International Post Corporation

IPC RFID STANDARD FOR TEST LETTERS USING THE ISO/IEC 18000-63 PROTOCOL

Version 1.0

22 February 2017

IPC RFID standard for: Test letters using the ISO/IEC 18000-63 protocol

Version 1.0

Contents

Introduction

This IPC test letter standard addresses the encoding rules to use ISO/IEC 18000-63 RFID technology for a quality of service monitoring system. It is intended to operate in parallel with a pre-existing system using active RFID tags.

Over time it is intended to replace the current active tag implementation, but still retain backward compatibility with it. Details of the active tag can be obtained from rfid@ipc.be.

As such the monitoring service will provide increased opportunities for integration with other IPC RFID standards that are based on ISO/IEC 18000-63 and the encoding rules defined in ISO/IEC 15962. This will enable enhanced support for resource sharing between postal services through greater interoperability of RFID tags and equipment.

The encoding structures that have been adopted for this IPC test letter standard have the potential to be used in a fully interoperable manner for other applications in future.

1 Scope

This IPC standard defines rules for encoding the identifiers of test letters in radio frequency identification (RFID) passive tags. The tags and other artefacts shall comply with ISO/IEC 18000-63 (previously known as ISO/IEC 18000-6 Type C) operating in the UHF frequency. The encoding rules are based on ISO/IEC 15962, which uses an object identifier structure to identify those elements. The current edition of this IPC test letter standard defines the rules for encoding a Unique Item Identifier in a specific Memory Bank known as MB 01.

Rules are also defined for efficient selection of test letters using criteria that can be implemented in the RFID interrogator.

Although the encoding on the ISO/IEC 18000-63 RFID tag is different from the encoding of the RFID active tag, a solution is provided for integration of both technologies. This allows RFID passive tag data capture and active tag data capture to be interoperable and to work concurrently in the same system.

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

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*

3 Terms and definitions

3.1
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

3.2
AFI
application family identifier
mechanism used in the data protocol and the **air interface protocol** 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 For this IPC test letter standard, the term is only relevant to Memory Bank 01, containing the data elements comprising the UII

3.3
arc
specific branch of an object identifier tree, with new arcs added as required to define a particular object

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

3.4
data format
mechanism used in the data protocol to identify how **object identifiers** are encoded on the RFID tag, and (where possible) identify a particular data dictionary for the set of relevant object identifiers for that application

3.5
EPCIS
Electronic Product Code Information Services
a GS1 standard for creating and sharing event data, both within and across enterprises, to enable users to gain a shared view of physical or digital objects within a relevant business context

3.6
MB
memory bank
designated name of a **segmented memory structure**

NOTE For the ISO/IEC 18000-63 tag the memory banks are: 00, 01, 10, and 11 using binary notation

3.7
logical memory
array of contiguous bytes of memory acting as a common software representation of the RFID tag memory accessible by an application and to which the object identifiers and data objects are mapped in bytes

3.8
object identifier
value (distinguishable from all other such values), which is associated with an object

3.9
segmented memory structure
memory storage that is separated into separate elements and requires multiple addressing elements for access

NOTE This has the same meaning as partitioned memory, a term used in some documents.

3.10
UII
unique item identifier
encodable data when combined with an object identifier prefix that renders the combination unique within the rules of the application domain

4 Symbols and abbreviations

IEC International Electrotechnical Commission
IPC International Post Corporation
ISO International Organization for Standardization
MHz Mega Hertz
RFID Radio Frequency Identification
UHF Ultra High Frequency
NOTE For RFID this is 860 MHz to 960MHz
UPU Universal Postal Union
URN Uniform Resource Name

5 RFID technology requirements

5.1 RFID air interface protocol

The air interface for compliant RFID tags and interrogators is specified in ISO/IEC 18000-63. There are different national and regional radio regulations for the use of RFID within the UHF frequency spectrum. It is essential to comply with such regulations, as follows:

- To meet with international requirements RFID tags should be able to operate between 860 MHz and 960 MHz, but shall comply with the national or regional requirements.
- RFID interrogators, or readers, shall operate at the nationally or regionally prescribed frequency within the 860 MHz to 960 MHz range. As a general guide:
  - Europe operates at the lower end: 865 MHz to 868 MHz
  - North America operates in the mid range: 902 MHz to 928 MHz
  - Japan operates at the upper end: 952 MHz to 958 MHz

More precise details are provided at http://www.gs1.org/docs/epcglobal/UHF_Regulations.pdf.

5.2 RFID tag

5.2.1 General tag features

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. Using binary notation, the memory banks (MBs) are:

00 – for passwords

01 – for the unique item identifier

10 – for tag identification, which can include serialisation

11 – for additional user data, which in the case of this IPC test letter standard is not required. If an RFID tag has this memory, MB 11 shall not contain encoding.

Memory is organised in a 16-bit word unit 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.

5.2.2 RFID tag memory parameter requirements

The ISO/IEC 18000-63 tag has four memory banks as described above. The following parameters are relevant to the tag specification relevant to this IPC test letter standard:

- MB 00 shall be provided with memory capacity for the Kill password. This may be used to ensure that the tag is not rendered unreadable. The Access password is not required for this IPC test letter standard.
- MB 01 is a mandatory requirement for ISO/IEC 18000-63 and shall have a minimum memory capacity to encode a UII of 96 bits.
- MB 10 is a mandatory requirement for ISO/IEC 18000-63. There is no requirement for the encoding by the IC manufacturer to be serialised, although this may be for some implementations.

— MB 11 is not required for this IPC test letter standard. If a tag does support MB 11, it shall contain no data to be compliant with this IPC standard.

5.2.3 Declaring the 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. There is no air interface requirement to read a chip id as a basic part of the protocol to select and read the RFID tag. The 18000-63 tag has, in MB 10, a code that identifies the IC manufacturer (or designer) and model.

The memory requirements for this Test Letter standard are defined in Annex A.1. To achieve interoperability in postal operations, IPC has adopted a set of test methods for the performance requirements for passive UHF RFID tags to qualify for postal operations (see 5.6 and Annex **Error! Reference source not found.**).

5.3 RFID interrogator (RFID reader)

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 as defined in 5.5, 5.7 and 5.8. This means that any one manufacturer's reading/writing equipment shall be able to read or write to any other manufacturer's RFID tags, and that any manufacturer's RFID tags shall be able to be read and/or programmed by any other manufacturer's reader/writer.

5.4 Required air interface commands

Table 1 identifies the mandatory and optional commands that are requirements for RFID for item management applications and therefore for this IPC test letter standard. Interrogators and tags claiming compliance with this standard shall comply with the item management requirements provided in the table.

Table 1 - Required commands and their codes

Function	Command code (binary)	ISO/IEC 18000-63 basic type	Required for this IPC standard
QueryRep	00	Mandatory	This is a RF level command and part of system set up.
ACK	01	Mandatory	This is a RF level command and part of system set up.
Query	1000	Mandatory	This is a RF level command and part of system set up.
QueryAdjust	1001	Mandatory	This is a RF level command and part of system set up.
Select	1010	Mandatory	This command is used to select tags by using the AFI for MB 01. It is also used in the IPC standard for an interrogator level *rapid reading* procedure (see Clause 10)
Reserved	1011	N/A	
NAK	11000000	Mandatory	This is a RF level command and part of system set up.
Req_RN	11000001	Mandatory	This is a RF level command and used to communicate with a particular tag.
Read	11000010	Mandatory	This command is used to read 16-bit 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 may be used in this IPC test letter standard to help avoid the UII being deleted. To support other IPC standards, the command shall be supported by interrogators.
Lock	11000101	Mandatory	This command is use to lock or permalock the individual passwords, or the entire MB 01.
Access	11000110	Optional	This command is not required for tags to support this IPC standard. The command shall be supported by interrogators to enable interoperability with other IPC standards.
BlockWrite	11000111	Optional	This command should be supported by interrogators, and may be supported by the RFID tag.
BlockErase	11001000	Optional	This command should be supported by interrogators, and may be supported by the RFID tag.
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, but is not required for RFID tags for this IPC test letter standard.

NOTE 1 Although the Access command is not required for this IPC test letter standard, it might be required for other applications. Therefore having the Interrogator support the command will ensure future interoperability and not require additional investment.

Although ISO/IEC 18000-63:2013 indicates that the *Kill* command can be used to re-commission a tag, this feature will be withdrawn from later versions of the air interface protocol.

NOTE 2 No tag products are known to support this feature.

5.5 Air interface conformance

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

5.6 Tag performance

Where there are requirements to test tag performance, this shall be done in accordance with ISO/IEC 18046-3. Additionally tags shall comply with the IPC set of test methods for passive UHF RFID tags (see Annex A.2).

5.7 Interrogator performance

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

5.8 System performance

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

5.9 RFID data protocol

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 MB 01 shall comply with the ISO/IEC 15962 rules for a Monomorphic-UII and specifically the URN Code 40 rules. Encoding in MB 01 is mandatory with the rules as defined in 8.4.3.

— MB 00 is intended for passwords. A 32-bit Kill password may be encoded during the encoding process, using the relevant air interface command. This can protect the tag from accidental or malicious destruction of its prime function. The Access password is not required for the IPC test letter standard.

— No encoding is possible in MB 10.

— No encoding is required in MB 11.

6 Data Protocol

6.1 Data protocol overview

The data shall be written to, and read from, the RFID tag using facilities functionally equivalent to the commands and responses defined in ISO/IEC 15961-1. 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.

6.2 Data constructs

6.2.1 Overview

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 6.2.2 and 6.2.3, together with their particular code values that have been assigned by the ISO/IEC 15961 Registration Authority for use by IPC.

6.2.2 AFI

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

The AFI value **A0**$_{HEX}$ has been assigned under the registration of ISO/IEC 15961-2 explicitly for use for IPC standards. This distinguishes postal 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 postal item and confusing the encoded content with data for its own application. It also enables a postal system to ignore items that carry a different AFI code or no AFI code (such as a GS1 EPC product code), possibly from a domain of a postal client (e.g. any content within a postal item).

The AFI is encoded in MB 01 (see 8.4.2). For IPC standards, the AFI declares that the UII that is also encoded in MB 01 is a Monomorphic-UII.

NOTE Unlike other ISO/IEC 15962 encoding schemes, Monomorphic-UIIs do not require the DSFID and some syntax to be encoded. All the requirements are declared by the AFI.

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.

6.2.3 Object identifier for postal 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 postal system, but between all domains. The object identifier may be split into two component parts. The Relative-OID, as defined in 6.3, 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 IPC standards is:

1.0.15961.14.

For all object identifiers specified in this IPC test letter standard, only the Relative-OID will need to be encoded.

6.3 The URN Structure

The Uniform Resource Name provides a means for extending the use of RFID beyond the base data capture. It provides a means to use:

- the Internet to enable searches from any computer with the appropriate browser rules,
- various layers of RFID communication standards from the device interface to the application and data exchange layers.

The generic URN structure for IPC is:

urn:oid:1.0.15961.14.{IPCApplicationType}.{UniqueID}

1.0.15961.14 is part of the registration with ISO, and is not encoded in the RFID tag, but declared by the AFI. This is called the Root-OID, and ensures that any IPC encoding in RFID tags and with the subsequent processing remains unambiguous.

The Relative-OID arc, value **B**, is assigned by IPC to distinguish RFID tags that encode the test letters from any other IPC RFID application standard. The arc, value **B**, is re-created by the RFID decoding process.

The specific URN structure for this IPC RFID test letter standard is:

urn:oid:1.0.15961.14.B.{UserType}{IssuerCode}{SerialNumber}

There are no dot separators within the data construction **{UserType}{IssuerCode}{SerialNumber}**, and this shall be encoded as a contiguous character string comprising:

— a single character UserType in the range A to Z and 0 to 9;

— a fixed length 3-digit IssuerCode, therefore always including any leading zeros;

— a fixed length 8-digit SerialNumber, therefore always including any leading zeros.

IPC shall control the assignment of UserType and IssuerCodes. Furthermore once assigned, an IssuerCode **shall not** be used in conjunction with another UserType.

NOTE The structure of the {IssuerCode}{SerialNumber} is compatible with existing IPC active tag data.

A postal operation may retain the UII in this format and add **1.0.15961.14.** as a prefix, or extract the test letter code depending on the business operation. Retaining the full OID structure, comprising of all arcs is useful where a system needs to distinguish between different OID structures or use resolver systems and other URN based systems, including the EPCIS. Extracting the test letter code achieves interoperability with the pre-existing active tag technology, although some additional processing is required (see 9.3.5). Both approaches may be used in the same operation to meet particular system requirements.

7 Unique item identifier (UII)

The unique item identifier (UII) is the mandatory data element to be encoded in Memory Bank 01 of an ISO/IEC 18000-63 RFID tag, which has 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. Specifically, the encoding shall comply with the URN Code 40 encoding rules as defined in ISO/IEC 15962.

NOTE 1 The Relative-OID in the UII is part of the data payload and therefore does not need to be encoded separately, nor is a DSFID or precursor required for MB 01. However, these features are required for encoding in MB 11 in other IPC standards.

The UII for this IPC test letter standard shall comprise these components:

— the IPC ApplicationType for test letters: the letter **B**,

— a ‘dot’ separator (the ‘dot’ is also known as a ‘full stop’ or ‘period’, ISO/IEC 8859-1 code point $2E_{HEX}$),

— the UserType, from a table of values provided by IPC,

— the IssuerCode, from a table of values provided by IPC,

— the SerialNumber.

NOTE 2 The ApplicationType is a mechanism that IPC can use to address other RFID applications and maintain full interoperability with this test letter standard.

The UII shall be locked to prevent various forms of digital vandalism and to ensure proof of source from a particular postal operator.

8 ISO/IEC 15962 encoding rules

8.1 General

The memory of an ISO/IEC 18000-63 tag is divided into four memory banks as defined in 5.2. 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 organised 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."

The logical memory is the software equivalent of the structure of the memory on the RFID tag itself. It is a mechanism used in ISO/IEC 15962 to represent all the encoding for a tag, including processes that need to be implemented for locking or selectively locking data. Once structured, the content of the logical memory can be passed to air interface protocol commands as the data 'payload' or to invoke other actions, like locking.

The following clauses identify the structure and rules as applicable for this IPC test letter standard.

8.2 Structure of MB 00

8.2.1 Supported passwords

This memory bank is used to store passwords. The 32-bit Kill password is stored at locations 00_{HEX} to $1F_{HEX}$. 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 $3F_{HEX}$. 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.

8.2.2 Kill password

A non-zero Kill password may be encoded. The value of the Kill password shall be any non-zero value selected by the organisation encoding the tag. By encoding a non-zero Kill password, the encoding on the tag can only be rendered unreadable by invoking the relevant password.

8.2.3 Access password

The Access password is not required for this IPC test letter standard.

8.3 Structure of MB 01

This memory bank contains the UII and associated syntax. The first word at memory address location 00_{HEX} to $0F_{HEX}$ 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 IPC test letter standard. The second word contains a protocol control word at memory locations 10_{HEX} to $1F_{HEX}$ as shown in Table 2, which shows the encoding for this IPC test letter standard in the last row.

Table 2 - Structure of Protocol Control Word

Protocol Control Word bits 10_{HEX} to $1F_{HEX}$															
Length indicator					UMI	XPC	NSI	ISO Application Family Identifier (AFI)							
10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
					0 = not used 1 = encoded	0 (N/A)	1 (ISO)	1	0	1	0	0	0	0	0

The structure is significant and relevant to this IPC test letter standard as follows:

- A UII length field is encoded in memory locations 10_{HEX} to 14_{HEX}. It represents the length of the encoding for UII. The value of the length field should be calculated automatically as defined in ISO/IEC 18000-63. The value of the length indicator will be:
  - **00100_2** for some of the lower value IssuerCodes,
  - **00101_2** for the remaining values of the IssuerCode.
- A user memory indicator (UMI) is held in location 15_{HEX}. Although MB 11 has no encoding for this IPC test letter standard, the UMI can have one of these two values:
  - **1** if the tag manufacturer sets this to indicated that MB 11 is present on the tag in accordance with an option introduced in the ISO/IEC 18000-63:2015 edition.
  - **0** if it is calculated directly by the tag, or alternatively by the interrogator to indicate that there is no encoding in MB 11 in accordance with an option retained in the ISO/IEC 18000-63:2015 edition from earlier editions.

  Annex A.1 provides information of the capability of different ISO/IEC 18000-63 tags compliant with IPC RFID standards.
- An extended protocol control indicator (XPC) is stored in location 16_{HEX}. The function of this bit is beyond the scope of this IPC test letter standard. If used in an IPC standard in future, it would be calculated automatically as defined in ISO/IEC 18000-63.
- A numbering system identifier (NSI) is encoded in memory location 17_{HEX}. This shall be encoded with the value '1' to indicate that the following eight bits are the AFI.
- Bit locations 18_{HEX} to $1F_{HEX}$ shall encode the AFI with the bit values **10100000_2.**

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 the UII length field (in memory locations 10_{HEX} to 14_{HEX}) is generated automatically as defined in ISO/IEC 18000-63.

8.4 Encoding in MB 01

8.4.1 Components

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

8.4.2 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.

In the absence of any more specific procedures for a creating air interface commands, the 16-bit string defined in Table 2, shall be used to construct the encoding of MB 01 bit positions 10_{HEX} to $1F_{HEX}$.

8.4.3 Encoding the UII

The Monomorphic-UII shall be encoded using the URN Code 40 encoding using these two rules, which are applied automatically by a compliant URN Code 40 encoder:

— Basic encoding from a table, as defined in Annex B.2.1.

— An integer encoding applied to a long numeric string, which is between 9 and 24 digits long. This is defined in Annex B.2.2.

The AFI declares the encoding scheme. The URN Code 40 encoding rules support variable length input without requiring a length to be encoded. These are the encoding steps:

1. Because a Monomorphic-UII is used, the encoder input is B.{UserType}{IssuerCode}{SerialNumber}. The resultant character string is submitted to the URN Code 40 encoder.

2. The encoder takes as input the first 3-character string **B.{UserType}.** Using the values from the basic encoding table and the formula defined in Annex B.2.1, converts it to a 16-bit string. This rule is applied whether the {UserType} is an alphabetic character or a numeric digit.

3. The {IssuerCode}{SerialNumber} is always 11 digits long, which includes any leading zeros in the IssuerCode or the SerialNumber (see 6.3). As this begins on a 16-bit word boundary, this invokes integer encoding.

    a. The numeric string is input to the encoding rule and converted to an integer.

    b. A prefix byte is calculated where the first hexadecimal character indicates the number of decimal digits more than 9 (e.g. a 9 digit string is [9 – 9 = 0] and a 24 digit string is [24 - 9 = 15 = F_{HEX}]). The second hexadecimal character indicates the number of converted bytes more than 4.

    c. The byte calculated in Step 3b is preceded by byte $\mathbf{FB}_{HEX}$ to indicate that what follows is an integer string.

    d. The resultant encoding of a long numeric string is a single 16-bit word, FB{hex character for numeric string length}{ hex character for number of compacted bytes}, followed by a number of compacted bytes. The number of compacted bytes might or might not align with a 16-bit word boundary.

4. The resultant byte string of steps 2 and 3 is encoded from bit location 20_{HEX}. If the URN Code 40 encoding does not align with the 16-bit word boundary of MB 01, then an additional $\mathbf{00}_{HEX}$ byte is added at the end.

    NOTE This is in keeping with other ISO/IEC 15962 encoding rules and by others including those applied by GS1 EPC. The core encoding rule encodes to its rules, and prior to creating the 'payload' for the air interface Write command final alignment is required. Thus a compliant IPC test letter encoding system would ensure alignment to 16-bit word boundaries. Similarly a compliant IPC test letter decoder would ignore any trailing 00_{HEX} byte added only to align with a 16-bit word boundary.

EXAMPLE This example illustrates the four steps defined above.

Basic structure: B.{UserType}{IssuerCode}{SerialNumber}

Step 1: **B.A12312345678**

Step 2: **B.A** encodes as 0001000011100010_2 = $10E2_{HEX}$

Step 3a: The numeric string 12312345678 is converted to a byte string = $02DDDF7C4E_{HEX}$

Step 3b: A prefix byte is calculated where the first hexadecimal character **2** indicates the number of decimal digits more than 9 (i.e. 11 – 9 = 2). The second hexadecimal character is **1** to indicate the number of converted bytes more than 4 (i.e. 5 – 4 = 1)

Step 3c: The byte calculated in Step 3b is preceded by byte **FB** to indicate that what follows is an integer string.

Step 3d: Therefore **12312345678** encodes as $FB2102DDDF7C4E_{HEX}$

Step 4a: The resultant byte string is: $10E2FB2102DDDF7C4E_{HEX}$

Step 4b: This is encoded from bit location 20_{HEX}. To round this up to 16-bit words, a trailing byte 00_{HEX} has to be added. Presented in 16-bit (double byte) blocks the full hexadecimal encoding is: 10E2 FB21 02DD DF7C 4E00

There are various points to note:

- In the encoding rules of the basic URN Code 40 table, it is impossible to have a double byte (16-bit) string beginning with **FB**$_{HEX}$. As a result the URN Code 40 encoding scheme uses this byte value, when it is the first byte of a 16-bit word, in a similar manner as a numeric lock on a keyboard. The difference is that the second byte provides rules to automatically switch off or switch back. In the case of this IPC test letter standard, this provides the automatic 'off-switch' of trying to decode any trailing last byte 00_{HEX} in the last word of some of the test letter codes.

- Even if the {UserType} is a numeric digit, it shall be encoded as the part of the first three characters, using the basic encoding table. Having it as part of a 12-digit string:

  - is at variance with ISO/IEC 15962 encoding,

  - results in a longer encoded string.

- Because the input is always a fixed length, the resultant encoding is always fixed length of either 64 bits or 72 bits, which encodes as 80 bits because of the 16-bit word of the ISO/IEC 18000-63 tag.

- The value of the second 16-bit word is **FB20**$_{HEX}$ for the encoding that only requires 64 bits of memory (for the lowest IssuerCode values) and **FB21**$_{HEX}$ for the remaining IssuerCode values.

- Although the trailing byte 00_{HEX} is added in the case of **FB21**$_{HEX}$, the syntax rules ensure that it is ignored during decoding.

This encoding example is shown graphically in Annex C.

8.4.4 Rules for writing and locking MB 01

MB 01 does not support any form of selective locking, therefore the entire memory bank shall be locked. This ensures that the tag is in a read-only state with the UII being protected from accidental or deliberate changes. Additionally if a non-zero Kill password (see 8.2.2) is used, the tag cannot be rendered unreadable.

There are no commands to determine if MB 01 is locked (see 8.1).

8.5 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 can be used to 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.

9 Decoding MB 01 to ISO/IEC 15962 rules

9.1 AFI

The AFI should be used as part of the air interface *Select* command, and as such is not read. Only tags with the IPC assigned AFI **$A0_{HEX}$** will respond. Tags with other AFI values, with no AFI value (e.g. with data encoded to GS1 EPC rules) will be ignored.

If there is a requirement to read the AFI, e.g. for diagnostic purposes, then the protocol control word is read. An RFID tag that conforms to this IPC standard has its last 9 bits with this value: **110100000_2.** The leading '1' bit is required to indicate that the following 8 bits are the AFI.

9.2 Decoding and processing the Monomorphic-UII

The UII for this IPC standard is either 64 or 80 bits long, which means that four or five 16-bit words will be returned.

The first 16-bit word should be retained in a binary or hexadecimal format to check that it is a valid test letter code conforming to this IPC standard.

- In hexadecimal, the first hexadecimal character of the first word shall be 1_{HEX}.
  - If so, proceed with the decoding
  - If not, the encoding is wrong

- In binary, the first 4-bits of the first word shall be 0001_2.
  - If so, proceed with the decoding
  - If not, the encoding is wrong

Once the UII has been shown to be valid, decode the words in succession, using the inverse of the URN Code 40 rules defined in 8.4.3. This results in a 14-character string in the format:

B.{UserType}{IssuerCode}{SerialNumber}

The following additional validity checks may be applied during the decoding process:

- The encoded first word is in the range of $10E2_{HEX}$ to 1108_{HEX}

- The first two characters of the decoded first word are always '**B.**'. The third character is in the range A to Z and 0 to 9, but further restricted by the user codes assigned by IPC.

- The encoded second word is always either **FB20**$_{HEX}$ or **FB21**$_{HEX}$. This indicates that the following bytes are an integer encoding of an 11 digit decimal number representing the 3-digit IssuerCode and the 8-digit SerialNumber. Decoding these bytes also reconstructs any leading zeros in the IssuerCode.

9.3 Interoperability between ISO/IEC 18000-63 tags and active tags

9.3.1 General position

The components of IssuerCode and SerialNumber are of the same format, whichever tag (passive or active) is used. The encoding on the tag differs, but a properly decoded tag should result in the same structure even though different RFID technologies are used. The following sub-clauses discuss some specific issues.

9.3.2 Dual form factor tags

One option that may be considered to migrate from active tag to ISO/IEC 18000-63 tag data capture is to add a passive tag to the active tag. Systems can be set to enable the ISO/IEC 18000-63 tag to be the preferred, or default, tag. This makes sense if other IPC standards are in place in the data capture system. The system may be set up so that data from the active tag may still be captured. This also allows for a staged migration by different postal services from the active tag to the passive tag on a managed basis over a period of time.

To ensure that the same test letter is being read using either technology, the IssuerCode and SerialNumber shall be identical in structure and content. This means:

— Reading and decoding the content of the active tag.

— Using the IssuerCode and SerialNumber as part of the process for encoding the ISO/IEC 18000-63 tag as defined in 8.4.3. The UserType shall be one of the codes assigned by IPC.

— Combining the two tags into one form factor.

9.3.3 Standalone ISO/IEC 18000-63 tags

ISO/IEC 18000-63 tags that are not associated with an active tag, shall apply the encoding rules as defined in 8.4.3. The rules for decoding the AFI (defined in 9.1) and for the Monomorphic-UII (see 9.2) shall apply.

9.3.4 URN interoperability for ISO/IEC 18000-63 tags

To maintain interoperability with a URN structure, for example for processing via an EPCIS system, the 14-character string is retained and shall be prefixed by **1.0.15961.14.** (note that the final 'dot' is required), thus creating a unique object identifier.

9.3.5 URN interoperability for active tags

It is also possible for the decoded data from the active tags to be processed via an EPCIS system. In this case the 11-character {IssuerCode}{SerialNumber} shall be retained and shall be prefixed by **1.0.15961.14.B.{UserType}** (note that a final 'dot' is **not required** after the UserType), thus creating a unique object identifier.

10 Selecting Specific Populations of IPC Tags

10.1 Overview

The following clauses describe tag command functions at the interrogator layer and methods that can be used to:

— select or de-select a specific population of tags to respond in an inventory round

— truncate back scatter response of selected tags

— increase read throughput of tags of interest.

The methods make use of the air interface *Select* command to read and process particular strings of "raw" bits.

10.2 Structure and purpose of the *Select* command

The ISO/IEC 18000-63 air interface *Select* Command is issued at the beginning of an inventory round.

A ***Select*** command can be used prior to a ***Query*** command to specify which tag population to select. It may be all IPC tags to include test tags, receptacle assets and other postal products; or it could be a subset of a particular IPC standard as defined in the following sub-clauses. Only those tags that hear the ***Select*** commands and meet the criteria defined by the ***Select*** command(s) can be instructed to participate or not participate in an inventory round. This means that not only does the ***Select*** command focus on the tags required at a given data capture point, but also that all other tags that do not fulfil the requirements are effectively ignored.

The following sub-clauses define the ***Select*** command parameters for this IPC test letter standard. A technical explanation of this command is provided in Annex D.

10.3 Fast select

This method is used to select ISO/IEC 18000-63 RFID IPC test letter tags from any others. This method excludes tags from all other domains and tags encoded to conform to other IPC standards, some of which have still to be developed.

The *Select* command shall be constructed with 13 bits from bit positions 17_{HEX} to 23_{HEX} of MB 01 with the Mask value $\mathbf{1101000000001_2}$ as part of the command structure defined in Table 3. This represents the bit that identifies the encoding as being complaint to ISO rules, the AFI assigned to IPC and the first 4 bits of the 16-bit word that identifies that this is a test letter.

Table 3 - *Select* command parameters for test letters

	Command	Target	Action	MemBank	Pointer	Length	Mask	Truncate	CRC-16
# of bits	4	3	3	2	EBV	8	13	1	16
description	1010	100	001	01	00010111	00001101	1101000000001	0	

NOTE 'EBV' stands for Extended Bit Vector, which is a method used in ISO/IEC 18000-63 tags to be able to encode and bit string in a self declaring manner. In this case each 8-bit block comprises of the lead or extension bit followed by 7 bits that represent the numeric value. If the extension bit = 0 then it is the last block. If the extension bit = 1, then it is followed by another block.

EXAMPLES	decimal 127	01111111
	decimal 128	10000001 00000000
	decimal 16384	10000001 10000000 00000000

This results in the selection of all tags that conform to this IPC test letter standard. With the *Select* set with these parameters, the complete UII is returned.

10.4 Select to UserType

The UserType is the third character encoded in the first byte. This means that by using a longer select parameter it is possible to select to a specific UserType. To achieve this, the *Select* command shall be constructed with 25 bits from bit positions 17_{HEX} to $2F_{HEX}$ of MB 01 with the Mask value **110100000{16-bit first word}**$_2$ as part of the command structure defined in Table 4. For illustration, assume that the UserType is **A**, as in the encoding example used in 8.4.3. The first 16-bit word is therefore: 0001000011100010. The entire 25-bit Mask is 1101000000001000011100010, This represents the bit that identifies the encoding as being complaint to ISO rules, the AFI assigned to IPC and the first 4 bits of the 16-bit word that identifies that this is a test letter with UserType **A**. The CRC-16 is calculated by the interrogator.

NOTE This illustration and the Mask value in Table 4 only applies to UserType A. The other UserType codes will have different Mask values, for the final 16 bits.

Table 4 - *Select* command parameters for a particular UserType

	Command	Target	Action	MemBank	Pointer	Length	Mask	Truncate	CRC-16
# of bits	4	3	3	2	EBV	8	25	1	16
description	1010	100	001	01	00010111	00011001	110100000 0001000011100010	0	

This results in the selection of all tags that conform to the particular UserType code. With the *Select* set with these parameters, the complete UII is returned.

If more than one UserType is required to be selected, a truncated version of the 16-bit first word may be used as follows:

— If all the UserTypes are numerical and in the range 1 to 9, then the first byte 11_{HEX} is common. This means that a 17-bit Mask value $\mathbf{11010000000010001_2}$ is required as part of the command structure defined in Table 4.

— If all the UserTypes are alphabetical, then the first byte 10_{HEX} is common. This means that a 17-bit Mask value $\mathbf{11010000000010000_2}$ is required as part of the command structure defined in Table 4.

— The Mask value for the alphabetical UserTypes will also have a response from tags with the UserType =**0**.

Operational conditions will determine whether this more refined selection is implemented.

10.5 Reading the UII as a raw bit string

There can be situations where there is an opportunity to avoid decoding the UII and simply read the UII as a binary string or hexadecimal string (depending in the output of the interrogator). This method can be used where the RFID tag on item is certain to conform to this IPC test letter standard, e.g. by invoking the methods in 10.3 or 10.4. These methods deliver the raw UII.

As an example, the raw UII data from test letters can be passed to a second stage process where decoding in invoked. This might enable a faster switch to a *Select* command that addresses production items.

Annex A (informative) -- Information about tag compliance

A.1 Memory requirements

These are the requirement guidelines to fully support this IPC test letter standard:

MB 00 This memory bank support passwords in ISO/IEC 18000-63 compliant products. The Access password is not required for this standard. The Kill password should be used for this test letter standard (see 5.2.2)

MB 01 Memory capacity is measured from bit position 20_{HEX}. This memory bank encodes the UII, i.e. the S10 code. Although this only requires 80 bits of memory, most tags support at least 96 bits of UII memory. Generally unused memory cannot be used, but the length indicator in the protocol word is used to restrict air interface transmission to the data that is encoded.

MB 10 This memory bank contains details of the chip manufacturer, model number and often a unique serial number. IPC has a list of compliant tags identified by manufacturer / model number. The memory bank in not generally required for this IPC S10 standard, except to confirm compliance, and to possibly use the serial number for diagnostic purposes.

MB 11 This memory bank is not required for this test letter standard.

A.2 Performance requirements

IPC has established an RFID tag conformance standards, to which all tags shall conform. It also maintains a list of tags that meet to performance requirements. More details can be obtained from rfid@ipc.be.

Annex B (normative) -- Monomorphic-UII and URN Code 40 encoding

B.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 IPC test letter standard, 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 item-related data is required for the application, then this shall use an entirely different AFI and Data-Format. For this part of IPC standard, the AFI is $A0_{HEX}$.

— 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 IPC test letter standard, the Object Identifier for the UII is 1.0.15961.14.

— Each AFI declares which of the encoding schemes, as defined in ISO/IEC 15962 is used. For all IPC standards, 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.

B.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 is required for resolving on the Internet.

The UII shall be encoded using the URN Code 40 encoding defined in the following sub-clauses.

B.2.1 Basic Character Set

For this IPC test letter standard, this shall only apply to the first three characters: **B.{UserType}**

The encoding process takes the string of three data characters (from Table 5) and compacts these into two bytes. The PAD character and punctuation characters shall not be used for this IPC test letter standard.

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 1 to 64000, which is converted to hexadecimal in the range 0001_{HEX} to $FA00_{HEX}$.

Table 5 - 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

B.2.2 Encoding a long numeric string

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 leading byte can never have a value FB_{HEX} in the table-driven solution. In fact, this byte value is used as syntax to encode different encoding, extending the capability of the scheme. The lead byte $\mathbf{FB}_{HEX}$ is used to signal long string integer encoding. If the number of contiguous decimal digits is equal to or greater than 9, 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.

NOTE Other lead byte values are specified for other purposes in ISO/IEC 15962. These are beyond the scope of this IPC test letter standard.

The encoding consists of three components, as shown in Table 6.

Table 6 — Structure for encoding long numeric strings

Lead Byte	Second Byte	Subsequent Bytes
FB	1st hex char = number of decimal digits 9 –24 2nd hex char = number of encoded bytes 4 - 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.

Annex C (informative) – Bit mapping of encoding example

This Annex illustrates the bit and byte mapping described in 8.4 and shows the encoding of the AFI (see 8.4.2) and the worked example of the UII (see 8.4.3), together with the tag selection process defined in 10.

IPC Test Letter using ISO/IEC 15962 URN Code 40 Rules

	CRC	Protocol Control Word																Unique Item Identifier (UII)								
Bit Address	00 - 0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20 - 23	24 - 27	28 - 2B	2C - 2F	30 - 3F	40 - 4F	50 - 5F	60 - 6F	70 - 7F
UII Bit Length																		4	8	12	16	32	48	64	80	96
	16-bit word	16-bit word																16-bit word				16-bit word	16-bit word	16-bit word	16-bit word	16-bit word
	CRC	Protocol Control Word																1st Word of UII				2nd word of UII	3rd word of UII	4th word of UII	5th word of UII	Not Required
		Length of UII							1= ISO	AFI								Test Letter identifier, 'dot', UserType code				Switch to numeric compaction. FB is switch key. (11-9) = 2decimal digits. (5-4) = 1 encoded bytes	3-digit Issuer Code, including any leading zeros. This is followed by 8-digit Serial Numberincluding leading zeros. Depending on the input values, this will encode in 4 bytes or 5 bytes. The shorther encoding ends at the 4th word of UII, The longer encoding requires a trailing byte 00, and the encoding ends at the 5th word of UII.				
Input Example																		B.A					12312345678			
Encoding Bits		0	0	1	0	1	0	0	1	1	0	1	0	0	0	0	0	0001	000011100010			1111101100100001	0000001011011101	1101111101111100	0100111000000000	
Encoding HEX										A0								10E2				FB21	02DD	DF7C	4E00	

Fast select	13 bits: 17 to 23 HEX
Select to User Type	25 bits: 17 to 2F HEX
Read Test Letter	80 bits 20 to 6F HEX for longer codes and 20 to 5F HEX for shorther codes - declared by the Length of UII bits in the PC word

URN Code 40: This is an encoding method (fully defined in ISO/IEC 19652) that takes a string of three characters and encodes them in 16 bits. The base character set is Uppercase Alphabetic, Numeric, and some punctuation to a total of 40 code points. Using the base set results in the most efficient coding of mixed alpha-numeric data at 5.33 bits per character. The encoding for test letters invokes an automatic switch to an efficient numeric encoding scheme.

IPC is responsible for assigning the UserType and Issuer codes.
The Issuer code and Serial Number are encoded as a contiguous 11-digit string. Any leading zeros are therefore retained during the decoding process.
The length of UII is 00101 in this example; it is 00100 for the shorter encoding.

Annex D (informative) ISO/IEC 18000-63 *Select* command

Successive ***Select*** commands can be used prior to a ***Query*** command to widen tag population selection to include test tags, assets and other postal products. Only those tags that hear the ***Select*** commands and meet the criteria defined by the ***Select*** command(s) can be instructed to participate or not participate in an inventory round.

When a tag meets selection criteria its **Select flag** gets asserted or de-asserted depending on the **Action** variable state issued in the ***Select*** command. When a ***Query*** command is sent after the ***Select*** command(s) at the beginning of a tag inventory round, it can request only those tags with **Select** and specific **Inventory** flags set to participate. The ***Query*** command also sets a session number S0 to S3 to support tag communication with multiple readers.

The ***Select*** command (Table 7 - ISO/IEC 18000-63 Select command parametersTable 7) includes the following parameters:

— **Target**: 3 bits: indicates which flags, SL (select flag) or inventoried and sessions flags are changed as a result of meeting ***Select*** command filter requirements.

— **Action**: 3 bits: This parameter is set to determine the tag response as shown in Table 8. Put more simply:

a) if **Target** specifies SL flag is changed – whether the SL flag is set (asserts) or resets (de-asserts) if a tag meets selection criteria

b) if **Target** specifies inventory flag change the options are change to A or to B.

— **MemBank:** Specifies whether the Mask applies to UII, MB 11 or TID memory. The ***Select*** command filter mask components can operate on the UII, MB11 or TID memory. Successive ***Select*** commands prior to a **Query** command can extend filtering for content in multiple memory banks.

— **Filter Parameters:** The filter parameters include **Pointer, Length** and **Mask. Pointer** and **Length** describe the memory range**.** The **Pointer** references a bit address using **EBV** formatting. **Length** defines the mask length in bits. **Mask** contains a bit string that a Tag compares against the memory location that begins at the pointer and ends **Length** bits later.

— **Truncate:** The **Truncate** function operator only works on UII backscatter response. If the ***Select*** Command has the **Truncate** parameter selected or asserted and if a subsequent **Query** command specifies **Sel** Paramaters (**Sel**=10 or **Sel**=11) then a matching tag will truncate its reply to an acknowledgement, **ACK**, to the portion of the UII immediately following the **Mask** followed by the **StoredCRC**. If a user is to apply or use successive ***Select*** commands and desires to truncate the response the user must assert or set **Truncate** in the last **Select** command prior to sending a **Query** command. The **Target** parameter must be set (100) such that tags will set **Sel** flag as a result of matching the **Mask** and the **Mask** ends with in the UII.

*NOTE: Not all reader vendors support the **Truncate** function of the **Select** command. As this is not required for this IPC receptacle asset standard it should be set to **0** to disable the function.*

Table 7 - ISO/IEC 18000-63 Select command parameters

	Command	Target	Action	MemBank	Pointer	Length	Mask	Truncate	CRC-16
# of bits	4	3	3	2	EBV	8	Variable	1	16
description	1010	000: **Inventoried** (S0) 001: **Inventoried** (S1) 010: **Inventoried** (S2) 011: **Inventoried** (S3) 100: **SL** 101: RFU 110: RFU 111: RFU	See Table 8	00: Reserved 01: UII 10:TID 11:User	Starting Mask address	Mask length (bits)	Mask value	0: Disable truncation 1: Enable truncation	

Table 8 - ISO/IEC 18000-63 tag response to Action parameter

Action	Matching	Non-matching
000	assert **SL** or **inventoried** $\rightarrow A$	de-assert **SL** or **inventoried** $\rightarrow B$
001	assert **SL** or **inventoried** $\rightarrow A$	do nothing
010	do nothing	de-assert **SL** or **inventoried** $\rightarrow B$
011	negate **SL** or $(A \rightarrow B, B \rightarrow A)$	do nothing
100	de-assert **SL** or **inventoried** $\rightarrow B$	assert **SL** or **inventoried** $\rightarrow A$
101	de-assert **SL** or **inventoried** $\rightarrow B$	do nothing
110	do nothing	assert **SL** or **inventoried** $\rightarrow A$
111	do nothing	negate **SL** or $(A \rightarrow B, B \rightarrow A)$

Bibliography

UPU Standards glossary (accessible at http://www.upu.int)

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