INTERNATIONAL STANDARD

ISO 17532

First edition
2007-08-01

Stationary equipment for agriculture — Data communications network for livestock farming

Matériel fixe pour l'agriculture — Réseau de communication de données pour fermes d'élevage

Reference number
ISO 17532:2007(E)

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

COPYRIGHT PROTECTED DOCUMENT

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 iv

1 Scope 1

2 Normative references 1

3 Terms and definitions 1

4 Abbreviated terms 4

5 General 6

6 Technical requirements and recommendations 6
6.1 Basic requirements 6
6.2 Connectors 6
6.3 Cables 7
6.4 Transport protocol 7

7 Network livestock farming communication 7
7.1 Farm use 7
7.2 Internet use 7
7.3 Multicast communication 8
7.4 TCP connections 8
7.5 Addressing devices 13
7.6 Configuration of network devices 13
7.7 Network management and monitoring 13
7.8 Communication steps 13
7.9 Communication levels 15
7.10 Communication functions 19
7.11 ADIS extensions 23
7.12 XML/ADED 28

8 Data dictionary (DD) 31
8.1 General 31
8.2 DD elements essential for network livestock farming 31
8.3 CODE SET and lists 32
8.4 Entities for describing communicated data contents 32

9 Electrical specifications 32

10 Mechanical specifications 32

Annex A (normative) Data elements 33

Bibliography 71

Foreword

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

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

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member 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 shall not be held responsible for identifying any or all such patent rights.

ISO 17532 was prepared by Technical Committee ISO/TC 23, *Tractors and machinery for agriculture and forestry*, Subcommittee SC 19, *Agricultural electronics*.

INTERNATIONAL STANDARD **ISO 17532:2007(E)**

Stationary equipment for agriculture — Data communications network for livestock farming

1 Scope

This International Standard specifies a protocol for the automatic and interactive communication and control of computer systems used in livestock production. It supports communication within the livestock production as well as across the Internet.

While it defines the basic protocol for data exchange, it uses generic data structures so that the protocol is extendable regarding future requirements.

The protocol is applicable only to simple and/or clearly defined entities.

This International Standard deals with the networking of those services used for livestock production which are provided by the devices in systems. It is not applicable to communication within subsystems.

The syntax of the transported data is based on the ADIS and ADED standards as defined in ISO 11787 and ISO 11788; alternatively, XML/ADED can be used as described in this International Standard. Like the ADIS standard specified in ISO 11787, it is implicit that the syntax is not intended for real-time data interchange.

2 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 3166-1, C*odes for the representation of names of countries and their subdivisions — Part 1: Country codes*

ISO 4217, C*odes for the representation of currencies and funds*

ISO 11787, *Machinery for agriculture and forestry — Data interchange between management computer and process computers — Data interchange syntax*

ISO 11788-1, *Electronic data interchange between information systems in agriculture — Agricultural data element dictionary — Part 1: General description*

IEC 60204-1, *Safety of machinery — Electrical equipment of machines — Part 1: General requirements*

IEC 60529, *Degrees of protection provided by enclosures (IP Code)*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 11787 and ISO 11788-1, and the following, apply.

3.1
bluetooth
industry standard for the wireless cross-linking of devices with small range

3.2
device
any machine or component which is connected to computer systems used for livestock production

3.3
MAC address
hardware address that uniquely identifies each node of a network

NOTE MAC is an acronym for media access control.

3.4
system
data processing or control component at the computer systems used for livestock production

3.5
subsystem
division of a system, which itself has the characteristics of a data communication system

3.6
network
interconnection of three or more communicating components which exchange data via a network

3.7
network management
execution of a set of functions required for controlling, planning, allocating, deploying, coordinating and monitoring the resources of a network

3.8
datagram
fundamental unit of information carriage in all modern computer networks

NOTE It consists of a header, which contains the information needed to get the datagram from the source to the destination, and a data area, which contains the data content (messages).

3.9
location
agricultural business and the various subsystems (shed, compartment, bay, place) of a farm

3.10
location address
sequence of sub-addresses separated by dots

NOTE The sequence begins with the farm number (15 N). The farm number starts with the numeric ISO Country Code (ISO 3166/3 N) and is immediately followed by a unique national farm ID (12 N).

3.11
server
program in a device which provides a socket with which a client can link up to exchange ISO 17532 messages and services

3.12
client
program in a device that is used to contact and obtain data from a server program on another or the same device

NOTE An ISO 17532 client is designed to work with ISO 17532 server programs only.

3.13
parameter group
group of related parameters which can be used to configure machine configurations for ISO 17532 communication

3.14
parameter class
contains one or more parameter groups which are classified by a special function or task

3.15
parameter identifier
item type for machine configuration

3.16
DD entity
data element definition from the data dictionary (DD)

3.17
DD entity number
number used to identify a data entity in a data dictionary (DD)

3.18
DD item
data details of the data elements from the data dictionary (DD)

3.19
code set
fixed set of states with defined contents

3.20
multicast
delivery of information to multiple destinations simultaneously, using the most efficient strategy to deliver the message over each link of the network

NOTE Delivery is performed only once and copies are created only when the links to the destinations split. “Multicast” is used to refer to an IP multicast, which is a protocol for efficiently sending to multiple receivers at IP networks at the same time.

3.21
multicast address
address of a multicast connection

3.22
transmission control protocol
TCP
connection-oriented, reliable delivery byte-stream transport layer communication protocol

3.23
user datagram protocol
UDP
minimal message-oriented transport layer protocol

3.24
Internet protocol
IP
data-oriented protocol used by source and destination hosts for communication data across a packet-switched Internetwork

3.25
virtual private network
VPN
private communication network used within a company or by different companies or organizations communicating over the Internet

NOTE Secure VPN use cryptographic tunnelling protocols to provide the necessary confidentiality, sender authentication and message integrity to achieve the privacy intended.

3.26
unity
comparison figure with which the values of defined items are expressed

3.27
handle number
unique number used to identify a transaction or a named query

3.28
device ID
unique device number constructed by the MAC address ID

3.29
port
network port
interface for communicating with a computer program over a network

NOTE Network ports are numbered. UDP and TCP will attach a port number to the data sent, which is used by the receiving network component to determine to which application on the device data should be sent.

3.30
session
period of communication activity

NOTE Reliable, interactive transfer of data between two devices in ISO 17532 communication is called a “session”.

3.31
transaction
one interaction between client and server

3.32
session end line
termination
end of a session

4 Abbreviated terms

ADIS	agriculture data interchange syntax
ADED	agriculture data element dictionary
AN	alphanumeric
C	conditional
DA	data authentication
DD	data dictionary
DHCP	dynamic host configuration protocol

DNS	domain name server
DF	data line of faulty data + error item + impact item
DN	definition line type of status normal (N)
EDI	electronic data interchange
FTP	file transfer protocol
IAONA	international automation open networking alliance
IEEE	institute of electrical and electronic engineers
IETF	Internet engineering task force
IP	Internet protocol
K	key data element
LAN	local area network
M	mandatory
MCC	multicast communication
N	numeric
NLF	network livestock farming
O	optional
PIG	industrial Ethernet planning and installation guide
PO	processing instruction open
PP	processing instruction pending
PPP	point-to-point protocol
PR	processing instruction result line
RJ45	registered jack type 45
SBC	session-based communication
SN	sequence number
TCP	transmission control protocol
TN	termination line with status normal (N)
UDP	user datagram protocol
URI	uniform resource identifier
URL	uniform resource locator
UTC	universal time (also known as Greenwich mean time)
VA	value line for authentication
VE	value line error handling information

VF	value line faulty data + error code + impact code
VN	value line type and status
VPN	virtual private network
WLAN	wireless local area network
XML	extensible markup language
ZN	end session

NOTE For descriptions of line type and status character abbreviations, see Figure 8.

5 General

Networks of computer systems are necessary in livestock farming for controlling various processes (e.g. feeding and climate), environmental balance (field yard balance), and comprehensive environmental and animal protection, as well as for economic reasons.

In order to use services over a networked infrastructure, information must be transported. For this, the way in which the information is to be communicated between the communication partners needs to be known in advance. The purpose of this International Standard is to provide a protocol that serves this need.

IP is used as the basis for ISO 17532 communication and TCP streams are used to ensure reliable communication. For management and short messages, UDP datagrams are multicasted.

For the notation of the description of the ADIS syntax, see ISO 11787.

As the physical layer for the IP, the Ethernet as defined in IEEE 802.3x is the most commonly used today, and this International Standard is designed to support it. IP packets can be transported using wired or wireless communication.

This International Standard defines the requirements for the physical connections and the data communication in a network used for livestock production. The requirements apply for the devices connected directly to this network.

6 Technical requirements and recommendations

6.1 Basic requirements

Fast data communication between the components shall be guaranteed by the devices involved.

If real-time communication is needed, logical separated network segments shall be configured for communication in real time within the important segment.

A structured cabling or wireless connection shall be provided.

6.2 Connectors

The use of RJ45 connectors is recommended. Depending on the environmental conditions (light or heavy duty), the connections shall comply with protection classes IP 20 or IP 67 in accordance with IEC 60529.

6.3 Cables

The devices of the manufacturers shall be star, bus or ring-shaped connections. A four-wire cable is sufficient for the data connection.

For the wiring of the stable net, cable types corresponding to the IAONA recommendations (PIG, release 4.0, 2003) should be used. The transmission speed is determined as a minimum of 10 MB/s for copper wire connections. For the wiring within the barn area, safety class system IP 67 (heavy duty) should be used. In less susceptible environments (stable office), IP 20 (light duty) is sufficient. As basic requirements for the physical layer, the recommendations of the IAONA shall at least be met.

6.4 Transport protocol

Both TCP/IP and UDP are used — TCP/IP for reliable connections, UDP for multicast addressing communication. Components (WLAN, Bluetooth devices) that can be connected to the farm network via TCP/IP and UDP, and which conform to corresponding IEEE standards, are supported. Figure 1 shows this architecture graphically.

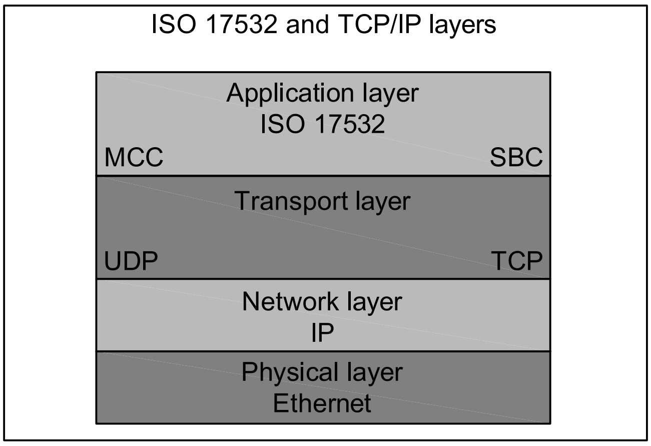


Figure 1 — ISO 17532 and TCP/IP layers

7 Network livestock farming communication

7.1 Farm use

There are no limitations on ISO 17532 communication within livestock production, i.e. within one physical network. A local network can be connected via the Internet using VPN technology, whereby the network can be extended beyond the borders of one farm (physical network).

7.2 Internet use

The coupling of devices using multicast communication over the Internet is not supported. The data exchange between farm devices and Internet partners shall be organized by special services.

NOTE In Internet connections, only TCP connections are supported in ISO 17532 communication. TCP packets can be transported using a variety of physical layer protocols, e.g. PPP (modem use) or Ethernet.

7.3 Multicast communication

The multicast address 224.111.234.123 and port numbers 2434 for ADIS/ADED and 2435 for XML/ADED should be used in local networks. If other multicast addresses are used, they must be configured manually in all the devices.

Due to limitations in the transport of UDP multicast datagrams via the underlying physical layer (e.g. Ethernet), the data area of a datagram must not exceed 1 024 B. The sender of a datagram with ADIS lines also needs to ensure that the datagram length of the transformed XML data does not exceed the length of 1 024 B. Each UDP multicast datagram starts with 8 B for a header containing information about the ISO 17532 MCC communication. Bit 0 of the first byte is set when the datagram is transferred by the transformer to the other MCC address. If bit 1 is set, there is no need to transform the datagram. To keep this sort of message small, no header lines are sent in the datagram. In multicast communication, no header data with information on the data dictionary used, etc. can be sent, so the recipients of the data must be able to react flexibly to the data structure.

In UDP datagrams, a value line (ADIS and XML/ADED) is preceded by a definition line referring to the same entity.

NOTE 1 Management functions and the distribution of simple messages can be easily conducted using UDP as multicast messages without knowing the network address of every single device.

NOTE 2 UDP communication does not acknowledge receipt of UDP packets and is not absolutely secure.

NOTE 3 Owing to the nature of multicast, every device connected to an ISO 17532 network receives the messages sent to the commonly agreed multicast address. It is then the responsibility of the administrator to check bus load and time behaviour.

7.4 TCP connections

7.4.1 General

TCP connections shall be used when it is necessary to ensure that no information is lost or mixed up. The initiating device is called the “client” and the answering device, which was listening for a connection, is called the “server”. Therefore, TCP connections are always point-to-point connections (by default, the communication partners use port number 2434). If, on one computer, several SBC services are offered by several programs, it is necessary during publish/subscribe to arrange for alternative ports. This is the purpose of specifying the port number (item number 901044) in the entities service request (990110) and service inquiry reply (990109).

For transforming data streams between XML/ADED and ADIS/ADED, conversion software is needed.

NOTE When implementing software that handles TCP connections or network connections in general, it is necessary to consider using a timeout to close a connection when the communication partner does not send any messages or no longer responds. However, it has to be kept in mind that a request to search certain data can cause a database system to require quite some time. So timeouts have to be selected carefully. For very long-lasting orders, asynchronous transactions are a means of carrying out the order without the need of setting very long timeouts.

7.4.2 Session

Each session is divided into separate transactions and is always carried out using a TCP connection between two devices (point-to-point communication).

A session starts with information on the authentication of the client. If the authentication stage is left out by the client, the session starts with the header line. Authentication and header lines are encapsulated in one transaction each. Whether a login is necessary depends on the demands of the devices. It can be negotiated during publication of the services in the boot-up stage of a device. The header line supplies basic information on how the following data in this session has to be interpreted in terms of the DD version that was used to compile the data of the session.

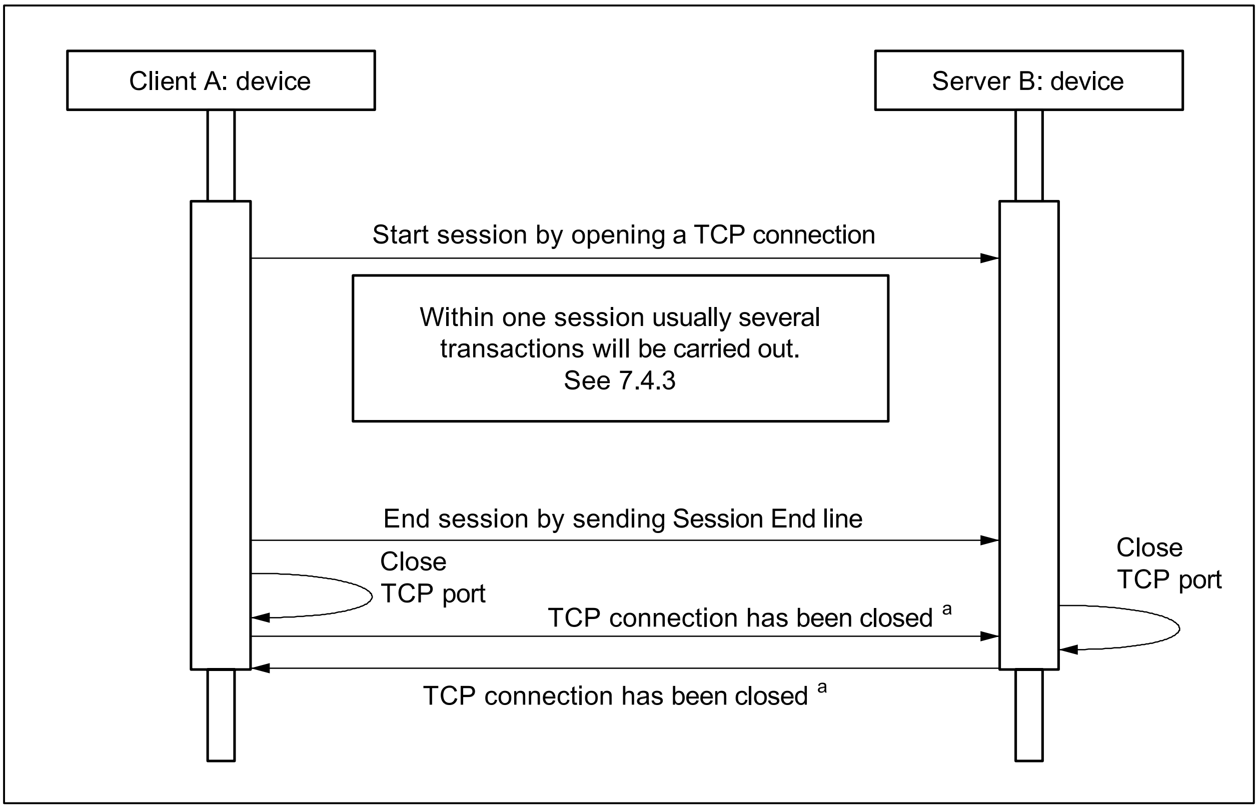
A session consists of 1..n transactions and ends with a Session End line. Generally, the client sends a Session End line to the server to make sure the server takes note that the TCP connection is closing. Otherwise, there could be a lock-up of the used server TCP port for several minutes (depending on the operating system) before the same TCP port can be used again.

Within a transaction, a variety of functions can be carried out:

- authentication;
- header lines;
- sending or receiving data;
- issuing a search request;
- executing a named query;
- executing a processing instruction.

See Annex A for a detailed list and description of these functions.

Figure 2 shows the flow of the described session.


Client A: device
Server B: device
Start session by opening a TCP connection
Within one session usually several transactions will be carried out. See 7.4.3
End session by sending Session End line
Close TCP port
Close TCP port
TCP connection has been closed [a]
TCP connection has been closed [a]

[a] Both these messages are part of the operating system's implementation of the TCP stack.

Figure 2 — Session

7.4.3 Transaction

Each transaction is separate from the next. Figure 3 shows that device A (client) starts the transaction and device B (server) processes the sent data and sends back either results, error information, comments or nothing.

The client is shown as initiating a transaction by sending the starting line of one of the functions specified in 7.4.2.

a) Sending a Definition Line starts the function “sending data”.

b) Search Request starts with either a Search or a Request Line.

c) Posting a Named Query is started by sending a Query Line.

d) Processing instructions are started by a Processing Instruction line.

e) Login (Authentication) and Header are handled alike.

f) Header lines are formulated as special transactions.

It is not possible to use this feature by means of multicast messages.

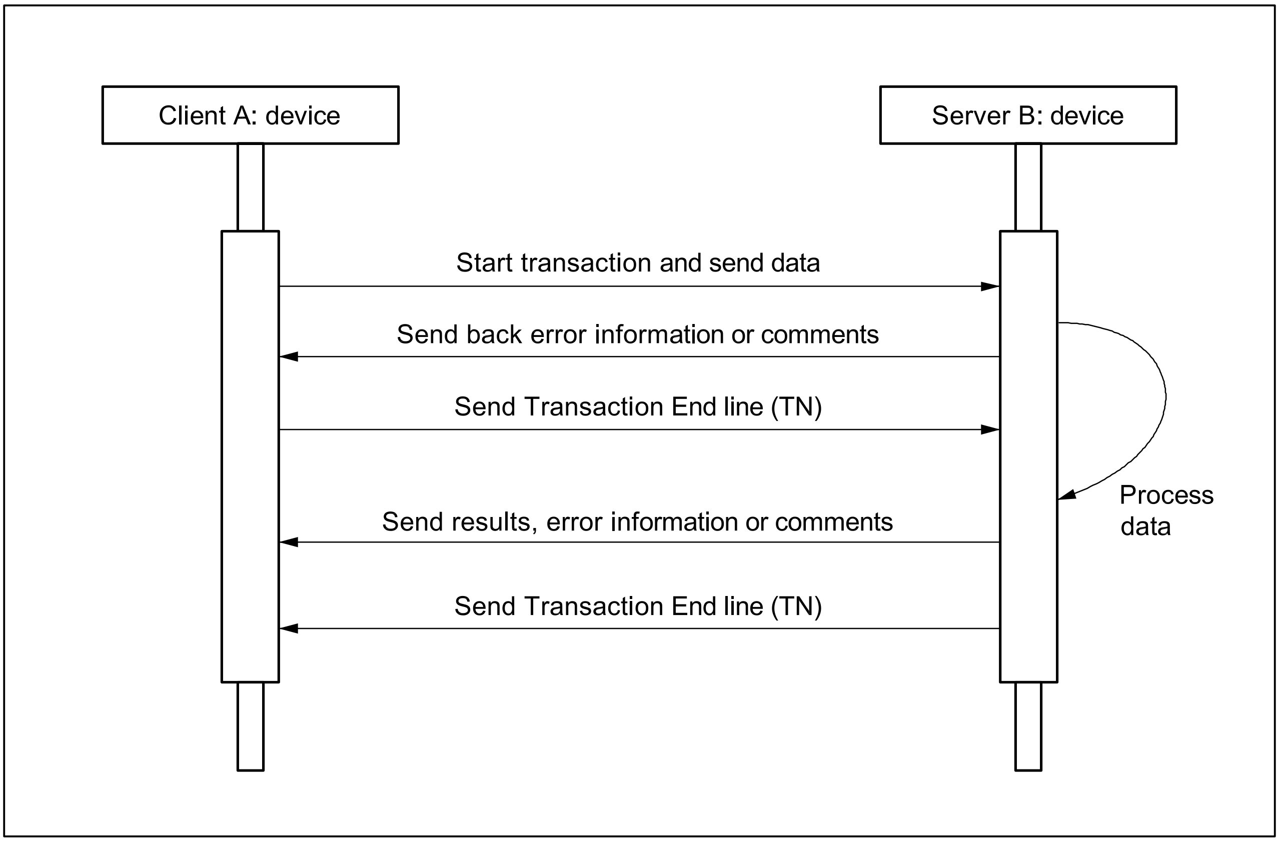

Client A: device
Server B: device
Start transaction and send data
Send back error information or comments
Send Transaction End line (TN)
Process data
Send results, error information or comments
Send Transaction End line (TN)

Figure 3 — General handling of transactions

In contrast with the start of a transaction, the end is always marked with the same line, the Transaction End line. Because the client initiates a transaction, it is up to the client to send the Transaction End line after issuing all necessary information. The receiver may have to process the transaction. The receiving end (server) then needs to interact with the client by sending back the requested information or, possibly, any error codes, comments or the like. After processing the transaction that was sent by the client and sending back any information (requested or not), the server then sends back the Transaction End line.

Before starting a new transaction, the client (initiator of the connection to the receiving end) has to wait for the handshake, i.e. the reception of a Transaction End line from the server.

If a transaction cannot be handled correctly for any important reason by one of the communication partners, the session can be cancelled by one of the partners by sending a Session End line. The current transaction is then cancelled, too. After cancelling a session, the TCP connection must be shut down immediately. A TCP connection may not be shut down generally without issuing a Session End line due to technical details of the IP. In certain circumstances it is not possible to open a connection to the same port for a certain time after irregularly ending a TCP connection.

On the one hand, the transaction partners can generally decide whether they want to react to the incorrect transaction end with a rollback or by sending more data. On the other hand, a regular Transaction End indicates that the services within the transaction have definitely been carried out correctly. An exception, of course, is the asynchronous transactions because they are carried out while the initial transaction, session and TCP connection are shut down. Asynchronous transactions (see 7.4.4) are identified by a handle number for future reference.

7.4.4 Asynchronous transactions

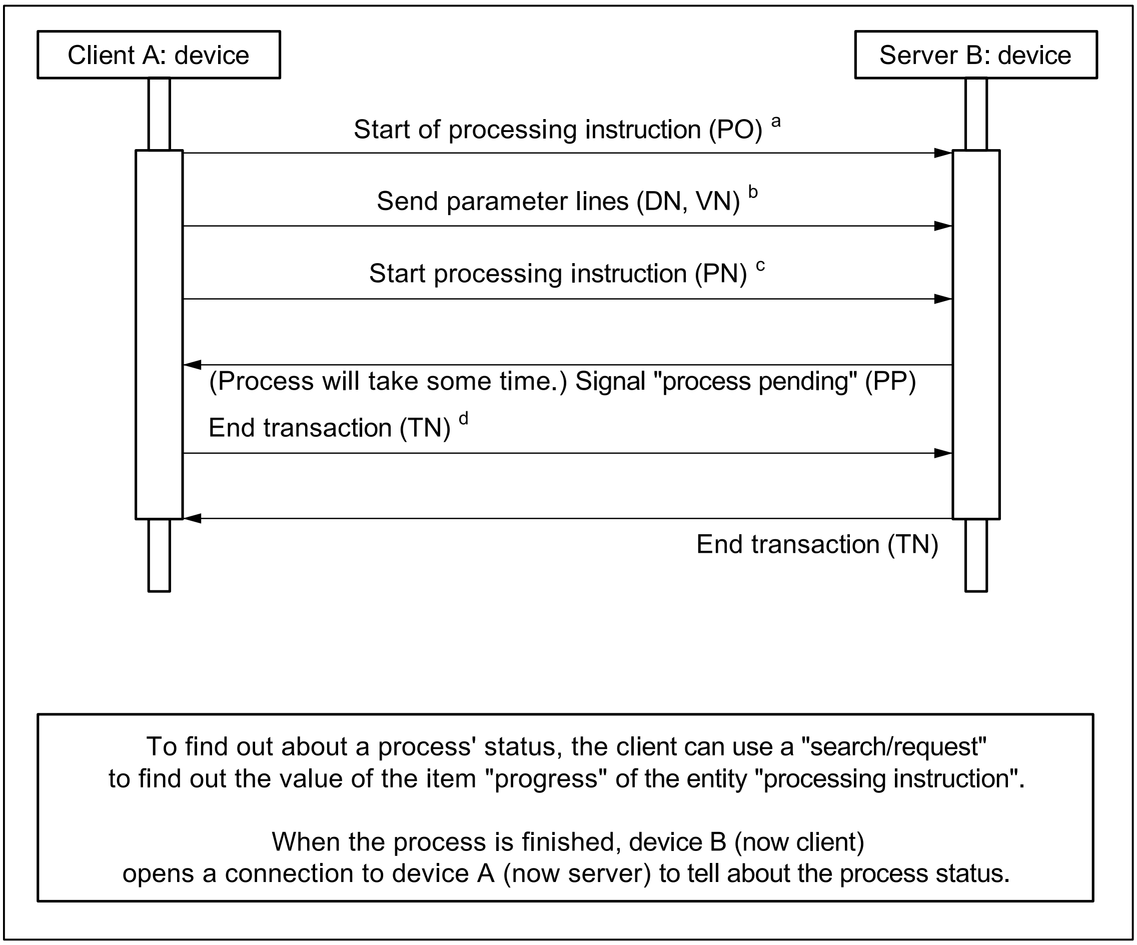
Asynchronous transactions are special in the sense that they are carried out while there is no communication connection open between the requesting and the executing end of the interaction of two devices. Immediately in the login, but also in the query or processing instruction, the client can issue a URI to which the server should transfer asynchronous results. If this URI is not given, a TCP connection is created from the server to the default port of the client as a reverse channel as soon as the server wants to send back the asynchronous results.

To refer to the issued function, the asynchronous result carries the same, unique handle number.

The operator finds out whether the data needs to be sent, e.g. as an e-mail by parsing the URI. Within the URI “e-mail” is one of the predefined URI schemes that also gives information about the access mechanism (how to send an e-mail).

NOTE RFC 2396 gives details on URI and URI schemes.

The new ADIS status characters presented in 7.11.3 inform the client that data will be sent back asynchronously. Figure 4 shows in detail how asynchronous transactions are handled.


Client A: device
Server B: device
Start of processing instruction (PO) [a]
Send parameter lines (DN, VN) [b]
Start processing instruction (PN) [c]
(Process will take some time.) Signal "process pending" (PP)
End transaction (TN) [d]
End transaction (TN)
To find out about a process' status, the client can use a "search/request" to find out the value of the item "progress" of the entity "processing instruction".
When the process is finished, device B (now client) opens a connection to device A (now server) to tell about the process status.

[a] Start of communication, such as authentication, is not shown in this diagram.

[b] DN, VN lines that, e.g., describe the entity “mix jab”.

[c] Return transaction ID identifying the process.

[d] The server can also respond with a PR line (“process result”). This tells the client that the request process is already finished.

Figure 4 — Asynchronous transactions

7.5 Addressing devices

7.5.1 TCP addressing

A device in the network has to be addressed using its IP number. This is given to the device by the configuration or by a DHCP server.

7.5.2 Addressing with URI

Transmitted data is often only part of a data complex where partitions of the data are kept in a number of distributed sources. ADIS already provides a way to include ADIS data stored in external files. The scope of this reference shall be extended to the extensive concept of URL and URI used in Internet communication.

An URI can be used to define the unity of an entity or as a part of an include line in the communication.

NOTE FTP is not supported because it cannot be assumed that all farm businesses are always accessible in the Internet under their own domain.

7.6 Configuration of network devices

The device must publish the device description with the device addressing URI. The configuration service (configurator) then establishes a connection to the device and sends a configuration processing instruction with all the necessary parameters like device-location linking (990104), machine-location-service linking and generic machine configuration (990114).

NOTE 1 Configuration of ISO 17532 devices can be handled manually by the technicians who install the devices. This International Standard offers elements which allow configuration of a device with processing instructions, which include the setting of parameters for the configuration of the device.

NOTE 2 The device configurator can be a web application, which can be controlled by a browser.

7.7 Network management and monitoring

There is a time limit on subscribing to data. If device A will not update its subscribing, device B stops sending after 24 h (default). This default can be changed using the item “Time period for delivery” (901011). Because multicast communication is not safe, a device shall re-send service information (i.e. climate data) after the “Time period for delivery” even if there is no change in data value. One device has to be the time server. This must be configured. This device will send the entity “Time synchronization” (990116) every hour by default. The entity contains the local time and the UTC. Alarm messages contain the source device, a timestamp, an alarm number and a describing text. A service listening for special alarm messages has to manage the alarms (e.g. text message to the farmer, triggering an acoustic or visual signal) according to the needs of the farm requirements. An alarm (entity 990117) is repeated under MCC until an alarm receiver acknowledges processing of the alarm. The answer is sent back with the status character Result. Monitoring and visualization tools can require that all devices in the farm network publish the device description, device location, device-location linking, machine-location-service linking, device-location linking and machine-location-service linking entity, in order to provide services for network monitoring.

7.8 Communication steps

The following list illustrates the basic steps in ISO 17532 communication. Steps a) to e) can be carried out using simple multicast messages. The final step, f), completes the list by adding the general possibility of transferring any data using a session.

NOTE 1 Possible ISO 17532 communication ranges from simple messages that are not guaranteed to be received by all the listeners (MCC) to session-based communication for reliable transfer of possibly large amounts of data.

a) **Publish (MCC)**

A device newly registered in the network after starting up publishes its services (Service-Publish-Entity).

b) **Request (MCC)**

After publishing its own services, a device may look for desired services within the network (Service-Inquiry-Entity).

c) **Publish-On-Request (MCC)**

When inquiries about certain services are sent to the network devices by means of multicast messages [step b)], then any device that offers such a service should answer this request (Service-Inquiry-Reply-Entity).

d) **Event messages (MCC)**

Devices send specific messages triggered by events or alarms. Any device should be able allow other devices on the network to configure when or why it should send these messages. Otherwise, the devices need to be manually configurable in respect of when and why they send these messages.

e) **Subscribing to services (preferably MCC)**

Whenever a device needs to rely on the data of a certain service, there is the possibility of requesting a subscription to the service (Service-Request-Entity). A device can subscribe to the services published under step c) if the device that offers the service can send the data using a session (i.e. TCP connection). The data source is required to send the data according to the conditions defined in the Service-Request-Entity. Apart from the option of using a session, there are other ways to provide the subscribed information, e.g. by e-mail, which can be specified in the Service-Request.

f) **Session-based communication (SBC)**

More extensive communication processes such as Named Queries, Processing Instructions and Search Requests can be handled using a session. Subscriptions to services, steps a) to e), can be handled by sending the subscribed services via a session. A session is one way of transferring data reliably. Therefore, the device that requests a subscription of certain services can rely on the fact that it will not remain unnoticed if the data cannot be transferred to the subscriber for any reason. Apart from using a session, there are other ways to transfer subscribed data (see above, see Service-Request-Entity: Item Type-of-provision).

NOTE 2 MCC is generally not possible using the Internet. It is a means of communication that enables devices within one farm (one local network) to discover each other and the other's services.

Table 1 describes the circumstances under which the above steps are allowed (+) or not permitted (–).

Table 1 — ISO 17532 communication steps

Step	MCC	TCP/IP	Farm network	Internet
a)	+	–	+	–
b)	+	–	+	–
c)	+	–	+	–
d)	+	–	+	–
e)	+	+	+	+
f)	–	+	+	+

7.9 Communication levels

Building on the communication steps given in 7.8 are the protocol's four communication levels, a) to d) below.

NOTE Of course, simple devices only support the ISO 17532 entry level, for example, a temperature sensor that only sends measurements and is not able to process requests for subscriptions to temperature data.

a) **Entry level: MCC via common port and common address**

Communication via this mechanism is designed to achieve various purposes, dispatching messages from manually configured components that send simple messages to the network over time (simple sensors, etc.).

1) Only multicast messages are used for communication.

2) Sensors and simple devices in general regularly send their data in simplified ADIS or XML/ADED syntax to the common multicast address (multicast group).

3) Because multicast messages are sent using UDP, no information unit may exceed the maximum size for a datagram (1 024 B for the data area in a datagram due to Ethernet-specific maximum transmission unit).

4) An information unit consists of a definition line and a value line. Even if the data is sent at high frequency, every information unit (multicast message) is required to contain both the definition line and the value line.

5) The recipients filter out the data relevant to them from the multicast communication using simple filter mechanisms, e.g. only accepting certain entity numbers.

b) **High-level MCC**

At a higher level, multicast messages are used to publish the device's services. The publication of services can be repeated regularly and the devices should be able to answer requests for services (Service-Inquiry-Entity and Service-Inquiry-Reply-Entity). Data should not be sent autonomously. Devices are asked to only send data after receiving at least one request for that service. This includes data sent on a regular basis, such as temperature data.

Figure 5 illustrates a boot-up sequence. The multicast address element shown in the figure should be considered as a distributor: it distributes messages to all devices that are part of that multicast group (for the purposes of this International Standard, all devices that are part of ISO 17532 communication in a local network).

1) Publish services: device A publishes information about the services offered to the multicast group. Any device interested in that information should take further action.

2) Service inquiry: device A is interested in certain services and sends a request about these services to the multicast group.

3) Service inquiry reply: at least one device is able to provide the requested service and sends back an answer to the request.

4) Service data: if data of a service can be sent using multicast messages (which are uncertain to be received to a certain degree), then devices can send these messages to the multicast channel. Device A will receive such messages. They can be a mixture of requested and not requested services.

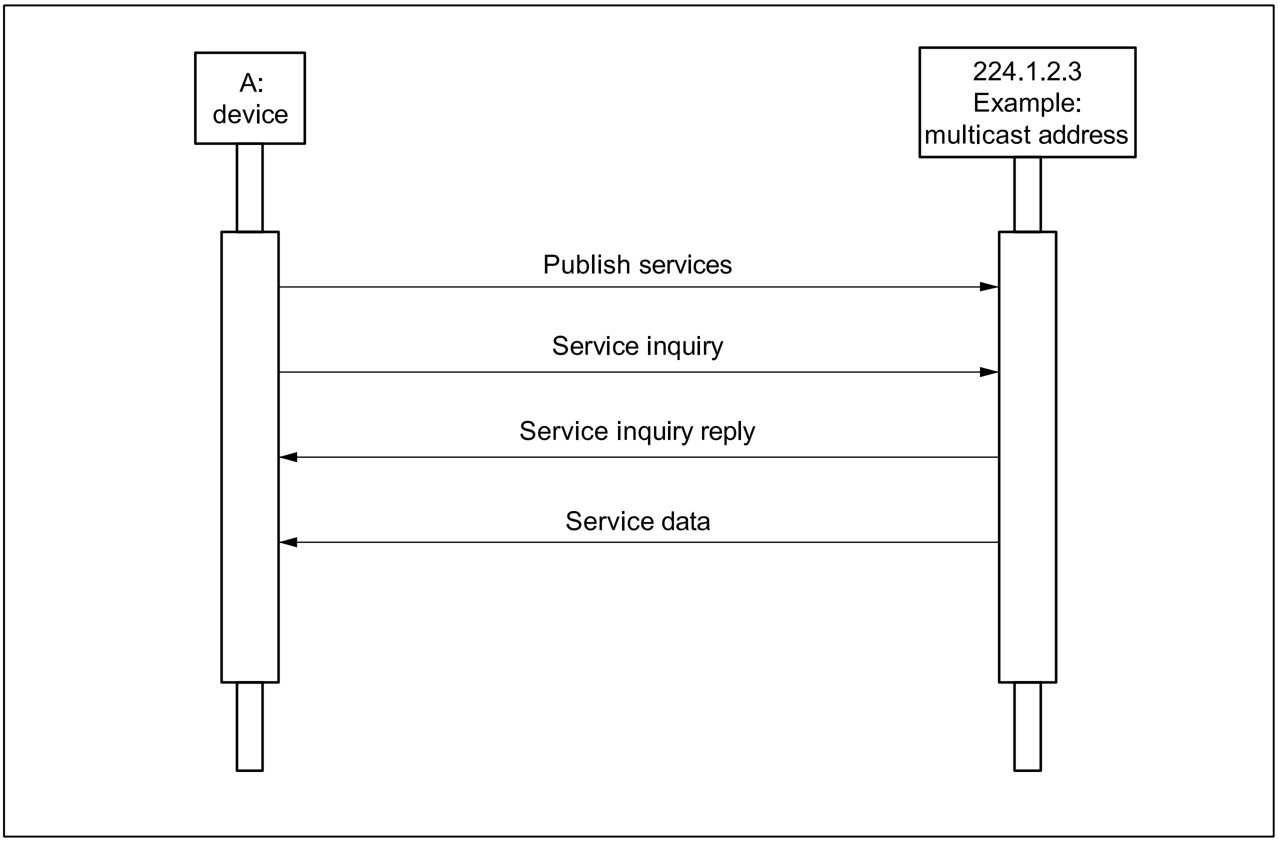


Figure 5 — Boot-up sequence — Type 1

NOTE No message sent to the multicast group is guaranteed to be received by all possible devices in the network, for a number of possible reasons: network failure, device is temporarily offline, and so on. To increase the chance of receiving an answer to a service inquiry, the inquiry should be repeated at least once. To ensure that high network load does not become a problem, messages should only be repeated after a certain amount of time. The time between two multicast messages should also be dependent on a random number. Adding a random number to the waiting time will lower the probability of network congestion.

c) High-level reliable communication using subscriptions

If communication on a high level needs to be reliable, then a session (see above) shall be used. During a session data is transmitted over a TCP connection. This type of connection is responsible in part for the reliability of the connection. The concepts of session and transaction are responsible for making the connection completely reliable.

Figure 6 shows a sequence of a device booting up and subscribing to certain services in which it is interested. The sequence starts with the first three steps, as in the boot-up sequence shown in Figure 5, however, device B, omitted from Figure 5 for reasons of simplification, is shown.

Again, a set of the already mentioned communication steps is used:

1) Publish services (MCC): device A offers services [see b)].

2) Service inquiry (MCC): device A is looking for services [see b)].

3) Service inquiry reply (MCC): device B is offering services on request [see b)].

4) Service request (MCC): device A now starts a request to subscribe to the service. Although the request is posted to the multicast group and all the devices are able to receive it, only device B will register a subscription of a service to device A.

5) The multicast message to request a service, i.e. to acquire a subscription, carries information about the device that was asked for a subscription.

6) Service request reply (MCC): upon successful registration of a subscription, device B sends back a service request reply.

7) Use session to send data: after sending back the acknowledgment of the subscription, the subscribed data is transmitted, if present. Subsequently, data is only transmitted if it has changed.

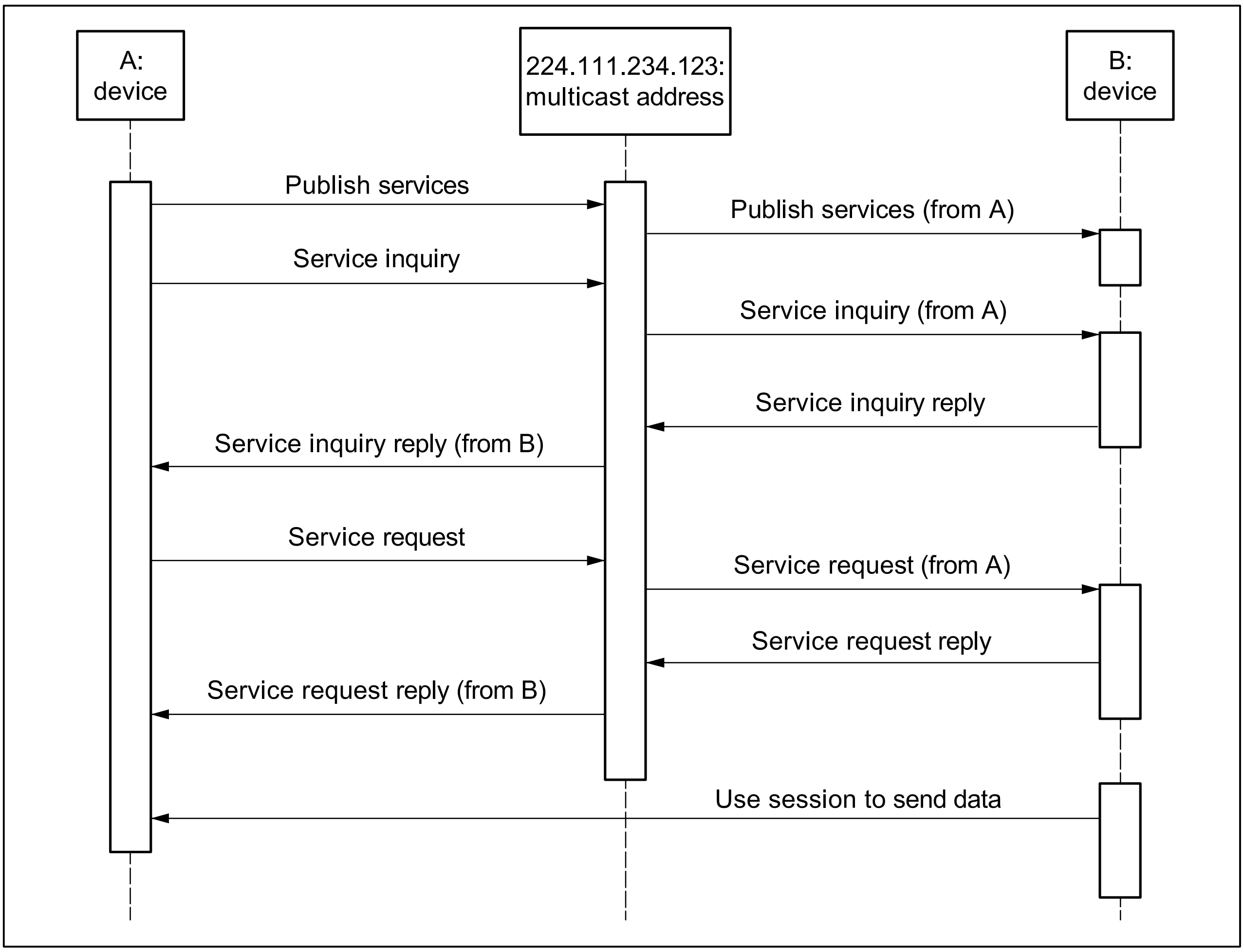


Figure 6 — Boot-up sequence — Type 2

d) **Receiver-controlled communication**

Without using a subscription, a device can inquire whether certain services are available (service inquiry) and then open a session (TCP connection) to the device that offers that service. There are numerous functions that can be used within a session like Search Request, Named query and Processing Instruction (see 7.10). These functions allow the receiver to specify what data should be sent exactly, e.g. by specifying search criteria. Figure 7 shows the detailed flow of a receiver-controlled communication and, again, the first three steps are the same as in boot up sequences types 1 and 2, shown in Figures 5 and 6.

The following communication steps are used.

1) Publish services (MCC): device A offers services [see b)].

2) Service inquiry (MCC): device A looks for services [see b)].

3) Service inquiry reply (MCC): device B offers services on request [see b)].

4) Start session using TCP connection: device A wants to receive certain data from device B. Therefore, a session is used to request and transfer that data immediately (or later, as asynchronous results).

5) Ask device B to send certain data: device A specifies which data is requested. With a search/request or a communication function such as Named Query or Processing Instruction, it is possible to specify the service the data should come from and the search criteria the data should fulfil.

6) Send answer: if possible, device B answers the request for data from a specific service immediately. As mentioned above, there is an exception: if, for example, a long-lasting processing instruction is requested, device B can send an answer to device A immediately saying that the results will be available in the future and that device B will then open a new session to device A to send the results. (Apart from processing instructions, there are other communication functions which can also have asynchronous results.)

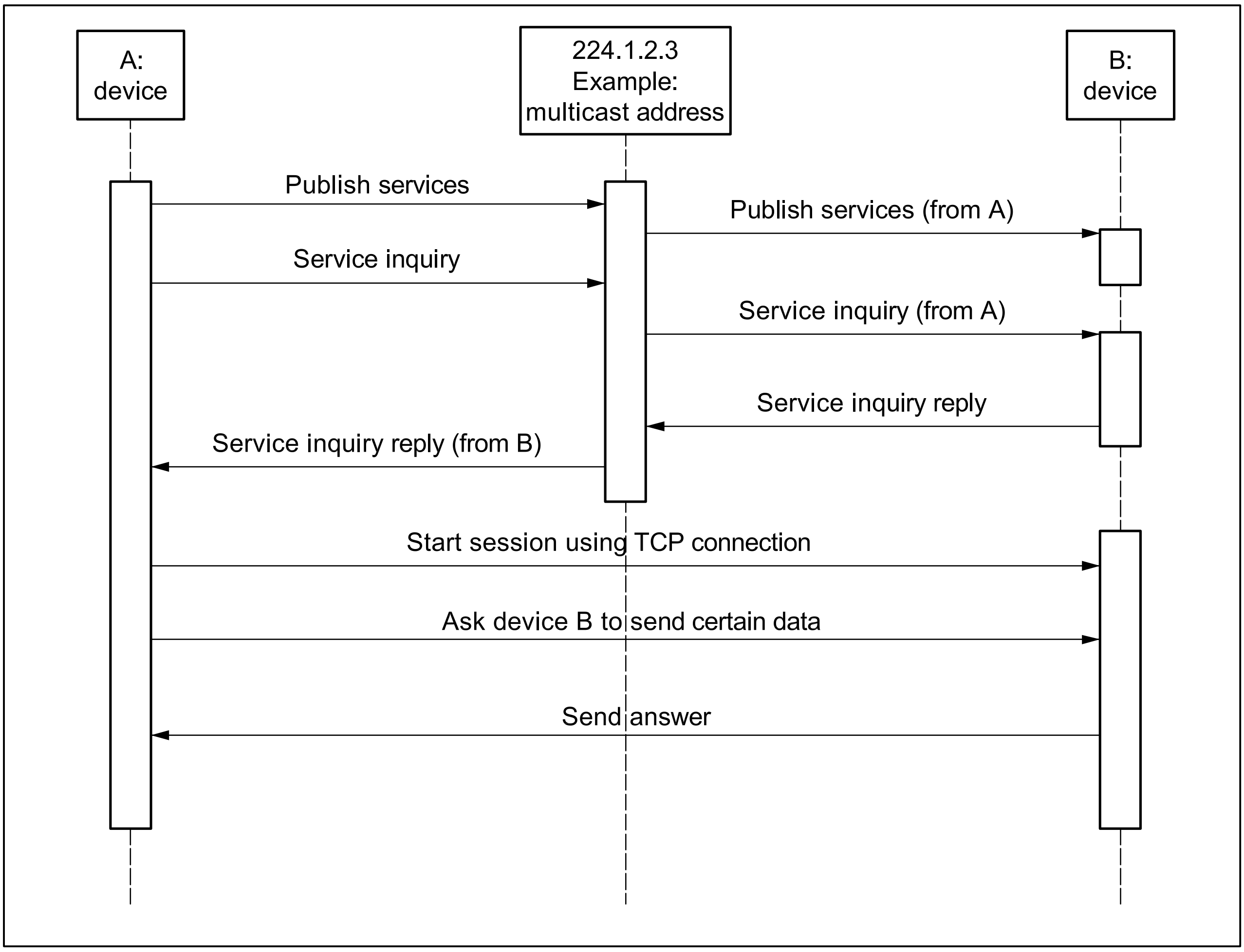

A: device
224.1.2.3 Example: multicast address
B: device
Publish services
Publish services (from A)
Service inquiry
Service inquiry (from A)
Service inquiry reply
Service inquiry reply (from B)
Start session using TCP connection
Ask device B to send certain data
Send answer

Figure 7 — Boot-up sequence — Type 3

7.10 Communication functions

7.10.1 General

This International Standard defines functions for interaction between communication partners in a livestock farming network. Identical functions and basic principles are independent of their concrete realization, either ADIS/ADED or XML/ADED. See the line types and status characters in Figure 8 for an overview of the described functions.

7.10.2 Login procedure

A client partner who wants to establish a connection to the server partner sends a definition and value entity for authentication. The login entity includes at least the following mandatory data items:

- encryption procedure — it shall be the first item in the entity;
- user name;
- password.

The entity definition is not encrypted. The values in the subsequent login values may be encrypted. If so, the encryption procedure item value (which itself must not be encrypted) indicates the encryption procedure that is to be used. A zero value means “no encryption”.

The encryption procedure shall not convert other characters to a CRLF character sequence (Base64 character encoding principles).

If login is demanded, it is performed as an independent transaction. If the login fails because the client is not authenticated successfully by the server, the server answers the rejected login with status “failure” and adds error message data if supported. In case of failure the client may either obtain guest rights or the server will cancel the connection.

The client ends the login procedure with a transaction end and the server acknowledges it also with a transaction end.

The server may cancel the connection at any time by sending a Session End. Upon receiving it, the client then has to close down the connection.

NOTE The optional login realizes the authentication which might be necessary for communication in TCP/IP ISO 17532 sessions, especially for remote systems via Internet. MCC does not require authentication and login.

7.10.3 Header information

The beginning of each communication starts with header data (after a possible login procedure).

In this International Standard, the header entity specifies the following data items, some of these being new to ADED as defined in ISO 11788:

- Data Dictionary Name (international or national);
- Data Dictionary Version (international or national);
- File Creation/Update Date;
- File Creation/Update Time;
- Sender Software Name.

Additional information valid for the whole session may be noted in the header. For example, character sets, currency, addresses of sender or receiver. These data elements are described in Annex A.

7.10.4 Pending state in asynchronous transactions

During any transaction requesting data (named query, processing instructions or classical entity requests), the server may decide that it is not possible or useful to send results such as requested data or acknowledgements immediately. Or else, the communication partners might have previously reached manufacturer-specific agreements, so that, instead of the result, the sender returns a certain pending command. The partners then close the current transaction. The results are sent to the client later in an asynchronous manner. To ensure data is complete in the responding transaction, only one named query, processing instruction or entity request can be sent in one transaction.

7.10.5 Inquire status of pending process

Certain transactions (search request actions, named queries, processing instructions) may be set to a pending state and then closed down by the receiver. The client may ask for the state of the pending action by request of a certain status inquiry entity. Mandatory items are the device ID of the sender and the handle number identifying the pending process call uniquely within the sender. Named queries and processing instructions additionally require the identifier of the named query or processing instruction respectively. Other items indicate the status of the pending process.

The entity can be retrieved by a request as usual and the values identifying the process given in previous search lines. The server will answer with a definition and values for that entity as usual.

This International Standard defines the Status-Inquiry-Entity with an entity number noted in the DD.

7.10.6 Processing instructions

Processing instructions are used to send complex instructions to communication partners.

Each processing instruction is a separate transaction. The recipient of the instruction may execute the instructions immediately or set the transaction to “pending” and answer possible results later, in an asynchronous manner.

Each processing instruction has a unique identifier (either a six-digit number or an alphanumeric identifier), which is defined in a central DD like any entity or named query numbers. A single processing instruction is sent in a single transaction to ensure completeness of the result in a pending transaction.

To definitely mark the start of a processing instruction transaction, a special “open” command is sent, with information about the identifier of the processing instruction, the device ID of the sender and the handle number identifying this processing instruction call uniquely within the sender.

7.10.7 Named Queries

This new concept for high-level database systems given in this International Standard significantly extends the functionality of inquiries. A set of data can be requested with one named query. The search conditions may include any data arguments which are not necessarily part of the requested entities as required for classical search request commands. How these arguments are transferred is part of the new syntax. The search conditions which have to be previously defined for the named query can include more complex conditions and combinations than the “and” combination of the classical search conditions in ISO 11787.

Each named query is identified by a unique number (six digits) or an alphanumeric name similar to function names in programming languages. A type attribute indicates whether an identifier is a name or number.

A named query has a set of 0 to n argument parameters which can be defined as mandatory or optional. One named query is sent in one transaction to ensure completeness of the result in a pending transaction.

The response to a named query includes the named query identifier plus some data to identify the original query and a result set. The result set is defined as one or more data entities with mandatory and/or optional data items for this query. This result data is transferred as for usual data entities. The results are possibly sent later in an asynchronous manner if the transaction was set to pending.

Therefore, the receiver of the named query needs to know the ID of the named query, a handle uniquely identifying the named query call within the sending device and the device ID of the sending device. These are given in the "open" command, which starts the named query transaction.

When the receiver cannot process the named query, an error is reported to the sender by return of the faulty data lines set to "Failure" status and the possible addition of an error code and error severity level (see 7.10.8).

Definitions of named queries are listed in the DD. Depending on the level of organization, they may be defined on a private, national or international level. The international level is preferred. The communication partners involved shall agree on the type, name, arguments and content of the result set.

7.10.8 Error handling

The management of a complex system such as that specified in this International Standard necessitates that error messages should provide more concrete information than just the return of incorrect data marked as faulty. Information about the reason for the error and its context needs also to be provided, as follows.

- Faulty data is returned after being marked as "faulty" data.
- One error code item and one item for the severity level of the error may be appended at the end of the returned data.
- If there is more than one error code to be noted, same value data should be returned several times, each defined with an error code value.
- Additional information necessary for describing the context of faulty data may follow the returned faulty data. In order to distinguish this additional information from normal data lines, a new status is introduced for error information data.
- If the receiver does not support any of these error message concepts, it should at least return the incorrect data marked as faulty and write some error information in a comment.

It is the responsibility of the sender of the original data to determine how to process the returned data.

7.10.9 Database-specific functions

7.10.9.1 General

Deletion, synchronization, insertion, update, storing and their special functionality for ISO 17532 communication are useful only for special data exchange between high-level database systems such as SQL database systems in connection-oriented communication such as socket sessions. Other database systems and devices that do not even store persistent data ignore this data, which is sent with special status markings.

For correct interpretation, a time stamp item is required in each entity sent with such a status mark. The time stamp item includes the date and time of last change (CCYYMMDDHHmmSSsss).

Special agreements between the partners are necessary to ensure the receiver can process all of the sender's commands. These agreements are not dealt with by this International Standard, which offers a general framework for those functions.

7.10.9.2 Deletion of data, enforced deletion

This International Standard defines a general concept for deletion of single data records of the specified entity in the receiver's database. It is appropriate only for communication between higher-level database systems.

To delete single records, all key fields must be set.

Value lines for the entity are the following. The value entries specify exactly which records are to be deleted.

For simple deletion, it is then up to the receiver as to whether the data will actually be deleted or not. In case of enforced deletion, whenever possible the receiving party shall delete the data in its database.

NOTE This International Standard describes deletion of data previously sent by ADIS data files, which more closely resembles an "undo" of a data transfer.

7.10.9.3 Synchronization data

This International Standard offers synchronization for initialization of a database (filling for the first time) or comparison of the data of the sending and of the receiving party to determine which is the correct data between the communication partners. No details are defined. The following rules define ISO 17532 communication between high-level database systems, with synchronization started from partner A with data of partner B.

- Partner A sends entity definition and values of the records it wants to have synchronized, including the time stamp item of date and time of last change to this record.
- Partner B returns the same entity definition and all value lines where time stamps for the last change in B are later than the time stamps of partner A. If the record did not yet exist in partner B, it inserts the record data without further notice. If this is not possible (e.g. because some data essential in B is missing), the value lines are returned marked as faulty.

7.10.9.4 Insertion of data, enforced insertion

Data lines (definition and values) sent for insertion create new records for the given entity in the receiver. In case of insertion, the receiver loads data only if appropriate to its application logic. In the case of enforced insertion, the recipient shall insert the records, if possible, in any way.

If the data cannot be inserted or is already included in the receiver database, the receiver returns the definition and value lines marked as faulty data.

7.10.9.5 Update of data, enforced update

Update definition and value data are used for updating existing records in the database of the receiver. With update, the receiver updates the data only if appropriate in the receiver's opinion. Enforced update means updating whenever possible.

If the data cannot be updated in the receiver database, the receiver returns the definition and values marked as faulty data.

7.10.9.6 Storing data, enforced storing

It is the responsibility of the receiver to interpret "storing data" either as insertion or as updating existing data. In the case of a simple store, it may refuse and report the failure to the sender. Enforced store means the receiver stores data whenever possible.

7.11 ADIS extensions

7.11.1 General

This International Standard adds to ADIS, as defined in ISO 11787, some line types and status characters and defines some extra behaviour. Some line types and status characters are simply ignored in some ISO 17532 communication levels, but the receiver should in turn ignore them and not return these lines as failures.

In socket-based communication there are no restrictions in line length.

As in ISO 11787, each line has a certain type given by its first character. The status character is the second character in a communication line. Line types and status characters are capital letters. Lower-case letters may be used for bilateral arrangements and shall be ignored by other parties.

Figure 8 shows the permitted (+) combinations between Line Type and Status Character. For the required combinations of various line types, see ISO 11787. After a PO line, one or more D/V lines as defined in ISO 11787 may follow. New combinations are those in the grey sections or similarly highlighted in Figure 8.

Line Type		Status Character												
		H	**N**	**F**	**A**	**P**	**R**	**O**	**E**	**D, K**	**S**	**I, J**	**U, V**	**W, X**
	Description	**Header**	**Normal**	**Fault**	**Login**	**Pending**	**Result**	**Open**	**Error**	**Delete, enforced Delete**	**Sync**	**Insert, enforced Insert**	**Update, enforced Update**	**Write, enforced Write**
D	**Definition**	+	+	+	+				+	+	+	+	+	+
V	**Value**	+	+	+	+				+	+	+	+	+	+
E	**End**		+											
C	**Comment**	+	+	+	+	+	+	+	+	+	+	+	+	+
S	**Search**		+	+										
R	**Request**		+	+		+	+							
T	**Terminate**		+											
F	**File**		+	+										
I	**Include**		+	+										
O	**Output**		+	+			+							
Z	**Phys. End**		+											
P	**Processing Instruction**		+	+			+	+						
Q	**Named Query**		+	+			+	+						

Figure 8 — Combinations of line types and status characters

7.11.2 Use of ADIS line types

7.11.2.1 Definition line D and value line V

In addition to the requirements given in ISO 11787, the following is applicable.

As in ISO 11787, a value line shall not occur without a previous definition line, both linked by their equal entity (event) number. This is also required in ISO 17532 multicast-based sessions: a definition line shall immediately precede the V lines.

In ISO 17532 communication, socket-based communication sessions, “compact transfer of definition lines” is permitted, where the value line need not follow immediately its definition line if the following additional arrangements are fulfilled.

The client sends a “DEFCOMPACT” Processing Instruction at the begin of a session. The item parameter_value (item number 901043) is sent with value “YES”. In the result line, the server answers with entity 990114. Parameter class, parameter group and parameter identifier in the answer of the server are set to values “DEFCOMPACT” and “YES” if the server supports compact transfer of definition lines. If not, the server answers “NO” as parameter value and the definition and value lines of one entity shall subsequently be sent as usual in ISO 11787.

If the client has applied for the compact transfer of definition lines and the server has confirmed, the arrangement is valid for the complete session.

If so, between definition and value lines, other line types or the lines of other entities (with different entity numbers) are allowed, except for T, termination, or Z, physical, end lines or header lines with status character H.

The ISO 17532 communication protocol defines some entities with fixed entity numbers, where a definition line and subsequent value lines may induce an answer of definition and value lines of another certain entity with fixed entity number.

EXAMPLE Sending a service inquiry entity with DN and VN lines to a device makes the device answer with DN and VN lines of the service inquiry reply entity.

The definition of these few entities and their entity numbers shall be listed in the central repository of code sets and constraints. This behaviour is similar to the NLF-specific new construct of the named queries, but named queries would mean additional parsing effort difficult in smaller devices which are the main users of these few special entities. Therefore, the original behaviour has been extended in these certain cases.

7.11.2.2 End line E

ISO 17532 devices should ignore E lines and shall not send back an error line.

7.11.2.3 Comment line C

This line shall be in accordance with ISO 11787.

7.11.2.4 Request line R and search line S

These lines shall be in accordance with ISO 11787. R defines the requested data, and S lines list search conditions on data ranges. If there is more than one S line, they are combined as “and” conditions. Subsequent search lines shall have the same entity number and shall be followed by a request line with the same entity number.

7.11.2.5 Termination line T

ISO 11787 defines T as the end of a main unit of information in an ADIS file, used to distinguish blocks within one file. It may never be followed by a DH + VH line pair.

In various ISO 17532 communication sequences, it is mandatory to define the end of a transaction sequence. For this purpose, a T line shall be used (terminate transaction). See 7.4.3.

7.11.2.6 File line F and include line I

In ISO 11787 the receiver uses the file of the given file name if it is available in the given path. Line type I means inclusion plus execution of the contents of the file. Line type F gives a file without demanding the execution — the receiver is free to decide what to do with the F line. The file given in an I line shall include valid ADIS lines. No further definitions about contents and the responsibility for the file contents are given. Therefore, these line types may be used only after additional agreements for the communication partners which are not within the Scope of ISO 11787. For ISO 17532 communication partners, if no agreements have been reached or the data is not accessible in the receiver's context, the receiver should ignore these lines and shall not send back an error line.

7.11.2.7 Output line O

This line is ignored in ISO 17532 communication sequences.

7.11.2.8 Physical end line Z

According to ISO 17532, it is essential that the end of a socket connection be announced using a ZN line. Either the server or the client shall send it before closing down the connection (see 7.4.2). ZN lines are not allowed in MCC.

7.11.2.9 Processing instruction P

In certain situations, a communication partner may wish to trigger the other partner to execute some actions not covered by other concepts realized in the ISO 17532 protocol.

EXAMPLE Complex data retrievals that cannot be requested by conventional request R and search S lines or using the new concept of named queries.

Alternatively, the receiver may execute certain program steps previously agreed upon. Such actions can be defined within a processing instruction, identified by its entity number or unique identifier as defined in the DD, together with a specification of the required action. The client sends a certain communication sequence of lines of type P to hand over the processing instruction and possible arguments to the receiver. The receiver may act immediately or later. If it has to return data, it may hand over the data immediately or later in an asynchronous way. See Annex A.

7.11.2.10 Named query Q

Named queries are requests with conditions that are more complex than simple S and R lines and with data items as arguments — similar to the functionality of named queries in SQL databases. The receiver answers data in D definition and V value lines. The named queries are identified by their entity number. They are to be defined in the central repository. See Annex A.

7.11.2.11 Additional arrangements and constraints for line types

Communication lines with line types D, V, R, S, P, Q in ISO 17532 communication should only be made using entities defined in the DD. Therefore, event number zero should not occur because this always requires additional arrangements not included in the general DD. Receivers may ignore lines with entity number zero.

7.11.3 Use of ADIS status characters

7.11.3.1 Header data H

See ISO 11787. Each data communication shall start with a header definition and value line, except for possible preceding comment lines or authentication lines (login). Essential data items in a header are described in the DD. In ISO 17532 communication, a termination line for transaction end (TN) shall be sent after the lines with header information.

7.11.3.2 Normal data N

See ISO 11787, normal data lines sent to the receiver. Additionally, in some special cases, where a receiver interprets definition and value lines of certain well-defined entities such as a request for a certain data entity, the value lines with status character N may contain data which is interpreted like arguments.

7.11.3.3 Faulty data F

Any line which the receiver cannot interpret completely is directed into an extra data channel (file for ISO 11787 or answer in an ISO 17532 data communication sequence) with status character changed to F and possibly more fields than the original line to describe the errors. See Annex A.

7.11.3.4 Login A

This International Standard defines data lines with status character A (for “authentication”) for the login procedure. See Annex A.

7.11.3.5 Pending P

The status character P for pending has been introduced by this International Standard in order to announce asynchronous communication. Pending indicates that the server will continue the communication in an asynchronous manner. Results may be returned later. See Annex A.

7.11.3.6 Result R

The status character R (for “result”) is used to announce result data returned later in asynchronous transactions. The entity in a result line specifies certain data necessary to identify the original source to which the data is returned. See Annex A.

7.11.3.7 Open O

Complex instructions in ISO 17532 communication, such as named queries and processing instructions, use this status character to mark a line which opens the transaction and defines some specific data. See Annex A.

7.11.3.8 Error E

This is the definition and value lines with additional error and error recovering information following the return of faulty data of line type F. See Annex A.

7.11.3.9 Deletion of data D and enforced deletion K

This is used only for high-level database systems with additional agreements between the communication partners.

A DD line defines the entity for which records are to be deleted. The following VD lines list at least the key field values of a single record to be deleted. It is then up to the receiver as to whether the data will actually be deleted or not. Status character K (for “kill”) means enforced deletion: whenever possible the receiving party shall delete the data in its database. If the data is not found or cannot be deleted, the value line shall be returned after changing the status character to F for faulty data.

7.11.3.10 Synchronization data S

ISO 11787 offers synchronization for initialization of a database (filling for the first time) or comparison of the data of the sending and of the receiving party to determine which are the correct data communicated between partners.

This International Standard defines the following rules for communication between high-level database systems, with synchronization started from partner A with data of partner B.

- Partner A sends entity definition line DS… and value lines VS… of the records it wants to have synchronized with status character S. All value lines include the time stamp when their records were last changed.
- If the record does not yet exist in B, the data is inserted if possible. If this cannot be done, the data is returned as faulty data (status character changed to F and the possible additional error information with status character E).
- If the records already exist in B, but have a later time stamp of the last change, the same DS entity definition line and all VS value lines of these records are sent back to A.

7.11.3.11 Insertion of data I and enforced insertion J

These are used only for high-level database systems with additional agreements between the communication partners.

The receiver creates new records for the data lines sent for the given entity. With I, the receiver loads data only if this seems appropriate. If not, the failure is noted to the sender. J signifies enforced insertion: the recipient shall insert the records, if possible, in any way.

7.11.3.12 Update of data U and enforced update V

These are used only for high-level database systems with additional agreements between the communication partners.

Data lines sent with U or V are for update of existing records in the database of the receiver. With U, the receiver updates the data only if this seems appropriate. Otherwise, the receiver may refuse data and note failure to the sender (returning lines with status F and possible additional error information with status E). Enforced update means updating whenever possible. If not, the return line is “faulty” (status character F), possibly with additional error information (status character E).

7.11.3.13 Storing Data W, enforced storing X

These are used only for high-level database systems with additional agreements between the communication partners.

It is the responsibility of the receiver to interpret “storing data” either as insertion or as updating existing data. In the case of W, it may refuse and note the failure to the sender (F and E lines). Enforced store means that the receiver tries to store the data whenever possible.

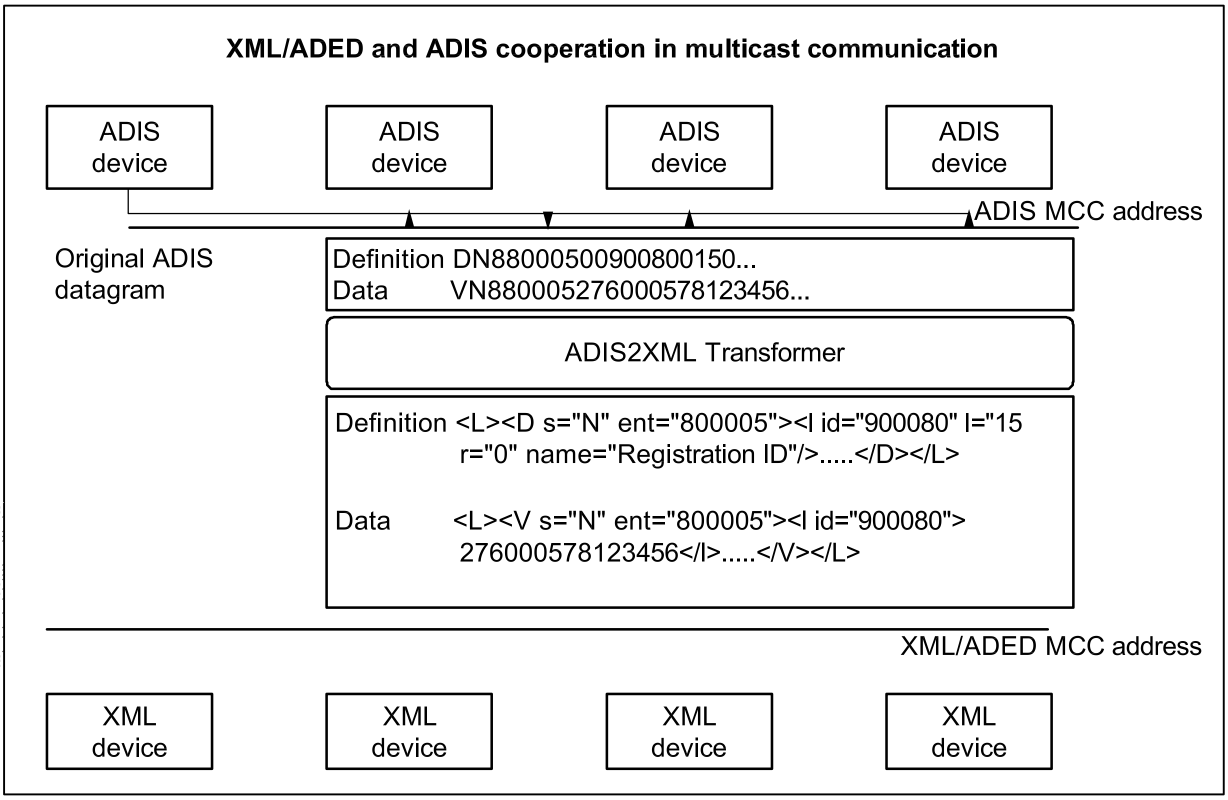
7.12 XML/ADED

7.12.1 General

XML/ADED is defined such that the extensions worked out can be implemented as simply as possible. XML/ADED is defined as a simple transformation from ADIS to XML. It implements all the positive features of ADIS, so that it will be the EDI interface with XML which creates a clear separation of the data content from the EDI function.

7.12.2 ADIS/ADED and XML/ADED

XML/ADED and ADIS/ADED may be used in parallel in ISO 17532 communication, because the protocols can be transformed with transformer software. For this, the data source is parsed with a simple parser and translated in a linear way into the other protocol. The DD is not required for parsing. Conversion between the MCC ports is achieved using a transformer, as shown in Figure 9.



```
Definition DN88000500900800150...
Data       VN8800052760005781234 56...
```

```
Definition <L><D s="N" ent="800005"><I id="900080" l="15
           r="0" name="Registration ID"/>.....</D></L>

Data       <L><V s="N" ent="800005"><I id="900080">
           276000578123456</I>.....</V></L>
```

Figure 9 — MCC with ADIS and XML/ADED in ISO 17532 communication

A separate multicast address is used for each protocol. Transformers are responsible for translation into the other protocol in each case, and transport the data — filtered, if necessary — to the relevant port. This results in complete transparency.

If socket communication is used, the transformation between ADIS/ADED and XML/ADED is performed using a translating service. The client sends the data flow to be transformed to this service and receives the translated data back line for line, then sends the data to the server. Companies which use ADIS must supply the configuration and implementation of the transformers and the translator, as soon as XML/ADED is used in the farm network.

7.12.3 Definition of XML/ADED

The XML/ADED string is placed behind the 8 B MCC header. The condition for using XML/ADED and ADIS/ADED side by side in ISO 17532 communication is simple conversion into each other. This will be accomplished using the ADIS/ADED line structure within XML/ADED. With this structure, data is organized based on ADIS/ADED items and entities.

XML/ADED is based on the XML protocol as defined by the W3C. It is compatible with XML 1.1.

XML/ADED uses the following elements of ADIS:

- line type;
- status character;
- event-numbers (ent);
- item numbers.

XML/ADED uses the XML schema for the description of EDI documents.

The schemata will be created automatically out of the ADED and, for an ADED available through the Internet, with the aid of the data model drawn on for standardization.

7.12.4 XML/ADED lines

An ADIS line is represented in XML/ADED by the <L> tag. Lines shall be specified using the attributes in accordance with Table 2, which shows the correlation between an ADIS/ADED line and an XML/ADED tag.

Table 2 — Attributes of XML element line <L>

Attribute	Description	Example	Type	Comment
t	Line type	t="D"	Mandatory	All line types shown in Figure 8 are allowed as attribute values.
s	Status character	s="N", s="H"	Mandatory	All status characters shown in Figure 8 are allowed as values. The relationships, as described in Figure 8, also apply for XML/ADED.
ent	Entity	ent="880005"	Mandatory in D, V, S, R, Q and P lines	—

func	Function name	func="address"
		func="alarm"
			Mandatory in Q and P lines	See 7.10.6 and 7.10.7.

maxrecords	Maximum number of records	maxrecords=100	Optional in S line	—

7.12.5 Items

An item element is represented by the <I> tag. Item elements shall contain attributes in accordance with Table 3.

Table 3 — Item attributes

Attribute	Description	Example	Type
id	Item number	id="900070"	Mandatory
l	Length	l="15"	Mandatory in D and R lines
r	Resolution	r="0"	Mandatory in D and R lines

Numbers with fixed decimal points are used.

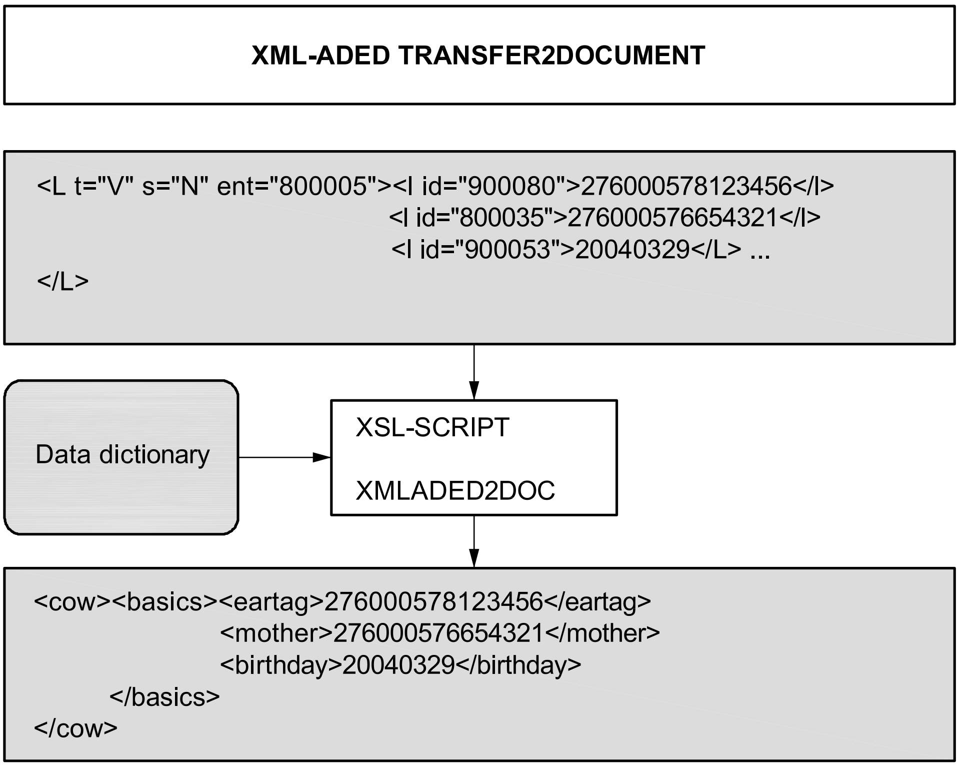
Search lines are required to describe the lowest and the highest values of a query field and are defined in special XML elements with separate tags. These elements are described in Table 4.

Table 4 — XML tags

Tag	Description	Example
<lowvalue>	Lowest value of a data field queried by a search line	<I id="900080" l="15" r="0" > <lowvalue>276000578331133</lowvalue> <highvalue>276000579000000</highvalue> </I>
<highvalue>	Highest value of a data field queried by a search line	
NOTE See Annex A.		

7.12.6 Transforming XML/ADED into humanly readable XML documents

Figure 10 demonstrates the ease of transforming XML/ADED-Flat into humanly readable XML.



```
<L t="V" s="N" ent="800005"><I id="900080">276000578123456</I>
                                 <I id="800035">276000576654321</I>
                                 <I id="900053">20040329</L> ...
</L>
```

```
<cow><basics><eartag>276000578123456</eartag>
                  <mother>276000576654321</mother>
                  <birthday>20040329</birthday>
          </basics>
</cow>
```

Figure 10 — Transforming XML/ADED into humanly readable XML documents

8 Data dictionary (DD)

8.1 General

ADIS/ADED provides the elementary EDI functions for entering, altering, synchronizing, error reporting and searching in remote databases in a simple and flexible way. The description of the metadata in a data dictionary enables strict separation of functions and data description.

The data elements and extensions used in ISO 17532 communication are described in 8.2 to 8.4.

8.2 DD elements essential for network livestock farming

8.2.1 Approach to using DD

A flexible approach to the use of definition and value lines is necessary with systems that communicate over time with various resources (hardwired process computers, computers with databases). The following remarks are made in this context.

The system will not work if a computer, finding in the header line that it has been presented with, e.g. an outdated DD, breaks off the communication. Better that it react flexibly and overlook unknown items and entities.

If after a search/request a definition line is sent back which does not comply with the requirements, the data recipient is requested to interpret the returned line and use it as far as possible.

8.2.2 Machine description, location description, service description

Various entities are under development to describe the interactions between device, service and location. It is most useful to describe devices using a device type. Devices shall provide two items of information for the communication:

- the services they can offer;
- the locations for which they can offer the services.

It is for this reason that the entities location description, service description and device description are just as essential for ISO 17532 communication as are the entities Device-Service-Interaction and Device-Location-Service interaction. The entities Device Description and Device-Location-Service shall be defined on configuration of the devices to ensure functioning of the inquire, publish, subscribe process. Entities such as Location Description, Location Occupation and animal-specific information, shall be available on request as a service from the relevant components in the network.

8.3 CODE SET and lists

Lists shall be predefined and described for various tasks, and shall be uniformly made available in an international DD. Table 5 shows the possible lists.

Table 5 — Lists

Named query list	List queries which can be predefined for complex inquiries
Processing instruction lists	List the commands necessary in the system (mix, feed, alarm, synchronize, air, reorganize data, issue lists, etc.) and define them clearly
List with functions and services	—
Error code lists	Codes for the severity of errors (note, warning, error, fatal) Error type

8.4 Entities for describing communicated data contents

It has been established that only a minimum amount of the data contents necessary for ISO 17532 is described internationally. To successfully operate ISO 17532, a procedure for harmonizing the data dictionaries must be implemented. The entities to be defined should be differentiated according to protocol-specific entities, entities describing farms, logistics entities and animal-related entities. ISO 17532 will focus on the protocol-specific DD elements.

9 Electrical specifications

Electrical requirements shall be in accordance with IEC 60204-1.

10 Mechanical specifications

The devices, connectors and cables shall be suitable for use in barns and milking parlours.

Annex A
(normative)

Data elements

A.1 Items and entities

A.1.1 Special items and CODE SET

See Tables A.1 to A.70.

Table A.1 — Device ID

Item	901001
Name	device_id
Data type	AN
Unity	Device type
Length	17
Resolution	0
Information	Will be compiled from the MAC address and an additional four-digit number for possible hidden devices. This means that several devices can be linked to the ISO 17532 network through a common network connection. If there is no hidden device “0000” is used. The MAC address and consecutive number are separated by a minus sign. If the MAC address of a device cannot be read off via software, it shall be manually entered. Caution: The MAC address is only used to allocate a unique number for the device. If the network card is replaced, the old MAC address remains as a basic element of the device. EXAMPLE 2f407a0022dd-0001

Table A.2 — Location ID

Item	901002
Name	location_id
Data type	AN
Unity	Location type
Length	40
Resolution	0
Information	Describes the location for which data is to be provided. A location is described by a sequence of sub-addresses separated by dots (e.g. “276123456789012:1.17.28”). The sequence shall begin with the farm number (15 N), starting with the numeric ISO country code [ISO 3166 (3 N)] and immediately followed by a unique national farm ID (12 N). This farm number shall be followed, separated by a colon (:), by a sequence of sub-addresses separated by dots. There is no dictate to assign each location to a defined object such as building, stable or valve. It is possible to label each location by an alias (e.g. Building, Silo, Computer, Feeding, V12) to create humanly readable locations. An asterisk (*) in a location code may be used to replace a sub-location. Only one asterisk in a location code is allowed. If sub-locations normally placed behind the asterisk are omitted, then all subsequent sub-locations shall be addressed. EXAMPLE 1 2761234:1.*.5 addresses all second-level sub-locations on a farm, with a third-level sub-address equal to 5. EXAMPLE 2 2761234:Stable1.* addresses all sub-locations in “Stable1”.

Table A.3 — TCP/IP address

Item	901003
Name	tcpip_address
Data type	AN
Unity	TCP/IP address
Length	39
Resolution	0
Information	TCP/IP uses IP addresses. IPv4 and IPv6 are supported. EXAMPLE 1 IPv4: 192.168.10.1. EXAMPLE 2 IPv6: 1111:00d3:0000:2f3b:0211:00bb:fe28:9c5a. The dot (.) and colon (:) are used to make it easier to remind the IP.

Table A.4 — MAC address

Item	901004
Name	mac_address
Data type	AN
Unity	MAC
Length	17
Resolution	0
Information	In a local area network (LAN) or other network, the MAC (media access control) address is the computer’s unique hardware number. (On an Ethernet LAN, it is the same as the Ethernet address.) When a device is connected to the Internet (or host, as the Internet protocol thinks of it), a correspondence table relates the IP address to the computer's physical (MAC) address on the LAN. The MAC address is used by the media access control sub-layer. In ISO 17532 communication, the MAC address is always given in hexadecimal form as EUI-48. With IPv6, EUI-64 is used. EXAMPLE For use under IPv4: “2f407a0022dd”.

Table A.5 — URI (uniform resource identifier RFC 1630)

Item	901005
Name	uri
Data type	AN
Unity	URI
Length	99
Resolution	0
Information	This is usually the TCP/IP-address (see item 901003). It is also possible to use a named address like “feedstation.one.farm.net”, if there is access to a DNS in the network. A universal resource identifier (URL) is a member of this universal set of names in registered name spaces and addresses referring to registered protocols or name spaces. A uniform resource locator (URL), defined elsewhere, is a form of URI which expresses an address which maps onto an access algorithm using network protocols. Existing URI schemes which correspond to the (still mutating) concept of IETF URLs shall be updated in accordance with the terms of IETF.

Table A.6 — Service type

Item	901006
Name	service_type
Data type	N
Unity	CODE SET
Length	3
Resolution	0
Information	In ISO 17532 communication, various services of different types are offered by various devices. The service types used are described in the service type code set. The service types shown below have so far emerged as useful. This item is bit-coded: The service type can be a combination of different types, so it is possible to say: Supports Message and/or Search/Request.

Table A.7 — CODE SET (901006)

Key	Description
1	Message: e.g. events or measured values that are reported when they occur in a fixed sequence. If the data supply service type is described, this refers to the supply of data with a concretely described entity.
2	Search/Request: data is accessed via Search/Request.
4	Named Query: queries based on the question, “Who can answer what queries?” This refers to the option of carrying out an inquiry with a known query deposited in the DD. As three query types are defined, all three shall be dealt with in the code set.
8	Processing Instruction: processing instructions based on the question: “Who can answer what queries?” Processing instructions are introduced to trigger mechanical processes, alarms, etc. If a device supports this service, it offers it with the processing instruction service type.
3	Message or Search/Request.
5	Message or named Query.
7	Message or Search/Request or named Query.

Table A.8 — Type of provision

Item	901007
Name	type_of_provision
Data type	N
Unity	CODE SET
Length	3
Resolution	0
Information	Informs which protocol must or can be used to answer a query. This item is bit coded.

Table A.9 — CODE SET (901007)

Key	Description
1	Multicast message
2	TCP socket connection: here, the service provider calls back to negotiate further details via a TCP connection.
4	E-mail
3	MM or TCP socket connection.
6	TCP socket connection or e-mail are supported.

Table A.10 — Service

Item	901008
Name	service
Data type	N
Unity	—
Length	8
Resolution	0
Information	Describes the service depending on the service type. For messages, the entity number is given. For named queries, the query ID is given. For processing instructions, the processing instruction code is given.

Table A.11 — Service inquiry number

Item	901009
Name	service_inquiry_no
Data type	N
Unity	—
Length	4
Resolution	0
Information	Consecutive number of the service inquiry generated by the requesting device. This number, together with the device ID, is used to recognize the service inquiry.

Table A.12 — Time period for delivery

Item	901011
Name	delivery_time
Data type	N
Unity	Seconds
Length	11
Resolution	0
Information	The time period entered for which the request is valid. If a device subscribes to an entity, this delivery time is a timeout, after which the server will be finished sending data. It shall not exceed 24 h (84 600 s). Useful if the subscribing device shuts down without informing the server.

Table A.13 — Immediate delivery

Item	901012
Name	delivery_immediately
Data type	AN
Unity	Boolean (T/F)
Length	1
Resolution	—
Information	If “T”, then the data source is forced to send the last valid message immediately; otherwise “F”.

Table A.14 — Time of genesis of combined event

Item	901013
Name	timestamp
Data type	N
Unity	Timestamp (CCYYMMDDHHmmSSsss)
Length	17
Resolution	0
Information	General time format. Used as local time. Also informs about the time of genesis of the combined event. This item shall be contained in all entities relating to measurements.

Table A.15 — Occurrence location

Item	901014
Name	occurrence_location
Data type	AN
Unity	Location type
Length	40
Resolution	0
Information	Informs about the location the event is allocated to and where it took place. It may be necessary to distinguish here between occurrence location and allocation location. This item shall, for example, be contained in all entities relating to measurements.

Table A.16 — Device that detected the event

Item	901015
Name	event_device_id
Data type	AN
Unity	Device type
Length	20
Resolution	0
Information	Informs regarding the device for which the event occurred and/or the measurement was conducted. This item shall be contained in all entities relating to measurements.

Table A.17 — Handle number

Item	901018
Name	handle_no
Data type	N
Unity	—
Length	6
Resolution	0
Information	Handle number, assigned to a device when opening a named Query or processing instruction as a reference number.

Table A.18 — Processing instruction ID

Item	901019
Name	id_of_pi
Data type	N
Unity	—
Length	8
Resolution	0
Information	Defines a processing instruction by ID, based on the DD. The definitions are registered in the DD according to the naming rules of a DD. Usually a six-digit ID is used with heading zeroes.

Table A.19 — Named query ID

Item	901020
Name	id_named_query
Data type	N
Unity	—
Length	8
Resolution	0
Information	Defines a named query by ID, based on the DD. The definitions are registered in the DD according to the naming rules of a DD (compare to item 901019). Usually a six-digit ID is used with heading zeroes.

Table A.20 — Name-based named query

Item	901021
Name	named_query
Data type	AN
Unity	—
Length	15
Resolution	0
Information	The name of the Query can be any given alphanumerical sequence of length 15. The definition of the named Query is registered in a national DD according to the naming rules of a DD.

Table A.21 — Responding device ID

Item	901022
Name	resp_device_id
Data type	AN
Unity	Device type
Length	17
Resolution	0
Information	This is the device ID of the responding device. For more details, see item 901001.

Table A.22 — Time the entity has been stored

Item	901024
Name	store_timestamp
Data type	N
Unity	Timestamp (CCYYMMDDHHmmSSsss)
Length	17
Resolution	0
Information	Informs about the time the entity has been stored in the actual database.

Table A.23 — Location type

Item	901025
Name	location_type
Data type	N
Unity	CODE SET
Length	43
Resolution	0
Information	Describes in a code set the different location types.

Table A.24 — CODE SET (901025)

Key	Description
1	trough
2	animal scales
3	vehicle scales
4	climate computer
5	PDA
6	PC

Table A.25 — Capacity type

Item	901026
Name	capacity_type
Data type	N
Unity	CODE SET
Length	3
Resolution	0
Information	Describes in a code set the different capacity types.

Table A.26 — CODE SET (901026)

Key	Description
1	count
2	volume
3	weight
4	length

Table A.27 — Capacity

Item	901027
Name	capacity
Data type	N
Unity	—
Length	9
Resolution	3
Information	Describes the value of the capacity depending on the capacity type and code set (901026).

Table A.28 — Description

Item	901028
Name	description
Data type	AN
Unity	—
Length	99
Resolution	0
Information	Special information for humans.

Table A.29 — Comment

Item	901029
Name	comment
Data type	AN
Unity	—
Length	99
Resolution	0
Information	Special information for humans.

Table A.30 — Service status

Item	901030
Name	service_status
Data type	N
Unity	CODE SET
Length	3
Resolution	0
Information	Informs about the status of a service in an asynchronous transaction such as a processing instruction or named query.

Table A.31 — CODE SET 901030

Key	Description
1	in process
2	delayed
3	stopped
4	error

Table A.32 — Manufacturer ID

Item	901031
Name	manufacturer_id
Data type	N
Unity	CODE SET
Length	15
Resolution	0
Information	ISO 17532 manufacturer ID structure: country code: national manufacturer number. Country code: ISO 3166-1 numeric, 3 digits. National manufacturer number: 12 digits.

Table A.33 — CODE SET (901031)

No.	Company
1	276000000000001 German company 1 [a]
2	040000000000001 Austrian company 1 [a]

[a] These are by way of examples.

Table A.34 — Hardware version

Item	901032
Name	hardware_version
Data type	AN
Unity	—
Length	20
Resolution	0
Information	Hardware version of device. Each manufacturer is free to define a hardware version for diagnostic purposes.

Table A.35 — ISO 17532 level

Item	901033
Name	nlf_level
Data type	N
Unity	CODE SET
Length	1
Resolution	0
Information	Supported ISO 17532 level of a device. Useful for diagnostic and network management. See 7.9.

Table A.36 — Service subscribe/unsubscribe

Item	901034
Name	subscribe_service
Data type	AN
Unity	Boolean (T/F)
Length	1
Resolution	0
Information	If subscribe to data, then “T”; if unsubscribe, then “F”.

Table A.37 — Initiation of delivery

Item	901037
Name	initiation_of_delivery
Data type	N
Unity	CODE SET
Length	1
Resolution	0
Information	How to deliver data. Described in a code set.

Table A.38 — CODE SET (901037)

Key	Description
1	—
2	Send, if changed.
3	Immediate delivery, pending is not allowed.
4	Send once.
5	Send periodically. In this case, item 901038 defines the period.

Table A.39 — Delivery frequency

Item	901038
Name	delivery_frequency
Data type	N
Unity	msec
Length	8
Resolution	0
Information	Sending interval of data. See also item 901037.

Table A.40 — Linking type

Item	901039
Name	linking_type
Data type	N
Unity	CODE SET
Length	2
Resolution	0
Information	Describes in a code set the different location-linking types. Here: “Which device is responsible for which location?”

Table A.41 — CODE SET (901039)

Key	Description
1	feeding
2	climate

Table A.42 — Parameter class

Item	901040
Name	param_class
Data type	N
Unity	—
Length	30
Resolution	—
Information	This item shall be used together with items 901041, 901042, 901043. For more details, see entity 990114 (Table A.84).

Table A.43 — Parameter group

Item	901041
Name	param_group
Data type	AN
Unity	—
Length	15
Resolution	0
Information	This item shall be used together with items 901040, 901042 and 901043. For more details, see entity 990114 (Table A.84).

Table A.44 — Parameter determiner

Item	901042
Name	param_determiner
Data type	AN
Unity	—
Length	15
Resolution	0
Information	This item shall be used together with items 901040, 901041 and 901043. For more details, see entity 990114 (see Table A.84).

Table A.45 — Parameter value

Item	901043
Name	param_value
Data type	AN
Unity	—
Length	99
Resolution	—
Information	Parameter values, numeric as well as alphanumeric, shall be sent by string format. This item shall be used together with items 901040, 901041, 901042. See Table A.84.

Table A.46 — Communication port

Item	901044
Name	ip_port
Data type	N
Unity	—
Length	5
Resolution	0
Information	IP port number used in MCC or socket connection.

Table A.47 — Alias

Item	901045
Name	alias
Data type	AN
Unity	—
Length	99
Resolution	0
Information	Label for a location, device, etc.

Table A.48 — Sender device ID

Item	901046
Name	sender_device_id
Data type	AN
Unity	device_typ
Length	17
Resolution	0
Information	The device ID of a sending device. Needed for the ADIS/XML gateway. For device ID details, see item 901001.

Table A.49 — Receiver device ID

Item	901047
Name	receiver_device_id
Data type	AN
Unity	device_typ
Length	17
Resolution	0
Information	The device ID of a receiving device. Needed for the ADIS/XML gateway. For device ID details, see item 901001.

Table A.50 — Receiver address

Item	901048
Name	receiver_address
Data type	AN
Unity	TCP/IP address
Length	39
Resolution	0
Information	The IP address of a receiving device. Needed for the ADIS/XML gateway. For address details, see item 901003.

Table A.51 — Named-based processing instruction

Item	901049
Name	named_pi
Data type	AN
Unity	—
Length	15
Resolution	—
Information	The name of the processing instruction can be any given alphanumerical sequence of length 15. The definition of the named Query is registered in a national or international DD according to the naming rules of a DD.

Table A.52 — Device type

Item	901050
Name	device_type
Data type	N
Unity	CODE SET
Length	8 data type AN
Resolution	0
Information	Item used to publish the device type as defined in a code set. This item is bit coded. The device type is defined in a code set. The code values are presented in hexadecimal digits.

Table A.53 — CODE SET (901050)

Key	Description
1	Climate computer
2	Mixing computer
4	Feeding computer
6	Mixing and feeding computer
8	Milking robot
…	…

Table A.54 — User name

Item	901051
Name	user_name
Data type	AN
Unity	—
Length	40
Resolution	0
Information	Used for the device login.

Table A.55 — Password

Item	901052
Name	password
Data type	AN
Unity	—
Length	8
Resolution	0
Information	Used for the device login.

Table A.56 — Encryption method

Item	901053
Name	encryption_method
Data type	N
Unity	CODE SET
Length	2
Resolution	0
Information	Used in the header for encryption of the session data.

Table A.57 — CODE SET (901053)

Key	Description
0	no encryption
1	DES
2	RSA
3	RC4
4	IDEA
5	Blowfish [a]

[a] Blowfish is supported. This code set is likely to be refined with codes for other methods.

Table A.58 — Sender address

Item	901054
Name	sender_address
Data type	AN
Unity	URI
Length	99
Resolution	0
Information	Address of the client in a session. Used in the header line.

Table A.59 — Receiver address

Item	901055
Name	receiver_address
Data type	AN
Unity	URI
Length	99
Resolution	0
Information	Address of the server in a session. Used in the header line.

Table A.60 — Currency

Item	901056
Name	currency
Data type	AN
Unity	—
Length	3
Resolution	0
Information	ASCII description of the currency used, defined in ISO 4217. Used in the header line.

Table A.61 — Character set

Item	901058
Name	character_set
Data type	N
Unity	CODE SET
Length	2
Resolution	0
Information	Reference for character CODE SET. Used in the header line. CODE SET from ISO 8859. Used in the header line.

Table A.62 — CODE SET (901058)

Key	Description
0	US ASCII
1	ISO-8859-1
2	ISO-8859-2
—	ISO-8859
99	Uni-Code Base64 coding

Table A.63 — ISO DD name

Item	901059
Name	dd_name
Data type	AN
Unity	—
Length	8
Resolution	0
Information	Name of the international data dictionary. Used in the header.

Table A.64 — National DD name

Item	901060
Name	dd_national_name
Data type	AN
Unity	—
Length	8
Resolution	0
Information	Name of the national data dictionary. Used in the header. Combination of country code and name.

Table A.65 — UTC timestamp

Item	901061
Name	utc_timestamp
Data type	N
Unity	Timestamp (CCYYMMDDHHmmSSsss)
Length	17
Resolution	0
Information	General UTC time format. Used to synchronize all devices in ISO 17532 communication.

Table A.66 — Error code

Item	901062
Name	error_code
Data type	N
Unity	CODE SET
Length	4
Resolution	0
Information	List of general errors which occur in ISO 17532 communication.

Table A.67 — CODE SET (901062)

Key	Description
0	No error

Table A.68 — Alarm code

Item	901063
Name	alarm code
Data type	N
Unity	CODE SET
Length	2
Resolution	—
Information	The code set is described in the international data dictionary.

Table A.69 — Alarm description

Item	901064
Name	alarm description
Data type	AN
Unity	—
Length	99
Resolution	—
Information	Alarm descriptions provide text-based information on the cause of the alarm.

Table A.70 — Alarm response

Item	901065
Name	alarm response
Data type	AN
Unity	Boolean (T/F)
Length	1
Resolution	—
Information	If alarm handled, then “T”; if not handled, then “F”.

A.1.2 Special entities

Various special entities shall be defined for the use of ADIS in network livestock farming (ISO 17532). See Tables A.71 to A.87.

Table A.71 — Header (990054)

Item	Name	Data type
000000 [a]	data dictionary type	M
900002 [a]	ADED DD version	M
901059	ADED DD name	M
900003 [a]	file date	M
900004 [a]	file time	M
900005 [a]	system status	O
900006 [a]	sender name	M
900007 [a]	receiver name	O
900008 [a]	sender software version	O
900009 [a]	ADED DD national version	O
901060	ADED DD national name	O
900011 [a]	process device type	O
900012 [a]	ADED manufacturer version	O
901054	sender address	O
901055	receiver address	O
901056	currency	O
901058	character set	O
[a] Item defined in ISO 11788-1.		

Table A.72 — Device description (990101)

Item	Name	Data type
901001	device_id	K
901031	manufacturer_id	M
900008	software_version	M
901050	device_type	M
901032	hardware_version	M
900002	supported_dd	M
901033	nlf_level	M
901005	URI	C
901028	description	C
NOTE Describes the device used and its function. Device items 900002 and 900008 are defined in ISO 11788-1.		

Table A.73 — Location description (990102)

Item	Name	Data type
901002	location_id	K
901025	location_type	M
901045	alias	O
901029	comment	O
NOTE Describes the location.		

Table A.74 — Location capacities (990103)

Item	Name	Data type
901002	location_id	K
901026	capacity_type	K
901027	capacity	M
901029	comment	O
NOTE Describes the location capacities. EXAMPLE The maximal animal amount for this location.		

Table A.75 — Device-location linking (990104)

Item	Name	Data type
901001	device_id	K
901002	location_id	K
901039	linking_type	M
901028	description	C
901029	comment	O
NOTE Describes the linking between device and location. Useful for configuring the ISO 17532 communication. EXAMPLE Device “12345-001” is responsible for location “0:12.6”.		

Table A.76 — Service (990105)

Item	Name	Data type
901008	service	K
901006	service_type	M
901029	comment	O
NOTE Describes a service.		

Table A.77 — Machine-location-service linking (990106)

Item	Name	Data type
901001	device_id	K
901002	location_id	K
901008	service	K
901028	description	O
901029	comment	O
NOTE Describes the service a device produces for a location.		

Table A.78 — Service publish (990107)

Item	Name	Data type
901001	device_id	K
901006	service_type	K
901008	service	K
901002	location_id	K
901007	type_of_provision	M
901028	description	O
901029	comment	O
NOTE Logging into the network is performed by announcing the participant's services/locations via multicast in the network. This ensures that all available network participants are informed about the new participant. This describes services published by a device on request or while connecting to the ISO 17532 network.		

Table A.79 — Service inquiry (990108)

Item	Name	Data type
901001	device_id	K
901009	service_inquiry_no	K
901006	service_type	M
901008	service	M
901007	type_of_provision	M
901002	location_id	M
NOTE Devices inquire about the service on entry into the network and then periodically via the central multicast address. The inquiry is carried out via the service inquiry entity. The answer is sent back to the multicast address via the service request reply entity. The device ID is that of the inquiry device. The item 901008 may use the wildcard to inquire of all services.		

Table A.80 — Service inquiry reply (990109)

Item	Name	Data type
901001	device_id	K
901022	resp_device_id	K
901009	service_inquiry_no	K
901007	type_of_provision	K
901005	URI	M
901044	ip_port	M

NOTE A device confirms that the inquired information is available. With this entity, the devices that received inquiries answer the inquiring device via the central multicast address if they want to offer their services. After receiving this entity, the computer will contact the answering systems and send a Service-Request entity subscribing the service. The “device_id” is the ID of the inquiry device. The “resp_device_id” is the ID of the responding device.

Table A.81 — Service request (990110)

Item	Name	Data type
901001	device_id	K
901018	handle-number	K
901008	service	M
901002	location_id	M
901034	subscribe_service	M
901022	resp_device_id	K
901005	URI	M
901044	ip_port	M
901006	service_type	M
901007	type_of_provision	M
901011	delivery_time	O

NOTE This entity is used to request an entity on a device in ISO 17532.

EXAMPLE Device 12345678-0002 wants to subscribe to the service 00610506 (Sell animals) as a message by socket connection. The answer has to be sent to IP 192.168.199.200 port 3001.

Table A.82 — Service request reply (990111)

Item	Name	Data type
901001	device_id	K
901018	handle-number	K
901008	service	M

NOTE This is the acknowledgement of the service request.

Table A.83 — Status (990113)

Item	Name	Data type
901001	device_id	K
901018	handle_no	K
901030	service_status	M
901029	comment	M
NOTE This entity is read by a Search/Request command to get the status of a processing instruction or a named query.		

Table A.84 — Generic machine configuration (990114)

Item	Name	Data type
901001	device_id	K
901040	param_class	M
901041	param_group	M
901042	param_determiner	M
901043	param_value	M
NOTE This entity is used to read/write special variables in the devices. Generic parameters for device configuration, location and service can be used with the "*"- pattern. EXAMPLE Set the variable "timeout" in parameter group "socket1" of class "communication" to 1 000 ms.		

Table A.85 — Login/authentication (990115)

Item	Name	Data type
901053	encryption_method	O
901051	user_name	K
901052	password	O
NOTE The encryption method shall be sent first in order to make it possible to encrypt the user name and password in the same value line.		

Table A.86 — Time synchronization (990116)

Item	Name	Data type
901001	device_id	K
901013	timestamp	K
901061	utc_timestamp	O
NOTE It is useful to have a time server in the ISO 17532 network. One device may be declared by configuration to a time server. This server sends its time at 1 min intervals: "timestamp" is local time; "utc_timestamp" is UTC.		

Table A.87 — Simple alarm (990117)

Item	Name	Data type
901001	device_id	K
901013	timestamp	K
901063	alarm code	K
901064	alarm description	O
901065	alarm response	O
901022	resp_device_id	O
NOTE The alarm sends items 901001, 901013 and 901063, and, optionally, item 901064. The alarm processor additionally replies with items 901065 and 901022. The item alarm_response is set to "F" when the alarm could not be processed and to "T" when the alarm has been successfully processed or cancelled by the alarm processor.		

A.2 ADIS sequences

A.2.1 Rules of syntax

The following rules are used to describe the syntax in accordance with ISO 11787, in respect of the definitions of items:

"DDI" is the item number as defined in the data dictionary;

"<length>" is the two-digit number for the number of characters for the value;

"<res>" is the single-digit number for the resolution if the value is a number;

"<crlf>" is a carriage-return and linefeed character pair terminating each line; items in round brackets;

"(…)" items in parentheses can occur a number of times repeatedly, at least once;

"[…]" items in square brackets are conditional (not mandatory).

A.2.2 Transaction "Start up of communication"

See Table A.88 for the manner in which each socket session starts up.

Table A.88 — Start up of communication

Communication partner A (client)	→	Connect to known TCP port	→	Communication partner B (server)

A.2.3 End Session

See Table A.89.

A ZN line may be sent either from the server or from the client.

Table A.89 — End session

Communication partner A (client) wants to terminate the connection	→	ZN<crlf> A closes the connection to the port after sending ZN	→	Communication partner B (server) B knows that A will close down the connection and will not expect any more data to be accepted by A

A.2.4 Login procedure

See Table A.90.

Table A.90 — Sequence diagram if authentication required

Communication partner A (client)	→	DA<LoginEntityNo><Code for Encryption Procedure DDI><length><res><User Name DDI><length><res><Password DDI><length><res><crlf> VA<LoginEntityNo><…><…><…><crlf> TN<crlf> if no encryption used, code value for encryption procedure is 0	→	Communication partner B (server)
Communication partner A (client)	←	TN<crlf>	←	If Login OK
Communication partner A (client)	←	DF<LoginEntityNo><Code for Encryption Procedure DDI><length><res><User Name DDI><length><res><Password DDI><length><res><Error Code DDI><length><res><Error Severity Level DDI><length><res><crlf> VF<LoginEntityNo><…><…><…><…><…><crlf> TN<crlf> ZN<crlf>	←	If Login fails and totally rejected, return faulty data refined by error code item and error severity level item
Communication partner A (client)	←	DF<LoginEntityNo><Code for Encryption Procedure DDI><length><res><User Name DDI><length><res><Password DDI><length><res><crlf> VF<Login EntityNo><…><…><…><crlf> CF <error message><crlf> TN<crlf> ZN<crlf>	←	If Login fails and totally rejected, return faulty data and report error in a subsequent comment line

Table A.90 (*continued*)

Communication partner A (client)	←	DF<LoginEntityNo><Code for Encryption Procedure DDI><length><res><User Name DDI><length><res><Password DDI><length><res><Error Code DDI><Error Severity Level DDI><length><res><crlf> VF<LoginEntityNo><…><…><…><…><…><crlf> TN<crlf> Or if no error message supported: DF<LoginEntityNo><Code for Encryption Procedure DDI><length><res><User Name DDI><length><res><Password DDI><length><res><crlf> VF<Login EntityNo><…><…><…><crlf> CF <error message><crlf> TN<crlf>	←	If Login fails, and guest login (restricted access level)

A.2.5 Header data

The beginning of each communication session shall start with header lines, see status character “H”.

For ISO 17532 communication, the header entity specifies the following data items given in Table A.91, some of which are additions to the international DD as defined in ISO 11788 for ADED.

See Table A.91.

Table A.91 — Sequence diagram for communication of header data

Communication partner A (client)	→	DH990001000000900002900003900004900006…<crlf> VH990001<DD Type><ADED ISO Version>… …<Creation Date><Creation time>… ...<Sender SW Name>...<crlf> TN<crlf>	→	Communication partner B (server)
Communication partner A (client)	←	Optionally the partner B/server answers its header data DH990001000000900002900003900004900006…<crlf> VH990001<DD Type><ADED ISO Version>… …<Creation Date><Creation time>… ...<Sender SW Name>...<crlf> The termination line which informs the client that its header data has been accepted is mandatory: TN<crlf>	←	Communication partner B (server)

A.2.6 Sending data (D and V Lines)

Classical sending of data with DN and VN lines: see Table A.92.

Table A.92 — Client A sends data to server B

Communication partner A (client)	→	DN<EntityNo1>(<DDI><length><res>)<crlf> VN<EntityNo1>(<...>)<crlf> VN<EntityNo1>(<...>)<crlf> Other entity definitions and value lines may follow optionally: DN<EntityNo2>(<DDI><length><res>)<crlf> VN<EntityNo2>(<...>)<crlf> VN<EntityNo2>(<...>)<crlf> Sending all definition lines at the beginning and their value lines later requires the DEFCOMPACT processing instruction (see details in section about definition and value lines). Instead of “DN + VN” lines with normal status “N” after special agreements in high-level database systems: DI + VI, DJ + VJ, DU + VU, DV + VV or DW + VW lines (for details, see description of these status characters)	→	Communication partner B (server)
Communication partner A (client)	→	TN<crlf>	→	Communication partner B (server)
Communication partner A (client)	←	DN + VN lines, see above, but a certain other entity (defined in Data Dictionary)	←	If the entity is a special one which requires another entity to be sent back
Communication partner A (client)	←	TN<crlf>	←	Always!

A.2.7 Failure management and error handling

Faulty DN/VN data shall be returned marked with status “F”, possibly refined with an error code item and an item for the error severity level.

Additional error information data may be appended in definition and value lines immediately following the VF lines. To distinguish this information from normal D/V data, the status of error information lines shall be set to a new status character E (for “error information”)

If the receiver does not support any of these error message concepts, it shall return the faulty data as DF/VF lines and write some error information in a comment CF line.

See Tables A.93 and A.94.

Table A.93 — Failure management of value lines

Communication partner A (client)	→	Sender sends faulty data: DN<entityNoX >(<ddi><length><res>)<crlf> VN< entityNoX (<value>)<crlf> TN<crlf>	→	Communication partner B (server)
Communication partner A (client)	←	Returning faulty data refined by error information: DF<entityNoX >(<ddi><length><res>)<error code-ddi><length><res><severity level-ddi><length><res><crlf> VF<entityNoX >(<values>)<error code value><severity level value><crlf>] TN<crlf>	←	The receiver processes the data and reports faulty data with appended error codes and error severity level
Communication partner A (client)	←	Returning additional error information data (optional) after the return of the DF/VF line(s): DE<errorEntityNoY >(<ddi><length><res>)<crlf> VE<ErrorEntityNoY>(<value>)<crlf> DE<errorEntityNoY ><key ddi><length><res>[<ddi><length><res>]<crlf> VE<ErrorEntityNoY><key value>[<value>]<crlf> TN<crlf>	←	The receiver processes the data and reports faulty data with appended error codes and error severity level
Communication partner A (client)	←	Error information in a comment line if no other error messages are supported: DF<entityNoX >(<ddi><length><res>)<crlf> VF<entityNoX >(<values>)<crlf> CF < text with error description><crlf> TN<crlf>	←	The receiver processes the data and reports faulty data with appended error codes and error severity level
If client A recognizes a failure in data received from server B, the same procedure for reporting an error shall be executed in the other direction.				

Table A.94 — Failure management of non-value lines (D, R, S, P, Q)

Communication partner A (client)	→ or ←	<LineType><StatusChar>….	→ or ←	Communication partner B (server)
Communication partner A (client)	← or →	If additional error information entities are supported: <LineType>F DE<ErrorEntityNoY>(<ErrorItem DDI>)(<ErrorSeverityLevel DDI><crlf> VE<ErrorEntityNoY>(<ErrorItem Value>)(<ErrorSeverityLevel Value>)<crlf> TN<crlf>	← or →	Line is erroneous
Communication partner A (client)	→ or ←	If no additional error information entities are supported: <LineType>F(<…>)<crlf> <LineType>F….<crlf> CF<error message><crlf> TN<crlf>	→ or ←	Line is erroneous
If client A recognizes a failure in data received from server B, the same procedure for reporting an error shall be executed in the order direction.				

A.2.8 Classical request of data (S and R lines)

In addition to ADIS ISO 11787, server B can optionally perform its return of the result in asynchronous manner and set the transaction to pending if the client has opened the transaction with the “RO” open request command.

See Tables A.95 to A.99.

Table A.95 — Classical request of data

Communication partner A (client)	→	Mandatory in SBC: RO<EntityNo>(<DDI-no><length> <res><value>)<crlf> Mandatory items in the RO line are the device ID of the sender and a handle number identifying this request call uniquely within the sender	→	Communication partner B (server)
Communication partner A (client)	→	(SN<EntityNo><search item1 no><search item1 lower value>><search item1 upper value><crlf>) (SN<EntityNo><search item2 no><search item2 lower value>><search item2 upper value><crlf>) RN<EntityNo>(<ddi><length><res>)<crlf>	→	Communication partner B (server)
Communication partner A (client)	→	TN<crlf>	→	Communication partner B (server)

Table A.96 — Case a) — Immediate results

Communication partner A (client)	←	If transaction was opened by an RO command: RR<EntityNo>(<DDI-no><field-length><resolution><value>)<crlf> The RR line shall at least include the entity number of the request, the device ID of the sender and the handle number as given in the RO line	←	Communication partner B (server)
Communication partner A (client)	←	“DN” + “VN” lines for EntityNoX, like in “Client A sends data to server B”	←	Communication partner B (server)
Communication partner A (client)	←	TN<crlf>	←	Communication partner B (server)

Set to pending

Table A.97 — Case b) — Asynchronous results

Communication partner A (client)	←	RP<EntityNo>(<DDI-no><length><res><value>)<crlf> Mandatory items in the “RP” line are the identifier of the processing instruction, the device-id of the sender and the handle number identifying this processing instruction call uniquely within the sender	←	Communication partner B (server)

Table A.98 — Results if type of provision is socket (B client and A server)

Communication partner A (server)	←	Connect to known TCP port	←	Communication partner B (client)
Communication partner A (server)	←	Possible login for authentication procedure DA…<crlf> VA…<crlf> TN<crlf> For details, see “Login procedure”	←	Communication partner B (client)
Communication partner A (server)	→	If login required: TN<crlf>	←	Communication partner B (client)
Communication partner A (server)	←	RR …<crlf> DN + VN lines (see immediate results) Or possible failures: RF …..<crlf>	←	Communication partner B (client)
Communication partner A (server)	←	TN<crlf>	←	Communication partner B (client)
Communication partner A (server)	→	TN<crlf>	←	Communication partner B (client)

Table A.99 — Results if type of provision is e-mail

Communication partner A (client)	←	Server creates an e-mail to client A with the following contents in the mail itself (no file attachment)	←	Communication partner B (server)
Communication partner A (client)	←	RR …<crlf> DN + VN lines (see immediate results) Or possible failures: RF …..<crlf> TN<crlf>	←	Communication partner B (server)

A.2.9 Named Query (Q lines)

See Tables A.100 to A.104.

Table A.100 — Named query (Q lines)

Communication partner A (client)	→	QO<EntityNoX>(<DDI-no><length><res><value>)<crlf> Mandatory items in the QO line are the identifier of the named query, the device ID of the sender and a handle number identifying this named query call uniquely within the sender	→	Communication partner B (server)
Communication partner A (client)	→	QN<EntityNoX><QueryIDItemNo.><length><res><name>[(<argumentItemNo.><length><res><value>)]<crlf>	→	Communication partner B (server)
Communication partner A (client)	→	TN<crlf>	→	The receiver starts to process the named query.

Table A.101 — Case a) — Immediate results

Communication partner A (client)	←	QR <EntityNoX>(<DDI-no><length><res><value>)<crlf> The QR-line shall at least include the identifier of the named query, the device ID of the sender and the handle number as given in the QO line	←	Communication partner B (server)
From the QR line, A knows the query requesting these data lines.	←	“DN” + “VN” lines with their entity numbers as defined in the result set for the named query, see “Server B sends data to client A” Or “QF” lines in case of failure	←	Communication partner B (server)
Communication partner A (client)	←	TN<crlf>	←	Communication partner B (server)

Set to pending

Table A.102 — Case b) — Asynchronous transactions

Communication partner A (client)	←	QP<EntityNoX>(<DDI-no><length><res><value>)<crlf> Mandatory items in the QP line are the identifier of the named query, the device ID of the sender and the handle number identifying this named query call uniquely within the sender	←	Communication partner B (server)
Communication partner A (client)	←	TN<crlf>	←	Communication partner B (server)

Table A.103 — Asynchronous results if type of provision is socket

Communication partner A (server)	←	Connect to known TCP port	←	Communication partner B (client)
Communication partner A (server)	←	Possible login for authentication procedure DA…<crlf> VA…<crlf> TN<crlf> (in case of successful login) For details, see A.2.4 "Login procedure"	←	Communication partner B (client)
Communication partner A (server)	→	TN<crlf> (if login required)	←	Communication partner B (client)
Communication partner A (server)	←	QR …<crlf> (DN + VN lines) See immediate results Or possible failures: QF …..<crlf>	←	Communication partner B (client)
Communication partner A (server)	←	TN<crlf>	←	Communication partner B (client)
Communication partner A (server)	→	TN<crlf>	←	Communication partner B (client)

Table A.104 — Asynchronous results if type of provision is e-mail

Communication partner A (client)	←	Server creates an e-mail to client A with the following contents in the mail body itself (no file attachment)	←	Communication partner B (server)
Communication partner A (client)	←	QR…<crlf> ("DN" + "VN" lines) For details, see "immediate results" Or possible failures: QF…..<crlf>	←	Communication partner B (server)
Communication partner A (client)	←	TN<crlf>	←	Communication partner B (server)

A.2.10 Processing Instruction (P lines)

See Tables A.105 to A.109.

Table A.105 — Processing instruction (P lines)

Communication partner A (client)	→	PO<EntityNoX>(<DDI-no><length><res><value>)<crlf> Mandatory items in the “PO” line are the identifier of the processing instruction, the device ID of the sender and a handle number identifying this processing instruction call uniquely within the sender.	→	Communication partner B (server)
Communication partner A (client)	→	Optional: “DN” + “VN” lines with their entity numbers for data or parameters necessary to perform the processing instruction (see “Client A sends data to server B”). All subsequent “DN”/“VN” lines of the “PO” line sent from client A within this transaction are part of the input for the processing instruction in the receiving server B. Their entity numbers and contents are defined in the data dictionary.	→	Communication partner B (server)
Communication partner A (client)	→	PN<EntityNoX>(<DDI-no><length><res><value>)<crlf> Mandatory items in the “PN” line are the identifier of the processing instruction, the device ID of the sender and an address (URI/URL) of the sender.	→	Communication partner B (server)
Communication partner A (client)	→	TN<crlf>	→	The receiver starts to process the processing instructions.

Table A.106 — Case a) — Immediate results

Communication partner A (client)	←	PR<EntityNoX>(<DDI-no><length><res><value>)<crlf> Mandatory items in the “PR” line are the identifier of the processing instruction, the device ID of the sender and the handle number identifying this processing instruction call uniquely within the sender and a result status.	←	Communication partner B (server)
Communication partner A (client)	←	(“DN” + “VN” lines may follow, depending on the definition of the processing instruction.) For details on “DN” + “VN” lines, see “DN” + “VN” lines, see “Server B sends data to client A”)	←	Communication partner B (server)
	←	TN<crlf>	←	

Table A.107 — Case b) — Pending transaction — Asynchronous results

Communication partner A (client)	←	PP<EntityNoX>(<DDI-no><length><res><value>)<crlf> Mandatory items in the “PP” line are the identifier of the processing instruction, the device ID of the sender and the handle number identifying this processing instruction call uniquely within the sender.	←	Communication partner B (server)
Communication partner A (client)	←	TN<crlf>	←	Communication partner B (server)

Table A.108 — Asynchronous results if type of provision is socket

Communication partner A (server)	←	Connect to known TCP port	←	Communication partner B (client)
Communication partner A (server)	←	Possible login for authentication procedure DA…<crlf> VA…<crlf> TN<crlf> (in case of successful login) For details, see “Login procedure”	←	Communication partner B (client)
Communication partner A (server)	→	TN<crlf> (if login required)	←	Communication partner B (client)
Communication partner A (server)	←	PR …<crlf> (“DN” + “VN” lines) Or possible failures: PF ….<crlf>	←	Communication partner B (client)
Communication partner A (server)	←	TN<crlf>	←	Communication partner B (client)
Communication partner A (server)	→	TN<crlf>	←	Communication partner B (client)

Table A.109 — Asynchronous results if type of provision is e-mail

Communication partner A (client)	←	Server creates an e-mail to client A with the following contents in the mail body itself: (no file attachment)	←	Communication partner B (server)
Communication partner A (client)	←	PR …<crlf> (“DN” + “VN” lines) For details, see “immediate results” Or possible failures: PF …..<crlf> TN<crlf>	←	Communication partner B (server)

A.3 XML/ADED scheme

An example of XML/ADED scheme is shown in Figure A.1.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:nlf="NLF"
    targetNamespace="NLF">
<annotation>
  <documentation xml:lang="DE">
    Issued on : 16.12.2004
  </documentation>
</annotation>
    <element name="DTA" type="nlf:DTATyp"/>
  <complexType name="DTATyp">
    <sequence>
      <element name="L" type="nlf:LTyp" minOccurs='0' maxOccurs='unbounded'/>
    </sequence>
  </complexType>
  <complexType name="LTyp">
<sequence>
      <element name="I" type="nlf:ItemTyp" minOccurs='0' maxOccurs='unbounded'/>
    </sequence>
    <attribute name="t" type="nlf:LineType" use="required"/>
    <attribute name="s" type="nlf:StatusType" use="required"/>
    <attribute name="ent" type="positiveInteger"/>
    <attribute name="func" type="string"/>
    <attribute name="maxrecords" type="positiveInteger"/>
  </complexType>
<simpleType name="LineType">
    <restriction base="string">
      <enumeration value="D"/>
      <enumeration value="V"/>
      <enumeration value="E"/>
      <enumeration value="C"/>
      <enumeration value="S"/>
      <enumeration value="R"/>
      <enumeration value="T"/>
      <enumeration value="F"/>
      <enumeration value="I"/>
      <enumeration value="O"/>
      <enumeration value="Z"/>
      <enumeration value="P"/>
      <enumeration value="Q"/>
    </restriction>
  </simpleType>
```

Figure A.1 — Example of XML/ADED scheme

```
<simpleType name="StatusType">
        <restriction base="string">
            <enumeration value="H"/>
            <enumeration value="N"/>
            <enumeration value="F"/>
            <enumeration value="A"/>
            <enumeration value="P"/>
            <enumeration value="R"/>
            <enumeration value="O"/>
            <enumeration value="E"/>
            <enumeration value="D"/>
            <enumeration value="K"/>
            <enumeration value="S"/>
            <enumeration value="I"/>
            <enumeration value="J"/>
            <enumeration value="U"/>
            <enumeration value="V"/>
            <enumeration value="W"/>
            <enumeration value="X"/>
        </restriction>
    </simpleType>
<complexType name="ItemTyp" mixed="true">
        <sequence>
            <element name="lowvalue" type="string" minOccurs='0' maxOccurs='1'/>
            <element name="highvalue" type="string" minOccurs='0' maxOccurs='1'/>
        </sequence>
        <attribute name="id" type="positiveInteger"/>
        <attribute name="l" type="nonNegativeInteger"/>
        <attribute name="r" type="nonNegativeInteger"/>
    </complexType>

</schema>
```

Figure A.1 (*continued*)

Bibliography

[1] ISO 80000 (all parts), *Quantities and units*

[2] ISO 1000, *SI units and recommendations for the use of their multiples and of certain other units*

[3] ISO 8859 (all parts), *Information and technology — 8-bit single-byte coded graphic character sets*

[4] IEEE 802.3x, *IEEE Standards for Local and Metropolitan Area Networks: Specification for 802.3 Full Duplex Operation*

[5] RFC 988, *Host extensions for IP multicasting*

[6] RFC 768, *User Datagram Protocol (UDP)*

[7] RFC 761, *Transmission Control Protocol (TCP)*

[8] RFC 791, *Internet Protocol (IP)*

[9] RFC 2396, *Uniform Resource Identifiers (URI): Generic Syntax*

[10] RFC 1738, *Uniform Resource Locators (URL)*

[11] RFC 3548, *Base16, Base32, and Base64 Data Encoding*

[12] Industrial Ethernet Planning and Installation Guide, Release 4.0, 2003, IAONA

[13] *W3C Recommendation: Extensible Markup Language (XML) 1.0*. Second edition, October 2000

[14] XML, *Schema Part 0: Primer*, W3C Recommendation, May 2001

[15] XML, *Schema Part 1: Structures*, W3C Recommendation, May 2001

[16] XML, *Schema Part 2: Datatypes*, W3C Recommendation, May 2001

ICS 35.240.99; 65.060.01

Price based on 71 pages