INTERNATIONAL STANDARD

ISO/IEC 24791-5

First edition
2012-12-15

Information technology — Radio frequency identification (RFID) for item management — Software system infrastructure

Part 5: Device interface

Technologies de l'information — Identification de radio fréquence (RFID) pour la gestion d'élément — Infrastructure de systèmes logiciels

Partie 5: Interface de dispositif

Reference number
ISO/IEC 24791-5:2012(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword

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

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

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

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

ISO/IEC 24791-5 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology,* Subcommittee SC 31, *Automatic identification and data capture techniques*.

ISO/IEC 24791 consists of the following parts, under the general title *Information technology — Radio frequency identification (RFID) for item management — Software system infrastructure*:

— *Part 1: Architecture*

— *Part 2: Data management*

— *Part 3: Device management*

— *Part 5: Device interface*

Introduction

RFID air interface technology is based on non-contact electro-magnetic communication among interrogators and tags. RFID software systems are composed of RFID interrogators, intermediate software systems, and applications that provide control and coordination of air interface operation, tag information exchange, and health and performance management of system components. RFID technology is expected to increase effectiveness in many aspects of business by further advancing the capabilities of Automatic Identification and Data Capture (AIDC). To achieve this goal through the successful adoption of RFID technology into real business environments, RFID devices, software systems, and business applications shall provide secure and interoperable services, interfaces, and technologies. This is the goal of the standards defined for RFID Software System Infrastructure (SSI), ISO/IEC 24791. The composition and operations of SSI exist in systems that implement other RFID standards including the air interfaces described in the ISO/IEC 18000 series and the data and interface functions defined in ISO/IEC 15962, ISO/IEC 15963, ISO/IEC 24753 and others.

The goal of this part of ISO/IEC 24791 is to define a device interface that provides RFID controlling software with low-level access to RFID air interface hardware. This low-level access gives programmers a degree of control over the sequencing of air protocol commands and direct access to air protocol command parameter. Using this low-level interface, programmers can optimize RFID data access and control operations.

The interface defined by this part of ISO/IEC 24791 supports the following features:

- efficient, binary transfer syntax over TCP/IP
- access to RFID air protocol commands and command parameters
- support for optimized RFID tag access operations whereby multiple operations can be performed on a tag with minimal tag state changes
- direct read/write access to all data on an RFID tag
- read one or more individual tag data items (encoded as defined by ISO/IEC 15962) as specified by their OID using URN notation
- [optional] decode data items (encoded as defined by ISO/IEC 15962) into their Unicode representation (UTF-8 encoded). Encoding data items is not supported
- support for RFID air protocol type defined by Type C of ISO/IEC 18000-6.

The interface defined in this part of ISO/IEC 24791 provides access to RFID air protocol commands and their respective command parameters. Therefore using this interface, tag memory banks can be locked, tags can be killed, and raw-binary RFID tag data can be accessed directly on a tag for both reading and writing. In addition, individual data items (encoded as defined by ISO/IEC 15962) can be read by specifying each data item's OID. Optionally, the interrogator can decode data items read into their character string representation. If the interrogator cannot decode a data item, then it will return the entire encoded package within which the data item resides. In this case, it is the responsibility of higher-level software to further decode the data item. The interface does not support RFID tag data encoding. It is the responsibility of higher-level software (i.e., software outside the interface defined by this part of ISO/IEC 24791 to perform data encoding (i.e., binary tag data representation as defined by ISO/IEC 15962) that is stored on RFID tags.

This part of ISO/IEC 24791 is composed of the EPCglobal™ standard, *Low Level Reader Protocol* [LLRP], in its entirety with extensions that support reading RFID tag data items that are encoded according to ISO/IEC 15962. As does the LLRP standard, this part of ISO/IEC 24791 defines both the abstract functional capabilities of the interrogator interface and the binary transfer syntax between the interrogator and a controlling system device. The transfer syntax is defined to be communicated over TCP/IP.

Information technology — Radio frequency identification (RFID) for item management — Software system infrastructure

Part 5: Device interface

1 Scope

This part of ISO/IEC 24791 defines an interface within the Software System Infrastructure (SSI) that provides RFID system control components with low-level access to RFID interrogators for the purpose of optimizing RFID data access and control operations. This interface is designed to be modular with the ability to support multiple RFID air protocols. However, in this this part of ISO/IEC 24791, the only RFID air protocol supported is Type C of ISO/IEC 18000-6.

2 Conformance

Conformance for this part of ISO/IEC 24791 shall satisfy the requirements of the EPCglobal LLRP specification and the requirements of the LLRP extensions defined in clause 9 of this part of ISO/IEC 24791. Clause 9 is partitioned into sub-clauses and each sub-clause includes a specific compliance requirement paragraph. Therefore, conformance to clause 9 of this part of ISO/IEC 24791 is defined by the shall statements of the compliance requirement paragraphs found within the sub-clauses of clause 9. The EPCglobal LLRP specification also has specific compliance requirement paragraphs. Therefore, conformance to the EPCglobal LLRP specification is defined by the shall statements of the compliance requirement paragraphs found in the LLRP specification.

3 Normative references

The following referenced documents are indispensable for the application 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 8859-1, *Information technology — 8-bit single-byte coded graphic character sets — Part 1: Latin alphabet No. 1*

ISO/IEC 19762-1, *Information technology — Automatic identification and data capture (AIDC) techniques — Harmonized vocabulary — Part 1: General terms relating to AIDC)*

ISO/IEC 19762-3, *Information technology — Automatic identification and data capture (AIDC) techniques — Harmonized vocabulary — Part 3: Radio frequency identification (RFID)*

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

EPCglobal™ LLRP, *Low Level Reader Protocol*

4 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 19762-1, ISO/IEC 19762-3, and the following apply.

4.1
device interface
communications interface that provides tag data access and control operations between an RFID Interrogator and an upstream Client

4.2
endpoint
one of two components that either implements and exposes an interface to other components or uses the interface of another component

4.3
interrogator endpoint
component that implements and exposes the interrogator interface defined by this part of ISO/IEC 24791 to other system components

4.4
client endpoint
component that uses the interrogator interface defined by this part of ISO/IEC 24791 to access interrogators

4.5
interface
shared boundary between two functional units, defined by various characteristics pertaining to the functions, physical or software interconnections, signal exchanges, and other characteristics, as appropriate

4.6
interrogator controller
system component, possibly a distinct physical device, which is capable of exercising the data, control, and management of interrogators as described in this part of ISO/IEC 24791

4.7
data set
binary representation of a data element's identifier and its associated data value that is encoded according to the rules defined by ISO/IEC 15962 standard access method

4.8
data item
pairing of a data element's identifier and its associated data value that is encoded according to the rules defined by ISO/IEC 15962, regardless of access method

4.9
encoded package
self-describing binary data from an RFID tag memory bank that, when given the memory bank's DFSID (which defines the ISO/IEC 15962 access method and data format), decoding software can fully decode the individual data item(s) to extract the encapsulated data

4.10
protocol data unit
unit of information communicated between network peers

5 Abbreviated terms

For the purposes of this part of ISO/IEC 24791, the symbols and abbreviated terms given in ISO/IEC 19762-1, ISO/IEC 19762-3, and the following apply.

AIDC
Automatic Identification and Data Capture

DSFID
Data Storage Format Identifier

LLRP
Low Level Reader Protocol

RF
Radio Frequency

SSI
Software System Infrastructure

UML
Unified Modeling Language

6 Software System Infrastructure Architecture Overview

ISO/IEC 24791-1 defines the architecture for the Software System Infrastructure. The basic relationship among the interfaces and implementations of the Software System Infrastructure is depicted in Figure 1 — Architecture Overview including Relationships to other RFID Standards.

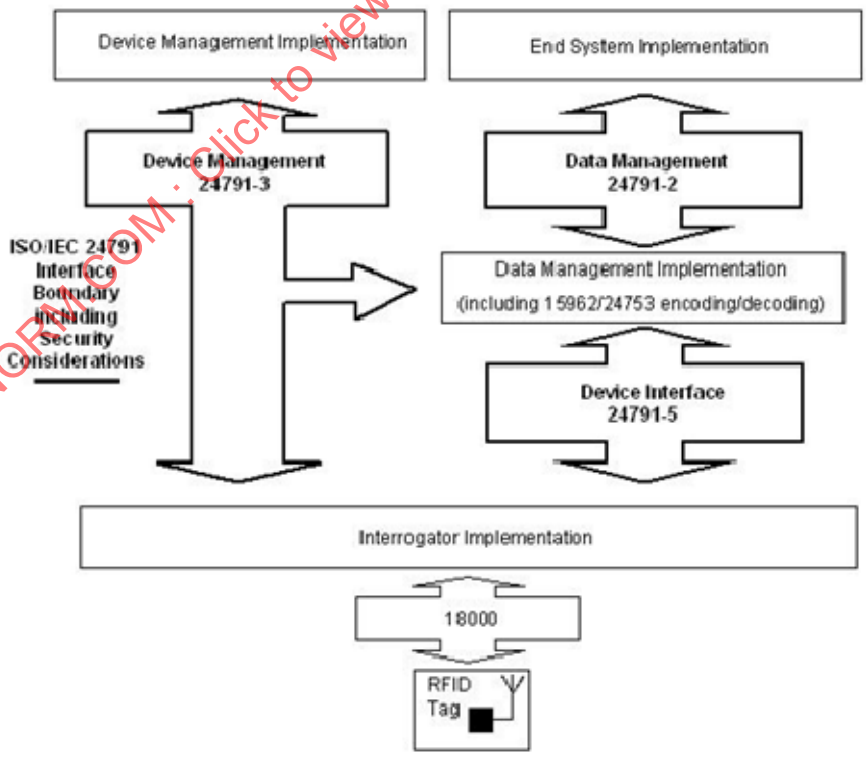


Figure 1 — Architecture Overview including Relationships to other RFID Standards

The Data Interface, Device Interface, and Device Management each provide one or more interfaces that allow a client to communicate with a service-providing implementation, either within the same computing device or across a network. These client and service implementations are consistently referred to as Client Endpoints and Services Endpoints, respectively, and in general, the Client Endpoint accesses the capabilities provided by the Services Endpoint. It is the responsibility of the specific standard to define the formats, procedures, operations, and conformance requirements of each interface.

7 UML Modeling

Although Figure 1 provides a general overview of the relationship between the interfaces and implementations in the SSI, Unified Modeling Language (UML) is used in the remainder of the document to graphically represent the organization and operation of the Device Interface and implementations so that a precise and common understanding of the relationships among the components can be defined.

UML is a very rich language, but for simplicity only the Physical Diagram subset of the language is used to represent the architecture of the Software System Infrastructure. Physical diagrams, composed of Component Diagrams and Deployment Diagrams, represent the relationships among the functions and the interfaces provided by the SSI architectural elements as well as how these functions might exist in standards compliant solutions, respectively. Refer to ISO/IEC 24791-1 for a more complete description how UML is used in this part of ISO/IEC 24791.

8 Device Interface

8.1 General

This clause describes the device interface defined by this part of ISO/IEC 24791 in terms of the overall software system infrastructure described in ISO/IEC 24791-1. Annex A of this part of ISO/IEC 24791 provides further references to several open sources projects that offer software libraries and utilities supporting this device interface.

8.2 Architecture

As defined in ISO/IEC 24791-1, the Device Interface defines *data* and *control* operation between the Device Interface Services Endpoint on an RFID interrogator and the Client Endpoint typically on a different physical device. A Device Interface component exposes an interface as represented in Figure 2 — Device Interface Representation whereby a Device Interface Services Endpoint provides services to a Device Interface Client Endpoint over an interface binding as specified in clause 10 of this part of ISO/IEC 24791, ISO/IEC 24791-5. The Client and Services Endpoints may exist in a single device or may be accessed across a network.

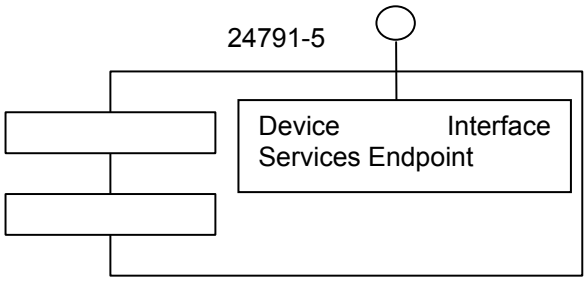


Figure 2 — Device Interface Representation

The Device Interface is responsible for supporting and exercising the capabilities provided by the RFID air protocols to achieve the tag and sensor access goals of an RFID software system. In order to achieve this goal, the standards provided for the data, control, and management functions at this level in the Software System Infrastructure are air-protocol aware and capable of fully exercising specific features of the supported air protocol standards. Equivalently, it is *not* a goal to provide a single, abstract interrogator interface that provides a common interface regardless of the specific air protocol in use. Doing so would reduce the ability of the software systems to access the specific features provided by existing and future air protocols.

The Device Interface supports requests for, and communication of, tag information between the Services Endpoint on RFID interrogators and the Client Endpoint on upstream devices. The data functions include capabilities such as reading, writing, filtering, and the reporting of tag and sensor data. Synchronous, event-based, and externally triggered operation may be supported for access requests.

The control functions in the architecture provide an interface for the Device Interface Client to exercise specific control over the access operations on an interrogator. This control is specific to each air protocol and is intended to provide the necessary capabilities to allow for software system, interrogator, and RF environment optimization. The control function of the Device Interface also supports the delivery of system, interrogator, tag, and RF environment data from the Services Endpoint to the Client Endpoint. This data can be used by implementations to provide the control required to achieve the desired system goals.

9 LLRP Extensions Abstract Definitions

9.1 General

ISO/IEC 24791-5 directly references and therefore utilizes the complete structure, definitions, formats, and procedures of the EPCglobal Low Level Reader Protocol (LLRP) specification **[LLRP]** with the addition of extensions that provide support for reading and decoding RFID Type C of ISO/IEC 18000-6 tag data formatted and operated upon according to ISO/IEC 15962. If an implementation of this part of ISO/IEC 24791 is unable to decode a tag data object, then it shall transfer to the client endpoint the raw, binary contents of the entire data set (i.e., the data object and its associated header data such as precursor and length fields).

Tag data may be encoded according to ISO/IEC 15962 to represent different character sets (e.g., the default ISO/IEC 8859-1, or UTF-8, or application defined). Tag data encoded as UTF-8 (compaction code 111) is fully compatible with this part of ISO/IEC 24791, and is transferred “as-is”. It is the responsibility of the client application to convert this UTF-8 data to other character sets if needed. Tag data found to be encoded as application defined (compaction code 000) cannot be decoded by an implementation of this part of ISO/IEC 24791 and as such, it shall be transferred as raw, binary data. Tag data that is encoded with no directory, directory, or tag data profile access methods and with the following compaction codes (001, 010, 011, 100, and 101) when decoded have the same single byte value on the tag and in UTF-8 format. If the compaction scheme is declared as Octet encoded (compaction code 110) then the encoding on the tag shall need to be converted to UTF-8 if any byte value is in the range between 80_h to FF_h. An example of the conversion algorithm is given in Appendix B.

The remainder of this part of ISO/IEC 24791 defines the ISO/IEC 24791-5 extensions to LLRP. This clause provides an abstract definition of interface parameters used to request and return decoded tag data. Clause 10 defines the transfer syntax details required to implement these interface parameters.

When referencing the EPCglobal LLRP specification, it will be helpful to understand that there are several EPCglobal terms that are synonymous with ISO/IEC terms. The following list enumerates EPCglobal terms with definitions using ISO/IEC terminology:

- **EPC (Electronic Product Code)**
  identifier corresponding to the ISO/IEC unique item identifier (UII) as defined by air protocol interface Type C of ISO/IEC 18000-6 when memory bank 01_2 bit 17_h is zero

- **EPC Memory Bank**
  equivalent to the UII memory bank as defined by air protocol interface Type C of ISO/IEC 18000-6

- **Class 1, Gen 2 (C1G2)**
equivalent to the air protocol interface, Type C, as defined in ISO/IEC 18000-6

- **Reader**
equivalent to an interrogator as defined in ISO/IEC 19762-3

The ISO/IEC 15962 RFID tag data encoding rules are defined in terms of data items that represent a pairing of the data item identifier (OID) and its associated data value. Data items, as defined by ISO/IEC 15962 rules, include data encoded as either packed objects or concatenated data sets (i.e., two different access methods defined by ISO/IEC 15962).

This part of ISO/IEC 24791 extends LLRP for the purpose of providing a Client endpoint with the following specific capabilities:

a) request an Interrogator to search a tag for one or more specific OIDs and to report back to the Client the occurrences of each OID as it is found on the tag;

b) request an Interrogator to report all OIDs found on a tag;

c) request an Interrogator to check for duplicate occurrences of one or more OIDs on a tag and return the number of instances found;

d) request for either raw or decoded data to be returned for each OID returned in the response to a particular request.

The following sub-sections provide abstract definitions of LLRP extensions designed to satisfy these capabilities. Clause 10 provides the binary transfer syntax definition for these extensions.

The LLRP extensions defined in this part of ISO/IEC 24791 correspond to the section titled *Access Operation* of the LLRP specification which defines air-protocol-specific access-operation parameters (e.g., Read, Write, Kill, Lock, etc.). These extensions define a new LLRP access-operation parameter and four new supporting parameters. The following sub-sections define these new LLRP parameters as they apply to Type C of ISO/IEC 18000-6.

9.2 LLRP Data Types

LLRP includes definitions of abstract data field types. For this part of ISO/IEC 24791, these data field types are extended with one additional field type:

ObjectIdentifier – This field type is used to represent Object Identifiers (OID) as defined by ISO/IEC 15961-1. An Object Identifier is as a character string that is structured following the URN syntax rules defined by RFC 3061.

9.3 ISO15962Read Parameter

9.3.1 General

This parameter, ISO15962Read, is an extension to LLRP. It is an access-operation parameter that supports reading one or more ISO/IEC 15962 data items from a specified memory bank on an RFID tag. The results of this access operation are reported with an associated ISO15962ReadResult parameter (see 9.4).

This access operation requires that the Interrogator is capable of parsing data encoded on the tag only enough to locate one or more encoded data items given a list of one or more object identifiers (OIDs). Although the Client may request decoded data from the Interrogator, the Interrogator is not required to support the ability to decode tag data and it may return the raw binary-encoded data as it is read directly from the tag. In this case, it is the responsibility of the Client software to decode this binary data. If the Interrogator does return raw binary data, then it will return the tag memory's DSFID and one or more entire *encoded packages* such that higher-level software will be capable of fully decoding the data items requested by the Client.

If the interrogator has a minimal AFI table that lists the received AFI (but does not list compaction modes for Monomorphic UIIs), then for a monomorphic UII the interrogator will return an empty DSFID field, and the undecoded data is returned.

If the interrogator has a full AFI table, then the interrogator will return an empty DSFID field, and monomorphic UII can be decoded and returned.

If the interrogator does not have an AFI table or the AFI is not listed in the interrogator's table, return an error code, and the application can issue a simple Read command instead.

In addition to requesting one or more specific OIDs, a Client can request to read all data items found in a specified memory bank on the tag.

NOTE In general, duplicate data items (i.e., two or more data items with identical OIDs) are not expected to exist within a tag's memory bank. Nevertheless, this part of ISO/IEC 24791 provides a mechanism for detecting such an erroneous tag state. If the *Find Duplicates* flag is set true, then the Interrogator will return a count of each OID found. Obviously, a count greater than one for any OID indicates that this OID is duplicated.

Compliance requirement: Compliant Interrogators and Clients shall implement this parameter. However, compliant Interrogators MAY only implement return data control codes 0-1 (none or raw data) and not support code 2 (decoded data).

ISO15962Read Parameter

OpSpecID : Unsigned Short Integer

This data field is a number generated dynamically by the Client for each OpSpec parameter sent to the Interrogator. This data field value is then used by the Interrogator as a handle that is returned with the corresponding ISO15962ReadResult parameter. The handle can thereby be used by the Client to correlate specific read requests with their corresponding read results.

Memory Bank : Integer

Possible Values:

Value*	Definition
1	UII memory bank
3	User memory bank

OIDs: List <OID parameters> [Optional]

The Client provides this list of one or more OID parameters for the Interrogator to locate in a tag's memory bank.

If this list is omitted, then the Interrogator will report all OIDs it finds in the tag's memory bank.

Find Duplicates: Boolean

If this data field is set true, then the Interrogator will report every instance of each OID requested. If set true and the OIDs list if empty, then the Interrogator will report all data items it finds in the tag's memory bank including duplicate OIDs. If set false, then the Interrogator will report only the first instance encountered (if any) for each OID requested.

Return Data Control Code: Integer

Possible Values:

Value	Definition
0	None. (don't return data)
1	Raw Data. (return raw binary data read from tag)
2	Decoded Data. [optional]

*NOTE Memory bank values enumerated here are arbitrary code values assigned to represent the different memory banks in the interface specified by this part of ISO/IEC 24791. For convenience, the decimal values of 0,1,2,3 were chosen as code values to correspond to the air protocol memory bank binary values of 00_2, 01_2, 10_2, 11_2 that are specified by ISO/IEC 18000-6. In the case of this parameter (ISO15962Read), only memory banks UII and User Memory apply.

9.3.2 OID Parameter

This parameter, OID, simply provides a parameter encapsulation of an OID for the purpose of supporting a list of zero or more OIDs as found embedded in the ISO15962Read parameter.

The monomorphic UII can be searchable by OID if the interrogator can match the monomorphic UII with the specific OID by means of an AFI table.

Compliance requirement: Compliant Interrogators and Clients shall implement this parameter.

ObjectIdentifier Parameter

OID: ObjectIdentifier

9.4 ISO15962ReadResult Parameter

9.4.1 General

This parameter, ISO15962ReadResult, is used to report the results of the ISO15962Read tag access operation (see 9.3).

DSFID is defined as an array which can be zero length and it is therefore optional by default.

Compliance requirement: Compliant Interrogators and Clients shall implement this parameter.

ISO15962ReadResult Parameter

OpSpecID: Unsigned Short Integer

This data field is a number generated dynamically by the Client and supplied with the ISO15962Read parameter associated with this read result. This number is a handle that can be used by the Client to correlate the read results in this parameter with a specific read request.

Memory Bank: Integer

The memory bank specified by the Client is returned here as a convenience for the Client.

Possible Values:

Value*	Definition
1	UII memory bank
3	User memory bank

AFI: Byte Array

This data field returns the tag's AFI contents to the Client.

DSFID: Byte Array

This data field returns the memory bank's DSFID contents to the Client.

OIDs Read: List <ISO15962OIDReadResult>

A list of zero or more ISO15962OIDReadResult parameter entries, one for each requested OID found in the tag's memory bank. If the Client requested to read all OIDs, then this list will include one parameter entry for each OID found in the tag's memory bank.

If the Client requested decoded data and the Interrogator is capable of providing this decoding, then the decoded data values are returned with each ISO15962OIDReadResult parameter.

Read Data: List <ISO15962RawDataReadResult>

A list of zero or more ISO15962RawDataReadResult parameter entries. Each parameter encapsulates an encoded package from the tag's memory bank (as specified in the read request) representing one or more requested OIDs. For instance, if the tag's memory bank's access method is standard data sets, then there may be one ISO15962RawDataReadResult parameter returned for each OID requested. Or, if all of the OIDs exist in a single packed object, then there may be just one ISO15962RawDataReadResult parameter returned representing all OIDs requested.

NOTE The raw data returned is the complete encoded package including both the data item's encoded OID and its data value.

If the Client requested decoded data but the Interrogator is not capable of providing this decoding, then the corresponding encoded package will be returned in a ISO15962RawDataReadResult parameter.

Result: Integer

Possible Values:

Value	Definition
0	Success
1	No OIDs found
2	Non-specific tag error
3	No response from tag
4	Non-specific Interrogator error

*NOTE Memory bank values enumerated here are arbitrary code values assigned to represent the different memory banks in the interface specified by this part of ISO/IEC 24791. For convenience, the decimal values of 0,1,2,3 were chosen as code values to correspond to the air protocol memory bank binary values of 00_2, 01_2, 10_2, 11_2 that are specified by ISO/IEC 18000-6. In the case of this parameter (ISO15962ReadResult), only memory banks UII and User Memory apply.

9.4.2 ISO15962OIDReadResult Parameter

This parameter, ISO15962OIDReadResult, is used to report the results of reading individual ISO/IEC 15962 data items.

Compliance requirement: Compliant Interrogators and Clients shall implement this parameter. Interrogators MAY omit support for the Decoded Data field of this parameter.

ISO15962OIDReadResult Parameter

OID: ObjectIdentifier

The data item's object identifier.

Count: Unsigned Short Integer

The number of instances that the OID was found in the tag's memory bank.

NOTE If the Find Duplicates field in the associated ISO15962Read parameter is set false, then the Count value may be less than the actual count that exists on the tag. However, the Count value will be at least one if one or more data items with the corresponding OID exists on the tag.

Decoded Data: String [optional]

If the Client requested decoded data and the Interrogator is capable of fully decoding the data item, then the decoded data is returned here as a Unicode string. If the decoded data is not returned, then an empty string will be returned. The structure, syntax, and valid characters of the decoded data is dependent on the data item's OID and is out of the scope of this part of ISO/IEC 24791. This information is under the jurisdiction of the entity that registered the OID.

Result: Integer

Possible Values:

Value	Definition
0	Success
1	Object data not read
2	Non-specific tag error
3	No response from tag
4	Non-specific Interrogator error
5	Unable to decode the data item

9.4.3 ISO15962RawDataReadResult Parameter

This parameter, ISO15962RawDataReadResult, is used to report the results of reading individual ISO/IEC 15962 data items. It reports one encoded package. This is the raw data returned that includes both a data item's encoded OID and its data value. The encoded package returned represents at least one data item but it can also represent multiple data items. For instance, if the tag's memory bank's access method is set to packed objects, then the encoded package will include one or more data items.

Client software is responsible for decoding the encoded package returned, parsing the individual data items, decoding OIDs, and decoding compressed data values.

NOTE If additional information such as edit flags are required, then the Client software shall read raw data.

Compliance requirement: Compliant Interrogators and Clients shall implement this parameter.

ISO15962RawDataReadResult Parameter

Offset: Unsigned Short Integer

The bit offset where the ISO data item begins in the tag's memory bank.

Data: Byte Array

The encoded package (i.e., raw binary data) as found in the tag's memory bank.

NOTE The length of this array corresponds to the length of the data read from the tag's memory bank. For each data item encoded, the raw data includes both the data item's encoded OID and its data value.

Result: Integer

Possible Values:

Value	Definition
0	Success
1	Non-specific tag error
2	No response from tag
3	Non-specific Interrogator error

10 LLRP Extensions Binary Encoding

10.1 General

This clause and its sub-clauses define the specific contents of the binary transfer syntax of the extensions to LLRP as described in clause 9 above. See the LLRP specification **[LLRP]** for details about the encoding of LLRP parameters and fields within parameters. As defined by the LLRP specification, this binary transfer syntax is defined as an application protocol that is layered over TCP/IP. It can also be applied to any transport protocol that provides a reliable transport for protocol data units (i.e., message frames).

10.2 ISO15962Read Parameter

10.2.1 General

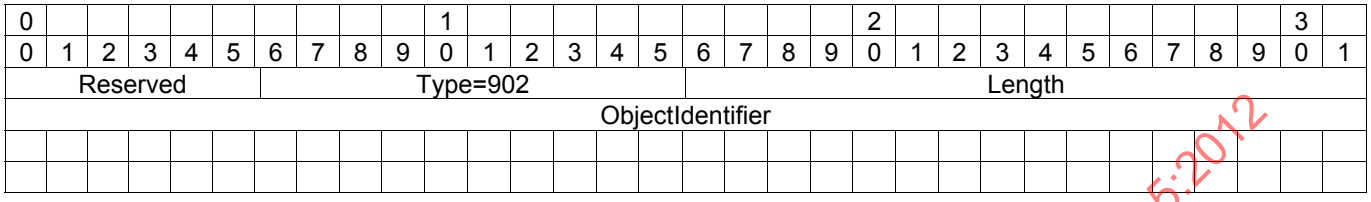
0										1										2									3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 901										Length															
OpSpecID																MB		D		Reserved				ReturnDataControlCode							
OID Parameter (0-n)																															

Abbreviations

MB – Memory Bank
D – Find Duplicates

See 9.3 for the abstract definition of this parameter.

10.2.2 OID Parameter

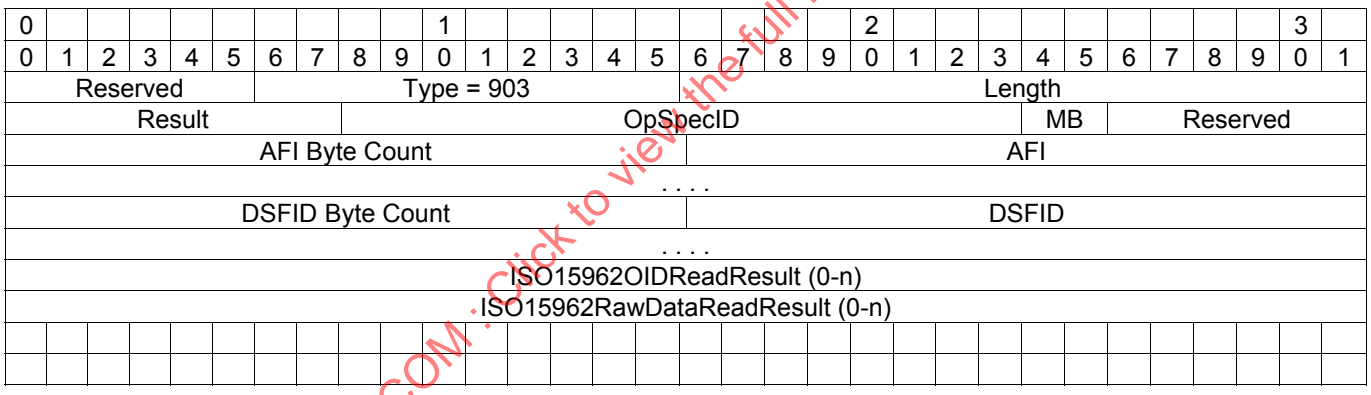


NOTE The ObjectIdentifier is encoded as a UTF-8 string which is a variable length field. The length of the ObjectIdentifier (i.e., number of bytes) is established by Length – 4. 4 is the size of the parameter header which is included in the Length value. This encoding scheme (i.e., a variable length array running to the end of the parameter) is called "Bytes To End" which is supported by open source LLRP parsing tools referenced in Annex A.

See 9.3.2 for the abstract definition of this parameter.

10.3 ISO15962ReadResult Parameter

10.3.1 General



Abbreviations

MB – Memory Bank

NOTE AFI and DSFID are defined as Byte Arrays. A Byte Array is encoded with a leading 16-bit byte count followed by "N" data bytes where "N" is established by the leading byte count value.

See 9.4 for the abstract definition of this parameter.

10.3.2 ISO15962OIDReadResult Parameter

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 904										Length															
Result										Count														OID Byte Count [15:8]							
OID Byte Count [7:0]										OID																					
. . . .																															
DecodedData Byte Count																DecodedData															
. . . .																															

NOTE OID and DecodedData are character strings that are encoded for transfer as UTF-8 strings. UTF-8 strings are variable length and therefore they are encoded as Byte Arrays. A Byte Array is encoded with a leading 16-bit byte count followed by "N" data bytes where "N" is established by the leading byte count value. The tag data is decoded following ISO/IEC 15962 rules. If the DecodedData parameter is present, it will always be represented as UTF-8 for transfer to the client endpoint. If the tag data was encoded with no directory, directory or tag data profile access methods and with ISO/IEC 8859-1 characters in the range 80_h to FF_h, then an implementation of this part of ISO/IEC 24791 shall convert these characters to their equivalent UTF-8 characters. As a specific example, if the client application requires ISO/IEC 8859-1 character representation, then it shall convert UTF-8 to ISO/IEC 8859-1 characters. Annex B is an informative example of algorithms that convert data between ISO/IEC 8859-1 and UTF-8 character strings.

See 9.4.2 for the abstract definition of this parameter.

10.3.3 ISO15962RawDataReadResult Parameter

0										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Reserved						Type = 905										Length															
Result										Offset														Data Byte Count [15:8]							
Data Byte Count [7:0]										Data																					
. . . .																															

NOTE The Data field is a variable length Byte Array. This field is encoded with a leading 16-bit byte count followed by "N" data bytes where "N" is established by the leading byte count value.

See 9.4.3 for the abstract definition of this parameter.

Annex A
(informative)

LLRP Open Source Project References

Support of the interface defined by this part of ISO/IEC 24791, known at the time of this writing, are three open source projects:

- LLRP Toolkit (see www.llrp.org). This project includes a machine-readable definition of the LLRP transfer syntax. It also includes tools that can be used to capture and exchange LLRP session contents.

- WireShark (formerly Ethereal. See www.wireshark.org). This project offers a LAN analyzer tool with an LLRP plug-in that can parse and decode LLRP protocol data units (PDUs).

- LLRP Virtual Reader (see www.rifidi.org). This project offers an RFID reader simulator that is accessible via the LLRP interface.