

ISO/IEC 15962:2004-10 (E)

Information technology - Radio frequency identification (RFID) for item management - Data protocol: data encoding rules and logical memory functions

Contents		Page
Foreword		vi
Introduction		vii
1	Scope	1
2	Normative references	1
3	Terms, definitions and abbreviated terms	2
3.1	Terms and definitions	2
3.2	Abbreviated terms	4
4	Protocol model	4
4.1	Overview	4
4.2	Layered protocol	5
4.3	Functional processes	6
5	Data structure	8
5.1.1	The 8-bit byte	8
5.1.2	N-bit encoding	8
6	Data Protocol Processor and the application interface	8
6.2	Universal Types	9
6.3	Length encoding and decoding	10
6.4	Decoding of Type values	10
6.4.1	Decoding of a BOOLEAN value	10
6.4.2	Decoding an INTEGER value	10
6.4.3	Decoding an OBJECT IDENTIFIER value	11
6.4.4	Decoding the OCTET STRING value	11
6.4.5	Decoding the SEQUENCE or SEQUENCE OF value	11
7	Data Protocol Processor and the air interface	12
7.1	Air interface services	12
7.2	Defining the system information	13
7.2.1	Tag Identifier	13
7.2.2	Physical block size	14
7.2.3	Number of blocks	14
7.2.4	Application Family Identifier	14
7.2.5	Storage Format	14
7.3	Configuring the Logical Memory	15
7.3.1	Non- directory structure of the Logical Memory	15
7.3.2	Directory structure of the Logical Memory	16
7.3.3	Self mapping RF tags	16
8	Data flows and processes	17
8.1	Application data	17
8.1.2	Processing example	18
8.2	Data object processing	19
8.2.1	Compaction process	19
8.2.2	Compaction Schemes	20
8.2.3	Compaction Type codes	21

8.2.4	Encoding the length of the compacted object	21
8.2.5	Example of encoding for the Logical Memory after compaction	21
8.3	Data formatting	22
8.3.1	Data Formatter Functions	22
8.3.2	Formatting the objectId	23
8.3.3	The Precursor for dataFormat not equal 2	24
8.3.4	The Precursor for the root-OID for dataFormat = 2	24
8.3.5	Encoding the RELATIVE-OID	24
8.3.6	Encoding the OBJECT IDENTIFIER	26
8.3.7	Encoding the root-OID for dataFormat = rootOidEncoded (2)	26
8.3.8	Encoding the object and its length	27
8.3.9	The offset byte	27
8.3.10	The Precursor expansion byte	27
8.3.11	The directory structure	27
8.3.12	Addressing from the directory	28
8.3.13	Structures of Logical Memory	28
8.4	Decoding the Logical Memory	28
8.4.1	Overall decode strategy	28
8.4.2	Decoding the storageFormat	28
8.4.3	Decoding the Precursor	29
8.4.4	Decoding the leading byte(s) of the encoded objectId	30
9	The Command / Response unit	31
9.1	Commands	31
9.1.1	Configure Application Family Identifier command	31
9.1.2	Configure Storage Format command	31
9.1.3	Inventory Tags command	32
9.1.4	Add Single Object command	32
9.1.5	Delete Object command	32
9.1.6	Modify Object command	32
9.1.7	Read Single Object command	33
9.1.8	Read ObjectIds command	33
9.1.9	Read All Objects command	34
9.1.10	Read Logical Memory Map command	34
9.1.11	Inventory And Read Objects command	34
9.1.12	Erase Memory command	35
9.1.13	Get Application-based System Information command	35
9.1.14	Add Multiple Objects command	35
9.1.15	Read Multiple Objects command	35
9.1.16	Read First Object command	36
9.2	Processing arguments	36
9.2.1	afiLock	36
9.2.2	avoidDuplicate	36
9.2.3	checkDuplicate	37
9.2.4	identifyMethod and numberOfTags	37
9.2.5	lockStatus	38
9.2.6	maxAppLength	38
9.2.7	objectLock	38
9.2.8	storageFormatLock	38
9.3	Completion codes	39
9.4	Execution codes	40
10	Communications between the Data Protocol and the RF tag	41
11	Compliance, or classes of compliance, to this International Standard	41
11.1	Compliance of the Data Protocol Processor	41
11.2	Compliance of the Tag Driver	41
Annex A (normative)	Pro Forma Description for the Tag Driver	42
A.1	Defining the tagId	42
A.2	System information : applicationFamilyId	42

A.3	System information : storageFormat	42
A.4	Memory-related parameters	42
A.5	Support for commands	43
kHz	44
at 13,56 MHz	45
at 13,56 MHz	46
2,45	GHz - Mode 1	47
2,45	GHz - Mode 2	48
to 960 MHz	49
Annex C (normative) Data Compaction Schemes		51
C.1	Integer compaction	51
C.2	Numeric compaction	51
C.3	5-bit compaction	52
C.4	6-bit compaction	52
C.5	7-bit compaction	53
C.6	Octet encodation	54
Annex E (informative) Encoding Example		59
E.1	Starting position	59
E.2	The initial state of the entry for the Logical Memory	59
E.3	The Logical Memory after data compaction	59
E.4	The Logical Memory after formatting with a noDirectory accessMethod	60
Annex F (informative) Logical Memory Structures		62
F.1	Notation	62
F.2	Non-directory structured Logical Memory with root-OID implicitly encoded	62
F.3	Non-directory structured Logical Memory with root-OID explicitly encoded	63
F.4	Directory structured Logical Memory with root-OID implicitly encoded	63
F.5	Directory structured Logical Memory with root-OID explicitly encoded	63