overwrite is not possible. 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.
- A special case exists (probably more in the future than at the time of initial publication of this document) of the RFID tag being provided with source coding from the publisher or book distributor.

If the tag is encoded with an Serialized Global Trade Item Number (SGTIN) to the GS1 rules defined in that organization's Tag Data Standard, then these procedures should be considered:

- The SGTIN may be used 'as-is' for the UII, as long as the library is able to use this as part of the LMS/ILS, or an interface is created to achieve the mapping from SGTIN to primary item identifier. The AFI shall not be encoded in MB 01, because any decoding rules will be completely corrupted.
- If MB 11 is available, the encoding of the DSFID and optional data as defined in this document should be encoded.
- The use of passwords, locking, and selective locking should be used.

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 will be a requirement to maintain in parallel both the legacy system and the system compatible with this document.

F.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 18000-63, there are still possibilities for migration.

It is unlikely that the RFID tag used in the legacy system operates at 860 MHz to 960 MHz, but there might be some rare cases. Effectively, what is required is to introduce a system that supports multiple air interface protocols operating in the UHF spectrum. 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 F.2 is generally applicable.

If the existing RFID system is operating at a different frequency (e.g. the commonly used 13,56 MHz), then a significantly more complex challenge exists with radio frequency. RFID systems that operate at frequencies other than 860 MHz to 960 MHz have significantly different operational features. 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 codes*

[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: General procedures and top arcs of the international object identifier tree — Part 1*

[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 29160, *Information technology — Radio frequency identification for item management — RFID Emblem*

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

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

ICS 35.040.50; 35.240.30

Price based on 54 pages