INTERNATIONAL STANDARD

ISO 14827-2

Second edition
2022-12

Intelligent transport systems — Data interfaces between centres for transport information and control systems —

Part 2: **AP-DATEX**

Reference number
ISO 14827-2:2022(E)

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

Published in Switzerland

Contents

Foreword

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

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

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. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 204, *Intelligent transport systems*.

This second edition cancels and replaces the first edition (ISO 14827-2:2005), which has been technically revised.

The main changes are as follows:

- the title has been modified;
- the concept of a platform-independent model (PIM) as defined in ISO/TS 19468 has been integrated;
- the message format previously defined in ISO 14827-1:2005 (to be withdrawn) has been included.

A list of all parts in the ISO 14827 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user’s national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

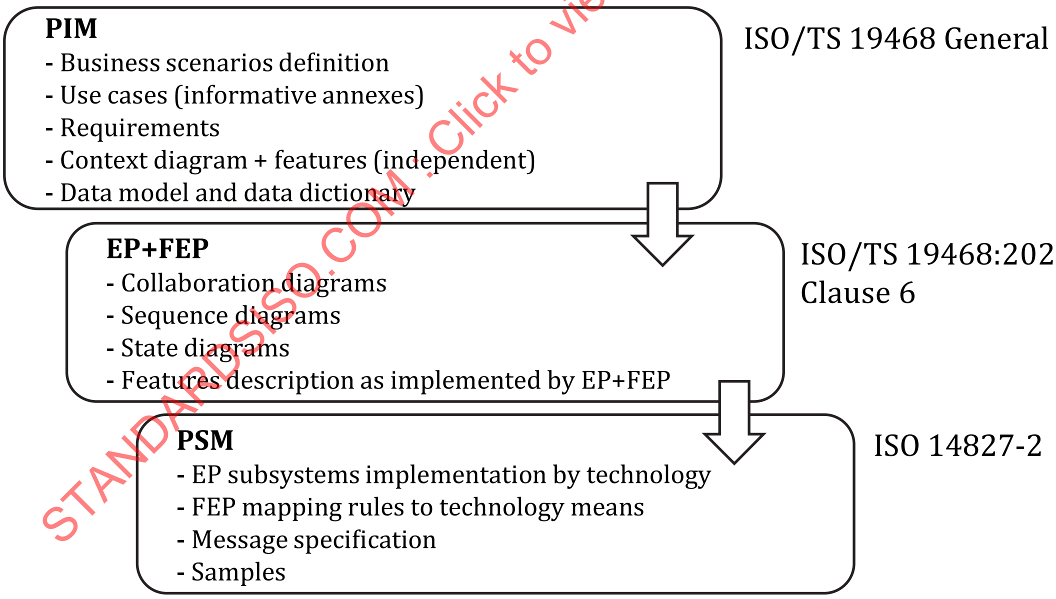
Data exchange among centres is a baseline service for implementing intelligent transport system (ITS) services. For interoperability purposes, data delivery and collaborative ITS services need to be implemented according to certain specifications based on fully-described interfaces.

This document has been revised based on the concept of a platform-independent model (PIM) as defined in ISO/TS 19468, maintaining backward compatibility with ISO 14827-2:2005 and taking into consideration the future withdrawal of ISO 14827-1:2005.

The development of the first editions of ISO 14827-1 and ISO 14827-2 began in the 1990s. These documents were published in 2005 based on European DATEX. Since then, the exchange environment of traffic information and traffic data has made a great deal of progress and DATEX II has been developed, enabling the distribution of traffic information and traffic management information in a way that is not dependent on language and presentation format. DATEX II is closely related to ISO/TS 19468. ISO/TS 19468 aims to describe the general exchange specification technology and to describe interaction through a high-level model which is not dependent on a specific technology in a model-driven approach; it defines functional exchange profiles by several possible exchange patterns.

According to this concept, ISO 14827-2 (this document) was revised as a platform-specific model for AP-DATEX (application profile-data exchange) and other Internet protocol (IP) networks. The relationship between ISO/TS 19468 and the ISO 14827 series (including this document) is shown in Figure 1. This document aims to define and describe the data exchange requirements using TCP/UDP (transmission control protocol/user datagram protocol) datagrams (defined as "DATEX-ASN") and the basics of ASN.1 messages, as defined in ISO 14827-1.

This document is not intended to conflict with existing International Standards on interfaces of data exchange among ITS centres.


PIM
- Business scenarios definition
- Use cases (informative annexes)
- Requirements
- Context diagram + features (independent)
- Data model and data dictionary
ISO/TS 19468 General
EP+FEP
- Collaboration diagrams
- Sequence diagrams
- State diagrams
- Features description as implemented by EP+FEP
ISO/TS 19468:2022, Clause 6
PSM
- EP subsystems implementation by technology
- FEP mapping rules to technology means
- Message specification
- Samples
ISO 14827-2

Key

PIM platform-independent model

EP exchange pattern

FEP functional exchange profile

PSM platform-specific model

Figure 1 — Relationship between exchange-related documents

Intelligent transport systems — Data interfaces between centres for transport information and control systems —

Part 2: **AP-DATEX**

1 Scope

This document defines a platform-specific model (PSM) for data exchange, which specifically uses ASN.1 and TCP/UDP (transmission control protocol/user datagram protocol) datagrams which were defined as "DATEX-ASN" in the first edition of this document for AP-DATEX (application profile-data exchange) and other Internet protocol (IP) networks. A PSM is an actual implementation of a platform-independent model (PIM) for exchange. This document specifies the message rules and procedures for communication between different systems for ITS using TCP/UDP datagrams.

This document deals mainly with the communication interfaces. It has been designed to meet the unique requirements of intelligent transport systems (ITS). However, it has also been designed in a generic fashion and thus can be used for other data exchanges as well.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/TS 19468, *Intelligent transport systems — Data interfaces between centres for transport information and control systems — Platform-independent model specifications for data exchange protocols for transport information and control systems*

ISO/IEC 8824-1, *Information technology — Abstract Syntax Notation One (ASN.1) — Part 1: Specification of basic notation*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/TS 19468 and the following apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org/

3.1
DATEX-ASN
data exchange protocol in abstract syntax notation as TCP/UDP (transmission control protocol/user datagram protocol) datagrams exchange

Note 1 to entry: This was initially defined in ISO 14827-2:2005 (first edition of this document).

3.2
DatexDatapacket
TCP/UDP (transmission control protocol/user datagram protocol) datagrams which are defined in ASN.1 as application layer data packets and can be exchanged using any compatible lower-layer combination

Note 1 to entry: See Annex B.

3.3
guaranteed delivery
TCP/UDP (transmission control protocol/user datagram protocol) datagrams exchange mechanism in which the client acknowledges the receipt of a publication (reply)

3.4
heartbeat
data packet sent to indicate that the sending system is still alive and communicating

3.5
publication
information (usually contained in payload) exchange from a supplier

Note 1 to entry: “Payload publication” is defined in ISO/TS 19468.

3.6
subscription
request to a supplier from a client for information exchange

3.7
transport profile
set of services which are responsible for providing a virtually error-free, point-to-point connection so that host A can send data packets to host B and they will arrive uncorrupted

4 Symbols and abbreviated terms

AP-DATEX	application profile-data
BER	basic encoding rule
CORBA	common object request broker architecture
DATEX-ASN	data exchange in abstract syntax notation
DTLS	datagram transport layer security
EDIFACT	electronic data interchange for administration, commerce and transport
EP	exchange pattern
FDDI	fibre distributed data interface
FEP	functional exchange profile
FrED	friendly exchange of data
FTP	file transfer protocol
IP	Internet protocol
ISDN	integrated services digital network

NTCIP	National Transportation Communications for ITS (intelligent transport systems)
OID	object identifier
PIM	platform-independent model
PN	port number
PPP	point-to-point protocol
PRL	protocol requirements list
PSM	platform-specific model
SNMP	simple network management protocol
TCP	transmission control protocol
TCIP	transit communications interface profiles
TFTP	trivial file transfer protocol
TICS	transport information and control systems
TLS	transport layer security
UDP	user datagram protocol
VMS	variable message sign

5 Conformance

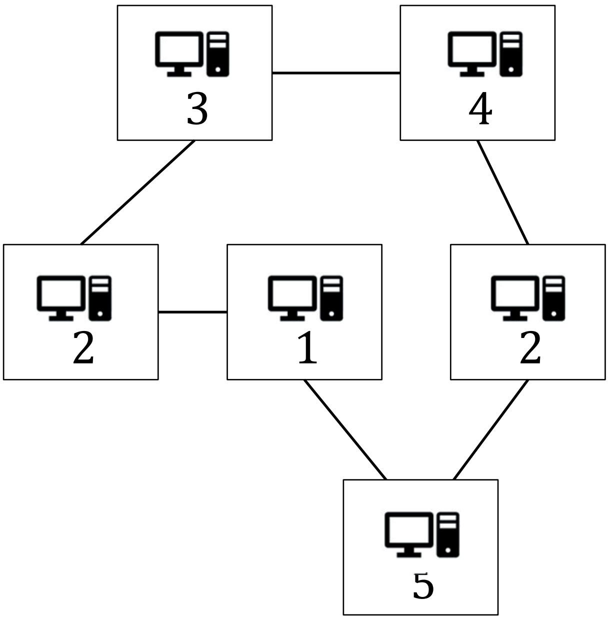
There is no explicit conformance test in this document. Conformance is achieved if the exchange data conform to the messaging rules of this document.

6 Exchange framework

6.1 General

TCP/UDP datagrams exchange allows different systems to exchange relevant data. The data are contained in end-application messages. Each end-application message shall be defined according to message definition requirements laid out in Annex A. TCP/UDP datagrams exchange defines how these end-application messages are packaged to form complete datagrams and also defines the rules and procedures for exchanging these datagrams. Systems using TCP/UDP datagrams exchange may implement additional end-application functionalities according to the user requirements.

A TCP/UDP datagrams exchange network comprises a certain number of systems, an example of which is provided in Figure 2. It is typically exchanged using well-known IPs, such as UDP/IP or TCP/IP, then may use IPsec, DTLS, TLS, etc. for security.



Key

1 weather system

2 traffic management system

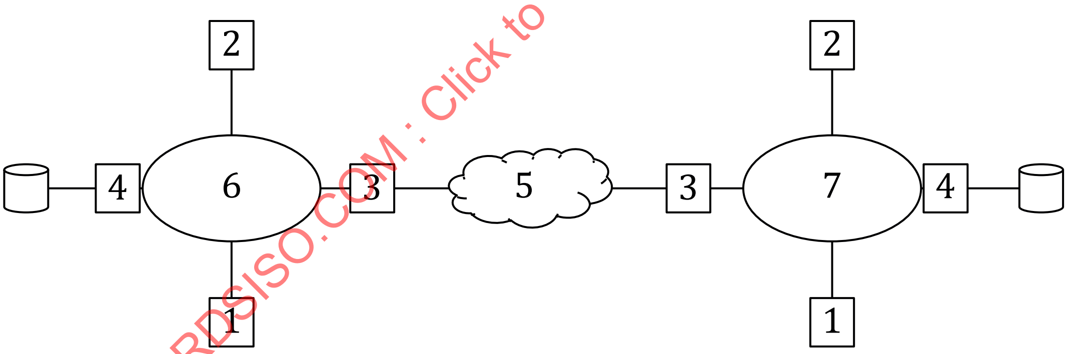
3 transit management system

4 emergency management system

5 information service provider

Figure 2 — An example of TCP/UDP datagrams exchange network

Each system can be viewed as consisting of the interfaces, as shown in Figure 3.



Key

1 application interface

2 operator interface

3 communication interface

4 database interface

5 communications cloud

6 client system

7 supplier system

Figure 3 — System interfaces

The data exchange environment and actors can be viewed as shown in Figure 4.

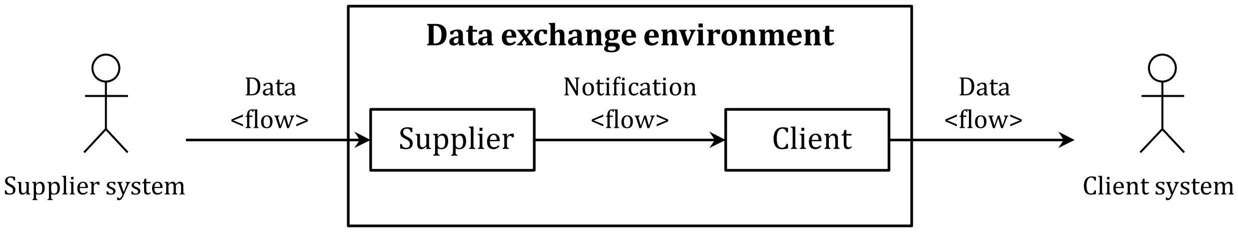


Figure 4 — Communication interfaces

Systems implementing this document sometimes operate simultaneously as a client and supplier, using multiple sessions. The communications cloud between the two systems can be complex or simple.

When implementing a specific PSM, a functional exchange profile (FEP), which is a selection of data exchange features, is identified.

The model driven approach defined in ISO/TS 19468 is summarized in Figure 5.

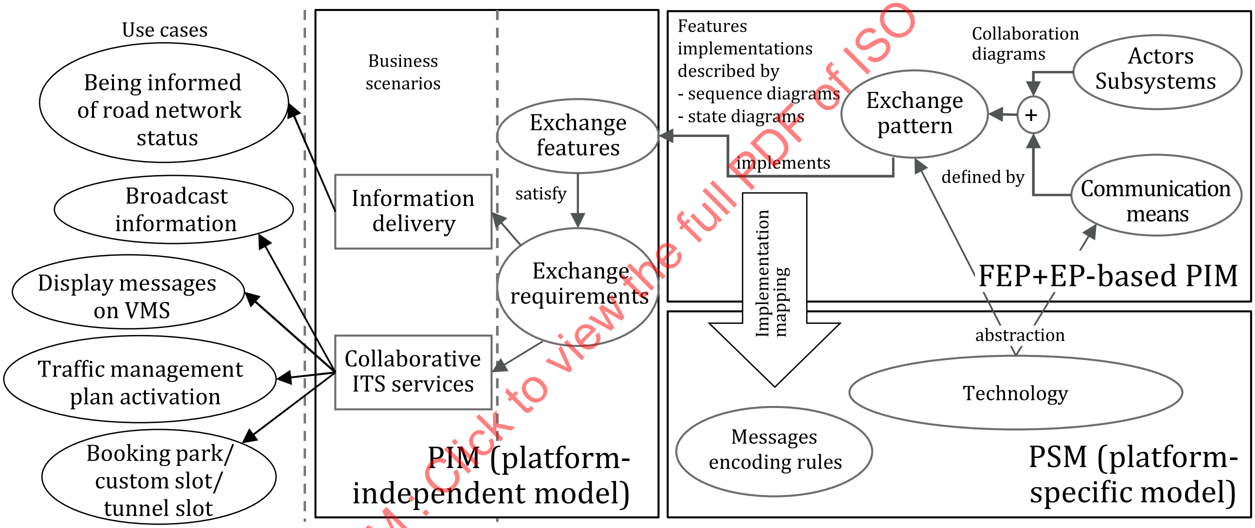


Figure 5 — Business scenario and functional exchange profile (FEP)

This document describes the mapping rules in order to implement specific platform push and pull and FEP+EP based PIM in TCP/UDP datagrams exchange which is based on ASN.1 message PSM. A PIM-level description of FEP+EP is detailed in ISO/TS 19468 and is referenced in this document.

6.2 Basic pull with TCP/UDP datagrams exchange PSM

6.2.1 Overview

The basic pull EP+FEP is based on an information request by a client from a supplier which delivers requested information to the client. It can be implemented in IPs. A selection of features for basic pull is shown in Table 1.

Table 1 — Selection of features for basic pull

Features area	Feature	Basic pull implemented
Subscription contract	Contract	Log in/Log out See 7.4.2, 7.4.4
	Catalogue	N

Table 1 *(continued)*

Features area	Feature	Basic pull implemented
Session	Session life cycle	Log in/Log out/Maintain See 7.4.2, 7.4.4, 7.4.3
	Link monitoring	N
Information management	Operating modes	Periodic or on occurrence (i.e. triggered by client conditions)
	Update methods	N
	Life cycle management	N
Data delivery	Data delivery	Y
	Data request	Y
	Large datasets handling	Y
	Synchronization	Y (periodic mode)
Self-description	Handshake	N
Communication	Security	N
	Compression	N
	Communication	N

6.2.2 Exchange pattern messages definition

6.2.2.1 Overall presentation

Exchange systems are used which provide tools enabling message generation and their transfer between supplier and client. A data flow between the supplier system and client system is shown in Figure 6.

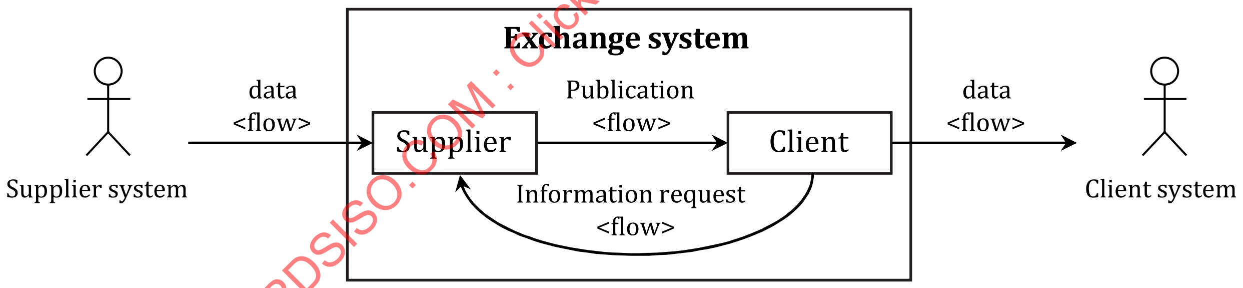


Figure 6 — Basic pull exchange actors

6.2.2.2 Exchange pattern definition

The basic pull client and supplier shall establish, maintain, or terminate a session according to the procedure described in 7.4 of this document.

The basic pull client shall request information according to the procedure described in 7.5 of this document.

The basic pull supplier shall provide information according to the procedure described in 7.6 of this document.

6.2.2.3 Relevant exchange information in exchange data model

No exchange information is needed in this pattern to implement data delivery features.

6.2.2.4 Exchange messages

Exchange messages are included in the payload and defined in Annex B.

6.2.3 State diagrams

State diagrams are not needed, and relevant procedures are described in Clause 7.

6.2.4 Features implementation description

6.2.4.1 Overview

This subclause provides a description and the corresponding specification for each feature identified in the context diagram according to the basic pull exchange architecture. The following features are specified:

- subscription contract;
- subscription (also known as session);
- information management;
- data delivery;
- communication/protocol.

6.2.4.2 Subscription contract

6.2.4.2.1 Contract

The session is established or terminated according to the procedure described in 7.4.

6.2.4.2.2 Catalogue

Catalogue is not managed.

6.2.4.3 Session

6.2.4.3.1 Session life cycle

The session is managed according to the procedure described in 7.4.

6.2.4.3.2 Link monitoring

Link monitoring is not provided.

6.2.4.4 Information management

6.2.4.4.1 Operating modes

The available operating mode for client pull is periodic, or on occurrence as described in 7.6.

6.2.4.4.2 Update methods

The update method is not provided.

6.2.4.4.3 Life cycle management

The life cycle is not managed.

6.2.4.5 Data delivery

6.2.4.5.1 Data delivery scheme

The data delivery scheme is described in 7.6.

6.2.4.5.2 Data request scheme

The data request scheme is described in 7.5.

6.2.4.5.3 Large datasets handling

The large datasets are handled as a data file. The data file scheme is described in 7.6.

6.2.4.5.4 Synchronization

Under periodic mode, publication cycle synchronization is available, and its scheme is described in 7.6.4.

6.2.4.6 Self-description

Handshake is not available.

6.2.4.7 Communication

Communication feature may be implemented at IP level.

6.2.4.8 General optimization issues

Implementation considerations and IP usages for TCP/UDP datagrams exchange shall be as shown in Annex E and Annex F. Requirements of protocol shall be as shown in Annex D.

6.3 Basic push TCP/UDP datagrams exchange PSM

6.3.1 Overview

The basic push EP+FEP is performed from a supplier which delivers information to a client without request by the client. It can be implemented in internet protocols. A selection of features for basic push is shown in Table 2.

Table 2 — Selection of features for basic push

Features area	Feature	Basic push implemented
Subscription contract	Contract	Log in/Log out See 7.4.2, 7.4.4
	Catalogue	N
Session	Session life cycle	Log in /Log out/Maintain See 7.4.2, 7.4.4, 7.4.3
	Link monitoring	N
Information management	Operating modes	Periodic or on occurrence (i.e. triggered by supplier conditions)
	Update methods	N

Table 2 *(continued)*

Features area	Feature	Basic push implemented
	Life cycle management	N
Data delivery	Data delivery	Y
	Data request	N
	Large datasets handling	Y
	Synchronization	Y (periodic mode)
Self-description	Handshake	N
Communication	Security	N
	Compression	N
	Communication	N

6.3.2 Exchange pattern messages definition

6.3.2.1 Overall presentation

Exchange systems are used which provide tools enabling message generation and their transfer between supplier and client. A data flow between supplier and client is shown in Figure 7.

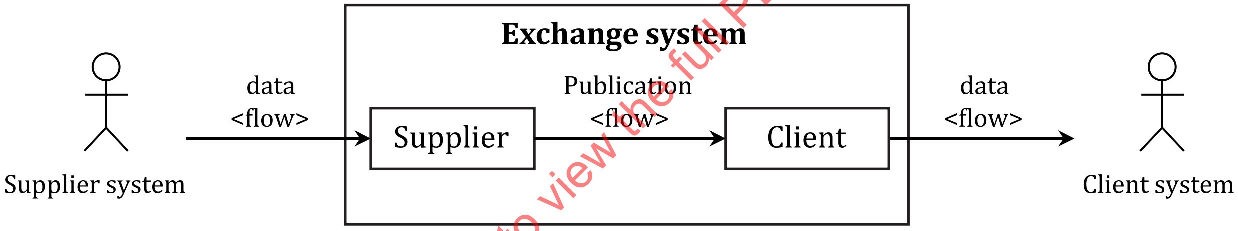


Figure 7 — Basic push exchange actors

6.3.2.2 Exchange pattern definition

The basic push client and supplier shall establish, maintain, or terminate a session according to the procedure described in 7.4 of this document.

The basic push supplier may request receiving information to client and shall provide information according to the procedure described in 7.6 of this document.

6.3.2.3 Relevant exchange information in exchange data model

No exchange information is needed in this pattern to implement data delivery features.

6.3.2.4 Exchange messages

Exchange messages are included in payload and defined in Annex B.

6.3.3 State diagrams

State diagrams are not needed, and procedures are described in Clause 7.

6.3.4 Features implementation description

6.3.4.1 Overview

This subclause provides a description and the corresponding specification for each feature identified in the context diagram, according to the basic pull exchange architecture. The following features are specified:

- subscription contract;
- subscription (also known as session);
- information management;
- data delivery;
- communication/protocol.

6.3.4.2 Subscription contract

6.3.4.2.1 Contract

The session is established or terminated according to the procedure described in 7.4.

6.3.4.2.2 Catalogue

Catalogue is not managed.

6.3.4.3 Session

6.3.4.3.1 Session life cycle

The session is managed according to the procedure described in 7.4.

6.3.4.3.2 Link monitoring

Link monitoring is not provided.

6.3.4.4 Information management

6.3.4.4.1 Operating modes

Available operating mode for client pull is periodic, or on occurrence as described in 7.6.

6.3.4.4.2 Update methods

The update method is not provided.

6.3.4.4.3 Life cycle management

The life cycle is not managed.

6.3.4.5 Data delivery

6.3.4.5.1 Data delivery scheme

The data delivery scheme is described in 7.6.

6.3.4.5.2 Data request scheme

Not implemented in this pattern.

6.3.4.5.3 Large datasets handling

The large datasets are handled as a data file. The data file scheme is described in 7.6.

6.3.4.5.4 Synchronization

Under periodic mode, publication cycle synchronization is available, and its scheme is described in 7.6.4.

6.3.4.6 Self-description

Handshake is not available.

6.3.4.7 Communication

The communication feature may be implemented at other internet protocol level.

6.3.4.8 General optimization issues

Implementation considerations and IP usages for TCP/UDP datagrams exchange shall be as shown in Annex E and Annex F. Requirements of protocol shall be as shown in Annex D.

7 Data exchange procedures

7.1 General

This document defines an application layer protocol by which data elements are exchanged between a client and supplier. Communication between client and supplier shall be accomplished by the exchange of datagrams and files as defined in this clause.

7.2 General datagrams procedures

7.2.1 General

TCP/UDP datagrams shall be constructed according to the formally defined ASN.1 data structure as DatexDataPacket, defined in Annex B.

The data dictionary for each field of DatexDataPacket is accordance with Annex C. The value domain specified in the data dictionary is accordance with Annex D.

7.2.2 Sessions

This document requires all datagrams to be transmitted in an application session. Within each session, one system shall act as a client and the other shall act as the supplier. Multiple sessions may exist simultaneously.

NOTE A pair of systems will have two concurrent sessions, one where System A acts as the client and System B acts as the supplier and the other where System A acts as the supplier and System B acts as the client. These sessions would be distinguished by lower layer protocols (e.g. TCP or UDP port numbers).

7.2.3 Transport requirements

Data may be exchanged over connectionless or connection-oriented transport profiles, but a single transport profile shall be used for all datagrams exchanged within a session.

EXAMPLE If the first datagram in establishing a session is transmitted using UDP, then all datagrams within that session will use UDP. Likewise, if the initial transmission is TCP, then all datagrams will be TCP.

7.2.4 Response time-outs

The client and supplier shall negotiate the response time-out period for each session. The response time-out period should be long enough to accommodate the network propagation delays for both datagrams as well as the turn-around time required to handle the message on the receiving end. In theory, this should be measured from when the last byte is transmitted to when the last response byte is received. However, it is expected that most implementations will measure the time from the return from the system write call to the return from the system read call.

NOTE A typical implementation is to set the time-out to be an integral multiple of the turn-around time and the multiplier is typically set to three. When some communications media and networks experience significant delays, the system will allow this multiplier to be set at run-time.

7.2.5 Retransmission

If a specific datagram requires a response and an appropriate response is not received within the response time-out period, the identical datagram (e.g. same datagram number, same time stamp, etc.) shall be retransmitted one time only. If no response is received to the second datagram, prior to a subsequent response time-out period, the datagram transmission shall be considered unsuccessful. If a response is received after the time-out period, it may be ignored.

7.2.6 Duplicate datagrams

Any time a client or supplier receives a datagram that requires a response, a new response datagram shall be prepared and transmitted as soon as possible, even if the received datagram was a duplicate datagram.

7.3 General file procedures

The client may request the publication (reply) data to be sent within the publication datagram, or the client may request the publication data to be stored in a file on the supplier with the publication datagram indicating the filename of the publication file. The file can then be retrieved by the client within the constraints set by the supplier. Such a publication file shall only contain the “TICS information” as defined by the publication datagram structure as defined in B.2.9.

7.4 Sessions

7.4.1 General

Within each session, one system is a client and the other is a supplier. A supplier with a given domain name shall not accept more than one session with any client domain name with a given transport profile. However, as a single system may have multiple domain names, multiple sessions can potentially exist between a given client system and supplier system pair. Multiple sessions may exist on a single physical link simultaneously.

NOTE 1 For an example of multiple sessions, system A can act as a supplier in one session with system B while also acting as a client in a second session.

NOTE 2 When a single client has sessions with multiple suppliers simultaneously, the complete session number over any given transport profile is defined by the supplier domain name followed by the client domain name.

NOTE 3 When some implementations have a need to frequently publish relatively large datagrams, there are various ways to achieve this, including: (1) increasing the UDP/IP datagram size to support the required size; or (2) maintaining a prolonged TCP connection over which the large datagrams are periodically sent. The preferred solution will depend on a number of implementation-specific issues such as media quality and required reliability of transmission.

NOTE 4 The usage of different transport profiles (e.g. one UDP and one TCP) will allow simultaneous sessions between a single client and supplier pair.

7.4.2 Establishing a session

The supplier may wish to establish a session. For example, this may be in order to publish information for a registered subscription (request) or allow a receipt of a subscription if the supplier is protected by a firewall. In this case, the supplier shall transmit an “Initiate” datagram, as defined in B.2.3, with the datex-Destination-txt and datex-Sender-txt fields set to the proper name.

A supplier should not terminate a session it initiated for a period of one heartbeat duration after final publication.

If the client receives an “Initiate” datagram or if the client wishes to establish a session, the client shall transmit a “Login” datagram, as defined in B.2.4.

Upon receiving a “Login” datagram, a supplier shall determine if the domain names, username, password, maximum heartbeat duration, response time-out period, allowed encoding rules, datagram size and login reason are valid for the request. The supplier shall also ensure that a session with the given domain name and transport profile does not already exist. If the request is found to be invalid, the supplier shall either:

— respond with a “reject” datagram, as defined in B.2.12, with the “error-code” set to the most appropriate code number which applies to the denial, or

— not respond if the supplier determines this is appropriate due to security reasons.

If the request is valid, the supplier shall respond with an “accept” datagram, as defined in B.2.11, and shall identify the selected encoding rules from the list of options in the login request. This completes the procedures to establish a session.

The procedure to establish a session is summarized in Figure 8. All datagrams exchanged during this procedure shall use the encoding rules that were agreed upon offline. All datagrams exchanged after the successful completion of this procedure shall use the encoding rules, as negotiated within the “Login” and “Accept” datagrams.

EXAMPLE As per Annex F, if the session is established over TCP/IP on Port 355, datagrams exchanged during this procedure will use BER encoding; datagrams exchanged after the successful completion of the login process would then use the encoding rules negotiated by the “Login” and “Accept” datagrams.

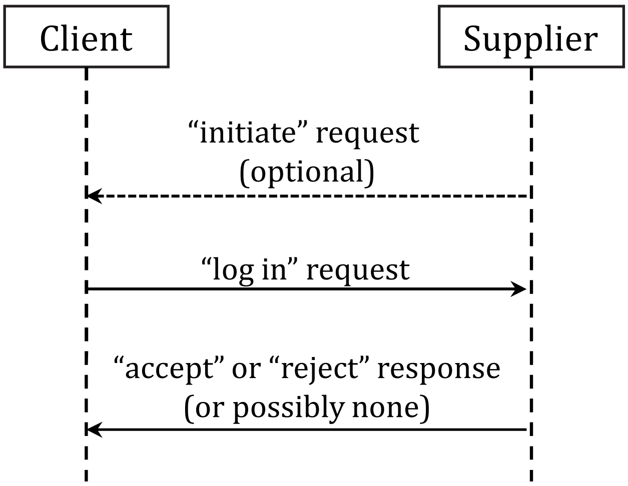


Figure 8 — Establishing a session

7.4.3 Maintaining a session

Sessions are maintained by the client and supplier exchanging “FrED” datagrams. If, at any point during a session, no datagrams are received from the other system for a period exceeding the maximum heartbeat duration, as specified in the login request, the session shall be immediately terminated by both the client and the supplier without exchanging any data. This type of termination is generally only encountered due to unusual circumstances, e.g. a system crash.

NOTE 1 FrED stands for a “Friendly Exchange of Data”. The datagram is generally used as an acknowledgement datagram, but it is also used as a system heartbeat when there has been a prolonged period of silence. Thus, the term “ack” does not truly apply to this datagram and the term “FrED” has been considered more appropriate.

NOTE 2 A session can be kept open permanently by meeting the requirements of this subclause.

The client shall maintain the session until the termination procedures are initiated as indicated in 7.4.4. The client shall keep track of the elapsed time since it received a datagram from the supplier and shall ensure that this time does not exceed the maximum heartbeat duration by generating “FrED” datagrams, as defined in B.2.5, as needed. The DATEX.FrED_ConfirmPacket_number-ulong shall be zero (0) for such FrED datagrams, hereinafter referred to as “FrED heartbeat datagrams". It is recommended that the client transmit FrED heartbeat datagrams roughly three times more often than the time specified by the maximum heartbeat duration.

The supplier shall acknowledge a “FrED” heartbeat datagram by transmitting a “FrED” datagram with the DATEX.FrED_ConfirmPacket_number-ulong set to the packet number of the “FrED” heartbeat datagram which is being acknowledged. This shall complete the session maintenance procedure.

When desired, the session shall be terminated according to the procedure described in 7.4.4.

The procedure to maintain a session is summarized in Figure 9.

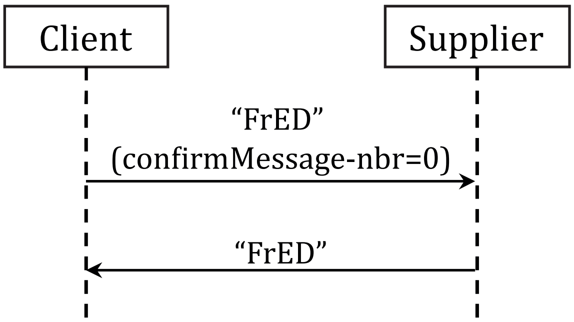


Figure 9 — Maintaining a session

7.4.4 Terminating a session

A session may be actively terminated by either the client or the supplier. If the supplier wishes to terminate a session, it shall transmit a “Terminate” datagram, as defined in B.2.6. If the supplier does not receive any response after two tries, the supplier shall terminate the session on its end.

If the client receives a valid “Terminate” datagram, or if the client wishes to terminate a session, it shall transmit zero or more subscription cancellations, as defined in 7.5 and B.2.8, if it wishes to cancel any persistent subscriptions, followed by a “Logout” datagram, as defined in B.2.7.

NOTE 1 Registered subscriptions do not expire with the termination of a session if the “Persistent” flag was set in the subscription. This allows systems to keep subscriptions active when the session is not active. For example, this can be useful for dial-up connections or to minimize the impact of system crashes.

NOTE 2 A supplier does not need to wait for a FrED for a guaranteed publication to a non-persistent subscription.

Upon receipt of a valid “Logout” datagram, the supplier shall terminate the associated session and issue a FrED, as defined in B.2.5. The client shall terminate the associated session upon receipt of the FrED. This shall complete the session termination procedure.

NOTE If the client does not receive the FrED, despite following the retransmission rules, it will terminate the session according to the provisions of 7.4.3.

The procedure to terminate a session is summarized in Figure 10.

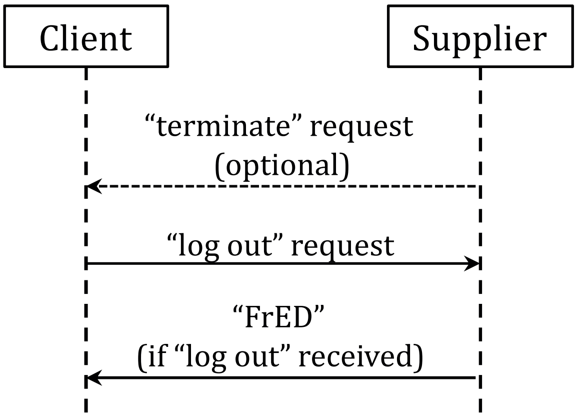


Figure 10 — Terminating a session

7.5 Requesting information

7.5.1 General

Clients and suppliers shall provide for offline subscriptions (requests) as defined in 7.5.2, online subscriptions as defined in 7.5.3, or both.

NOTE The subscription process will take place offline or online. This allows a supplier to transmit publications (replies) (e.g. accident publications) without having to support the associated subscription datagram. This can potentially be desirable in order to bring legacy systems into conformance or for security purposes.

7.5.2 Offline subscriptions

Suppliers may provide local mechanisms to register any and all subscriptions to which the supplier claims compliance. This feature shall be supported for all clients known to the supplier.

Clients may provide local configuration mechanisms to accept publications from a remote supplier so that the client may accept publications related to an offline subscription.

7.5.3 Online subscriptions

A client may support the ability to transmit “subscription” datagrams, as defined in B.2.8. If the client claims conformance for online subscriptions, it shall support this service for all subscriptions to which it claims conformance.

A supplier may accept “subscription” datagrams in order to allow for online requests to be processed. If the supplier claims conformance for online subscriptions, it shall support “subscription” datagrams for all subscriptions to which it claims conformance.

Upon receipt of a subscription datagram, the supplier shall respond with either an "accept" or "reject" datagram, as defined in B.2.11 and B.2.12. An "accept" datagram shall only indicate that the data was properly received and understood by the system; it does not guarantee that the end-application will accept the subscription. For example, if a valid subscription is received, but the owner of the given session is not authorized to receive the requested data, an “accept” datagram would be transmitted, but the end-application would immediately transmit a “publication”, as defined in 7.6, indicating that the subscription has been terminated with a reason of accessDenied.

This shall complete the subscription procedure.

If the subscription was accepted, the supplier shall publish data according to 7.6. A subscription can be cancelled by setting the datexSubscribe-CancelReason-cd field to one of the cancel reasons. The procedure to request information is summarized in Figure 11.

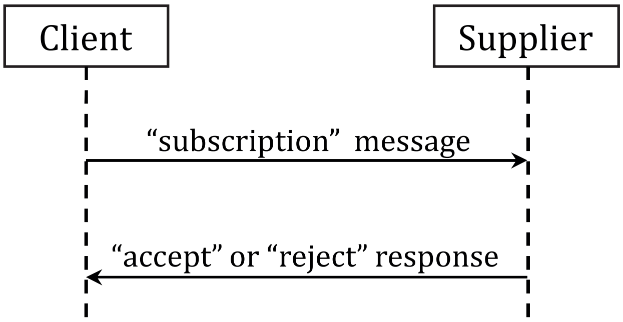


Figure 11 — Subscribing a session

7.6 Publication of information

7.6.1 General

Publication of information is partially dependent upon the type of request. The general procedure is described in 7.6.2; the specific procedures for the different types of requests are described in:

- 7.6.3 for single subscriptions, and
- 7.6.4 for registered subscriptions.

For each subscription to which a system claims conformance, the associated publication shall be supported. Support of the publication datagram is mandatory for all datagrams less than the maximum datagram size. If the system claims conformance to publications that exceed the maximum datagram size, the system shall support the “publication-notice” (see 7.6.2.4) and the file transfer mechanism (see 7.6.2.6); otherwise, support for the “publication-notice” and file transfer mechanism are optional. If a system claims conformance to the “publication-notice” and file transfer mechanism, the system shall support these features for all subscriptions to which it claims conformance. Support for multiple publications within a single datagram or single file is mandatory for clients to receive and optional for suppliers to send.

7.6.2 General procedures

7.6.2.1 General

The supplier shall generate a “publication” datagram, as defined in B.2.9, at the times specified in 7.6.3 and 7.6.4.

7.6.2.2 Guaranteed flag

The datexPublish-Guaranteed-bool shall be set to “true” if the associated subscription datagram requested guaranteed delivery. Otherwise, it shall be set to “false”.

7.6.2.3 Datagram publications

If the client subscribed for datagram publications, the publication datagram shall contain the publication data if the resulting datagram is smaller than the maximum datagram size defined in the Interchange agreement.

NOTE A memory table would most typically be transmitted as a datagram, although it could also be transferred as a file by creating a virtual disk in memory.

7.6.2.4 Publication-notice datagram

If the client subscribed for file publications, or if the resulting datagram is larger than the maximum datagram size, the data shall be stored in a file and the publication datagram shall indicate the path and filename of the publication data.

NOTE The file transfer uses standard file transfer protocols; thus, the file is exchanged in a separate session from the one used for the TCP/UDP datagrams exchange datagrams described in this document.

7.6.2.5 Client response

If the client determines that the publication is improperly encoded, except for any PublicationData structure that may be present, it shall issue a “reject” datagram, as defined in B.2.12. Otherwise, if datexPublish-Guaranteed-bool is set to true, the client shall issue an “accept” datagram, and if datexPublish-Guaranteed-bool is set to false, no response shall be sent.

NOTE 1 Any errors within the PublicationData structure are handled by the procedures in 7.6.2.7.

This shall complete the publication procedures if the “publication” datagram was invalid or if the “publication” datagram indicated a filename and that file was previously downloaded and not yet acknowledged with a transfer done datagram. If the “publication” datagram indicated a file that has not been previously downloaded, the procedures of 7.6.2.6 and 7.6.2.7 shall follow in order. If the “publication” datagram contained the publication data, the procedures of 7.6.2.7 shall follow.

NOTE 2 A duplicate filename can be received due to a duplicate message being sent because of a communications error or due to a recycling of message names. Suppliers can wish to name files with large sequential numbers to avoid duplicate filenames and prevent any loss of data. If the client has already been notified of the publication, there is no reason to start another file transfer process.

7.6.2.6 File transfer mechanism

If the “publication” datagram indicated a new filename (i.e. it was not a duplicate message), the client shall immediately retrieve the indicated file after sending the accept datagram, if required. The file transfer shall be via one of the supported file transfer mechanisms, as negotiated in the “subscription” and “publication” datagrams. Once the file has been transferred (or the client has exceeded its maximum number of tries to download the file), the client shall transmit a “transfer-done” datagram, as defined in B.2.10, to verify the receipt (or notify the supplier of failure). The supplier shall acknowledge the “transfer-done” datagram with a “FrED” datagram, as defined in B.2.5. The supplier shall attempt to keep the file available for downloading until the "transfer-done" notice is received.

7.6.2.7 Reject invalid publication data

For each invalid PublicationData structure contained within the publication (due to invalid encoding), the client shall send a reject datagram. This shall complete these procedures.

The procedure to publish information is summarized in Figure 12. A supplier may cancel any subscription at any time by sending a publication datagram with the publicationType field of the PublicationData structure set to one of the terminate codes.

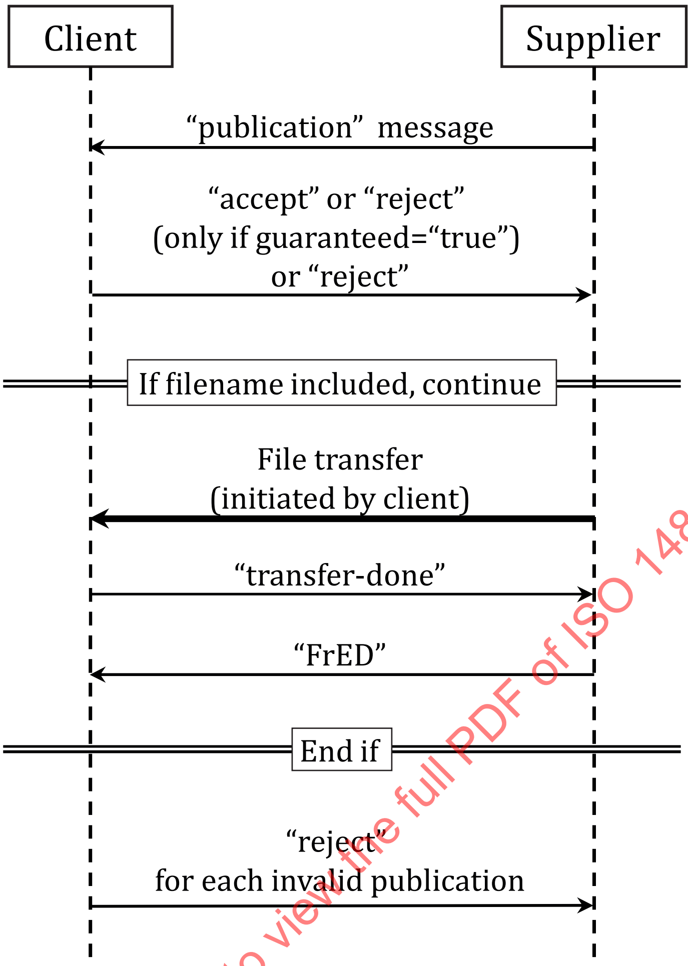


Figure 12 — Publishing information

NOTE The file transfer in the above exchange is a complex procedure, as defined by the associated file transfer standard. The file transfer is requested by the client but is shown as an arrow from the supplier to indicate that the file is on the supplier and is being sent to the client.

7.6.3 Single subscriptions

For one-time, or “single” requests, the supplier shall publish the requested data as soon as possible after completion of the “subscription” process. The publication shall contain all data fulfilling the subscription request.

NOTE Historical data are considered to be separate data elements; thus, historical data can be retrieved as any other information.

7.6.4 Registered subscriptions

Subscriptions may also be “registered” in order to request information as it becomes available or on a periodic basis.

a) Registered subscriptions shall either be continuous or daily (i.e. activated on specific days of the week).

   1) If the subscription is continuous, the subscription shall activate at the datexRegistered-StartTime and remain activated 24 hours a day, seven days a week, until the subscription expires (i.e. reaches the datexRegistered-EndTime) or is explicitly cancelled by a subsequent

subscription. If the datexRegistered-EndTime is less than or equal to the datexRegistered-StartTime, then no publication shall be made.

2) If the subscription is “daily”, the subscription shall become active within the supplier at the start time (datexRegistered-StartTime) on each valid day of week (datexRegistered-DaysOfWeek-cd) occurring on or after the specified start date (datexRegistered-StartDate) and before or on the specified end date (datexRegistered-EndDate). It shall be immediately activated if the current time is valid when the subscription is registered. The subscription shall deactivate at the end of the period defined by the start time (datexRegistered-StartTimeOfDay) plus the duration (datexRegistered-Duration). The end date shall not be earlier than the start date. A daily subscription may also be deactivated by a request that cancels or modifies the subscription. If the datexRegistered-EndDate is less than the datexRegistered-StartDate, then no publication shall be made.

b) Upon subscription activation (or re-activation), the supplier shall publish an initial publication message. If the supplier is unable to provide information at the indicated start time, it shall provide information as soon as possible.

c) Registered subscriptions shall also be classified as either: a) event-driven (i.e. provides information when a specific event occurs); or b) periodic (i.e. provides information at a defined frequency).

d) After the initial publication is produced, the supplier shall attempt to produce a subsequent publication message as follows.

1) If the mode is “periodic”, the supplier shall attempt to produce a new publication periodically at a frequency as defined by datexRegistered-UpdateDelay-qty. If the subscription is sent after the start time, the cycle shall be synchronized with the datexRegistered-StartTime.

NOTE 1 In this case, the initial publication will be at a random point in the cycle, and the second publication can follow at any fraction of the cycle later, but will occur on a cycle point as measured from the datexRegistered-StartTime.

In the periodic mode, a supplier should publish information at every cycle point. If the supplier is unable to publish the information within a period of 60 % of a cycle beyond the cycle point, the publication should not be transmitted. Both the supplier and the client should terminate less important subscriptions (e.g. as reflected in the datexSubscription-Priority field) to minimize the probability of this occurring.

NOTE 2 For example, it is assumed that there is a periodic subscription for data every second. The supplier responds originally at the start time, the next publication (containing information about conditions at 1,0 s) is sent at 1,25 s, the next publication (indicating conditions at 2,0 s) is sent at 2,5 s, and the next publication (containing data valid for 3,0 s) is not ready until 3,75 s. This last publication will be ignored (i.e. it is more than 0,6 s late) and the supplier and client are recommended to consider cancelling less important subscriptions. The intent is to not send old data and have the systems build up an infinite backlog of messages to send while the system is clearly at maximum capacity. Depending on message content and system design, this can sometimes be practical problem. Thus, this clause is provisional and it is intended that the implementation will take whatever appropriate actions are necessary to solve these problems.

2) If the mode is “event-driven”, the supplier shall produce a publication within a period of datexRegistered-UpdateDelay-qty after the supplier is notified of an event. Thus, in this case, the datexRegistered-UpdateDelay-qty parameter serves as a maximum latency value for event reporting. The subscription message shall define the term “event” in the definition and/or message body. If the maximum latency is exceeded, the data shall be published as soon as possible and the datexPublish-LatePublicationFlag shall be set. Suppliers should terminate less important subscriptions (as reflected in the datexSubscription-Priority field) to minimize the probability of this occurring.

Annex A
(normative)

Message definition requirements

A.1 General

Exchange information is defined in ISO/TS 19468. This annex provides messages definition requirements for an end application message conforming to this document.

An end application message conforming to this document shall be formally defined with the attributes defined in the following clauses.

A.2 provides an explanation of the message elements for TCP/UDP datagrams exchange. A.3 provides the Message Container class diagram based on ISO/TS 19468. A.4 provides formal ASN.1 information object specification used to document these attributes.

A.2 Message elements

A.2.1 Name

Each message shall be assigned a unique, descriptive name. This name may be used by some protocols for identification purposes.

A.2.2 Definition

Each message shall be assigned a formal, textual definition. The textual definition may reference figures and other information as appropriate. The definition shall provide a meaningful description of the message and clearly indicate the functionality required by the end system.

If the message can be used in an event-driven mode, the definition will define what constitutes an event.

A.2.3 Remarks

The message specification may also include additional remarks. Such remarks shall be considered to be informative only, with the intention of providing a better understanding of the message or of providing supplemental information for the reader.

A.2.4 Message body

The message body shall be completely defined within this field as an ASN.1 type using the notation as defined in ISO/IEC 8824-1.

This allows a consistent methodology for documenting the data; it does not imply the use of ASN.1 encoding rules. A protocol may select any of the ASN.1 encoding rules or use other rules and procedures (e.g. as defined for EDIFACT and CORBA).

NOTE This element corresponds to PayloadPublication of "MessageContainer" package (see ISO/TS 19468:2022, Annex C).

A.2.5 Message type

The message type shall indicate if the message defines a subscription or publication message structure. Subscription message specifications shall include the “subscription-type”, “initial publication”, and “subsequent publications” attributes. Publication message specifications shall not include these attributes.

NOTE This element corresponds to ExchangeInformation of "MessageContainer" package. (See ISO/TS 19468:2022, Annex C).

A.2.6 Subscription type

The subscription type shall be included in the message specification if the message type is “subscription”. This attribute indicates under what modes the message may be used, as follows:

On occurrence:

— single: An instance of this message shall only be valid when the request is for a single response. The response will be indicated by the “initial publication” attribute.

Periodic:

— periodic: An instance of this message shall only be valid when the request is for periodic updates. Upon receipt, the receiver shall send an instance of the “initial publication” and the supplier shall periodically send “subsequent publications” updates according to the rules of the specific protocol.

— single-event-periodic: An instance of this message may request a single response, an event-driven response, or a periodic response. The mode (either single, event or periodic) shall be explicitly indicated within the message instance. The action of the receiving system shall be as appropriate given the mode of the message instance.

Condition-triggered:

— event-driven: An instance of this message shall only be valid when the request is for event-driven notification. Upon receipt, an instance of the “initial publication” shall be sent, and when the defined event is detected, an instance of the “subsequent publications” shall be sent. The event shall be defined within the definition attribute.

Other:

— single-or-event: An instance of this message shall only be valid if the subject request is for a single response or if the subject request is for an event-driven response; an instance of this message may not be sent as a periodic request. The mode (either single or event) shall be explicitly indicated within the message instance. The action of the receiving system shall be as appropriate given the mode of the message instance.

— single-or-periodic: An instance of this message shall only be valid if the subject request is for a single response or if the subject request is for a periodic response; an instance of this message may not be sent as an event-driven request. The mode (either single or periodic) shall be explicitly indicated within the message instance. The action of the receiving system shall be as appropriate given the mode of the message instance.

— event-or-periodic: An instance of this message shall only be valid if the subject request is for an eventdriven response or a periodic response; an instance of this message may not be sent to request a single response. The mode (either event or periodic) shall be explicitly indicated within the message instance. The action of the receiving system shall be as appropriate given the mode of the message instance.

NOTE This element corresponds to ExchangeInformation of "MessageContainer" package (see ISO/TS 19468:2022, Annex C).

A.2.7 Initial publication

The initial publication attribute shall only be included in the specification of subscription messages. It shall define the message that will be transmitted upon receipt of the associated subscription.

NOTE This element corresponds to InformationManagement of "MessageContainer" package (see ISO/TS 19468:2022, Annex C).

A.2.8 Subsequent publications

The subsequent publications attribute shall only be included in the specification of subscription messages that are not of message type “single”. It shall define the publication message that will be transmitted for all subsequent messages after the initial publication.

NOTE This element corresponds to InformationManagement of "MessageContainer" package. (see ISO/TS 19468:2022, Annex C).

A.2.9 ID

Each message shall be assigned a globally unique, ASN.1 object identifier in the ID field. Some protocols may use this identifier to identify the message type rather than the message name.

NOTE This identifier is used to distinguish not exchange information which is actually transmitted, but type of message.

A.3 Message container class diagram

The message container class diagram is shown in Figure A.1.

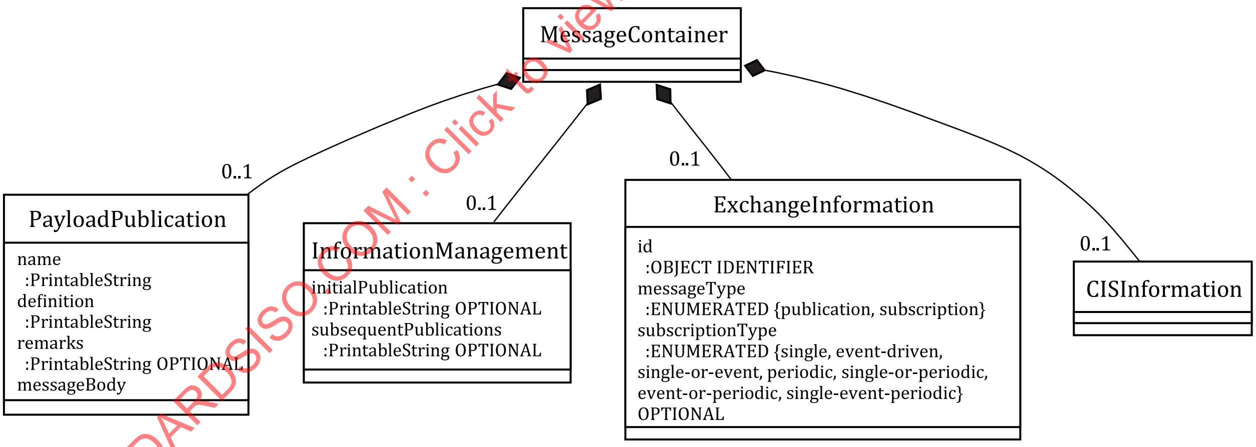


Figure A.1 — Message Container class diagram

A.4 Information object specification

```
ISO 14827-MESSAGE ::= CLASS {

    &name                          PrintableString (SIZE (0..255))

    &definition                    PrintableString (SIZE (0..65535))

    &remarks                       PrintableString (SIZE (0..2000)) OPTIONAL

    &MessageBody

    &messageType                   ENUMERATED {publication, subscription}
    &subscriptionType               ENUMERATED {single, event-driven, single-or-event,
                                    periodic, single-or-periodic, event-or-periodic, single-
                                    event-periodic} OPTIONAL

    &initialPublication             PrintableString (SIZE (0..255)) OPTIONAL

    &subsequentPublications         PrintableString (SIZE (0..255)) OPTIONAL

    &id                             OBJECT IDENTIFIER

}


WITH SYNTAX {

    NAME                            &name

    DEFINITION                      &definition

    [REMARKS                        &remarks]

    MESSAGE BODY                    &MessageBody

    MESSAGE TYPE                    &messageType

    [SUBSCRIPTION TYPE              &subscriptionType]

    [INITIAL-PUBLICATION            &initialPublication]

    [SUBSEQUENT-PUBLICATIONS        &subsequentPublications]

    ID                              &id

}
```

Annex B
(normative)

Datagram structures

B.1 General

TCP/UDP datagrams are defined in ASN.1 as DatexDataPacket of Application Layer datagrams and can be exchanged using any compatible lower-layer combination. All TCP/UDP datagrams shall conform to the DatexDataPacket structure (and appropriate substructures) as the ASN.1 module defined in B.2. Each field identified in this module is formally defined in Annex C of this document.

B.2 Datex Datagram structure

B.2.1 DatexDataPacket

```
ISO14827-2 {iso(1) standards(0) std14827(14827) part2(2)} DEFINITIONS AUTOMATIC TAGS ::=
BEGIN

DatexDataPacket ::= SEQUENCE {
  datex-Version-cd              ENUMERATED {
                                  experimental,
                                  version-1,
                                  ...
                                },
  datex-Data-txt                OCTET STRING,
  -- an optionally encrypted C2CAuthenticatedMessage
  datex-Crc-id                  OCTET STRING (SIZE (2))
}

C2CAuthenticatedMessage ::= SEQUENCE {
  datex-AuthenticationInfo-txt  OCTET STRING (SIZE (0..255)),
  datex-DataPacket-nbr          INTEGER (0..4294967295),
  datex-DataPacketPriority-cd   INTEGER (0..10),
  options                       HeaderOptions,
  pdu                           PDUs
}

HeaderOptions ::= SEQUENCE {
  datex-Origin-txt              UTF8String (SIZE (0..40))     OPTIONAL,
  datex-OriginAddress-loc       OCTET STRING                  OPTIONAL,
  datex-Sender-txt              UTF8String (SIZE (0..40))     OPTIONAL,
  datex-SenderAddress-loc       OCTET STRING                  OPTIONAL,
  datex-Destination-txt         UTF8String (SIZE (0..40))     OPTIONAL,
  datex-DestinationAddress-loc  OCTET STRING                  OPTIONAL,
  cost                          Cost                          OPTIONAL,
  datex-DataPacketTime          Time                          OPTIONAL
}

Cost ::= SEQUENCE {
  amount-Currency-cd            OCTET STRING (SIZE (3)),
  amount-Factor-qty             INTEGER,
  amount-Quantity-qty           INTEGER
}
```

B.2.2 Protocol data unit

The PDU structure allows multiple types of datagrams to be sent in the same overall structure as defined in B.2.1. The various types of structures that can be contained in the PDU are described as follows:

- initiate: allows the sever to request a new session;
- login: checks passwords, etc., and manages who is online;
- FrED: a “friendly exchange of data” used to confirm receipt of datagrams and to maintain a session when there are periods of silence;
- terminate: when the supplier (server) has to discontinue the session;
- logout: allows the client to discontinue the session;
- subscription: requests data (may be one-time only or registered);
- publication: provides the requested data;
- transfer-done: allows the client to notify the supplier (server) that a publication file has been retrieved;
- accept: accepts a login, subscription or publication;
- reject: rejects a login, subscription or publication.

```
PDUs ::= CHOICE {
  initiate                             Initiate,
  login                                Login,
  fred                                 FrED,
  terminate                            Terminate,
  logout                               Logout,
  subscription                         Subscription,
  publication                          Publication,
  transfer-done                        TransferDone,
  accept                               Accept,
  reject                               Reject
}
```

B.2.3 Initiate datagram structure

```
Initiate ::= SEQUENCE {
  datex-Sender-txt                     UTF8String (0..40),
  datex-Destination-txt                UTF8String (0..40)
}
```

B.2.4 Login datagram structure

```
Login ::= SEQUENCE {
  datex-Sender-txt                     UTF8String (0..40),
  datex-Destination-txt                UTF8String (0..40),
  datexLogin-UserName-txt              OCTET STRING,
  datexLogin-Password-txt              OCTET STRING,
  datexLogin-EncodingRules-id          SEQUENCE OF OBJECT IDENTIFIER,
  datexLogin-HeartbeatDurationMax-qty  INTEGER (0..65535),
  datexLogin-ResponseTimeOut-qty       INTEGER (0..255),
  datexLogin-Initiator-cd              ENUMERATED {
                                         serverInitiated,
                                         clientInitiated,
                                         ...
                                       },
  datexLogin-DatagramSize-qty          INTEGER (0..65535)
}
```

B.2.5 FrED datagram structure

```
FrED ::= INTEGER (0..4294967295)       -- datexFrED-ConfirmPacket-nbr
```

B.2.6 Terminate datagram structure

```
Terminate ::= ENUMERATED {             -- datexTerminate-Reason-cd
  other,
  serverRequested,
  clientRequested,
  serverShutdown,
  clientShutdown,
  serverCommProblems,
  clientCommProblems,
  ...
}
```

B.2.7 Logout datagram structure

```
Logout ::=  ENUMERATED {               -- datexLogout-Reason-cd
  other,
  serverRequested,
  clientRequested,
  serverShutdown,
  clientShutdown,
  serverCommProblems,
  clientCommProblems,
  ...
}
```

B.2.8 Subscription datagram structure

```
Subscription ::= SEQUENCE {
  datexSubscribe-Serial-nbr           INTEGER (0..4294967295),
  type                                SubscriptionType,
  ...
}

SubscriptionType ::= CHOICE {
  subscription                        SubscriptionData,
  datexSubscribe-CancelReason-cd      ENUMERATED {
                                        other,
                                        dataNotNeeded,
                                        errorsInPublication,
                                        pendingLogout,
                                        processingMgmt,
                                        bandwidthMgmt,
                                        ...
                                      }
}

SubscriptionData ::= SEQUENCE {
  datexSubscribe-Persistent-bool     BOOLEAN,
  datexSubscribe-Status-cd           ENUMERATED {
                                       new,
                                       update
                                     },
  mode                               SubscriptionMode,
  datexSubscribe-PublishFormat-cd    ENUMERATED {
                                       other,
                                       ftp,
                                       tftp,
                                       dataPacket,
                                       ...
                                     },
  datexSubscribe-Priority-cd         INTEGER(1..10),
  datexSubscribe-Guarantee-bool      BOOLEAN,
  message                            EndApplicationMessage
}

SubscriptionMode ::= CHOICE {
  single                             Null,
  event-driven                       Registered,
  periodic                           Registered
}
```

```
Registered::= CHOICE {
  Continuous      SEQUENCE {
                    datexRegistered-UpdateDelay-qty    INTEGER
                        (0..4294967295)  DEFAULT 0,
                    -- 0 means as soon as possible
                    datexRegistered-StartTime         Time    OPTIONAL,
                    -- defaults to immediate
                    datexRegistered-EndTime           Time    OPTIONAL
                    -- defaults to "until cancelled"
                  },
  Daily           SEQUENCE {
                    datexRegistered-UpdateDelay-qty     INTEGER
                                         (0..4294967295)  DEFAULT 0,
                    -- 0 means as soon as possible
                    datexRegistered-DaysOfWeek-cd   BIT STRING {
                                                      other,
                                                      sunday,
                                                      monday,
                                                      tuesday,
                                                      wednesday,
                                                      thursday,
                                                      friday,
                                                      saturday
                                                    } (SIZE (8)),
                   datexRegistered-StartDate        Time    OPTIONAL,
                   -- defaults to immediate
                   datexRegistered-EndDate          Time    OPTIONAL,
                   -- defaults to "until cancelled"
                   datexRegistered-StartTime       Time    OPTIONAL,
                   -- defaults to midnight
                   datexRegistered-Duration-qty    INTEGER (0..65535)
                                                          OPTIONAL

                   -- defaults to 1440 (i.e., 24 hours)
                  }
}

Time ::= SEQUENCE {
  time-Year-qty            INTEGER (-32768..32767)    OPTIONAL,
   -- defaults to current year unless otherwise specified
  time-Month-qty           INTEGER (1..12)            OPTIONAL,
   -- defaults to current month unless otherwise specified
  time-Day-qty             INTEGER (1..31)            OPTIONAL,
   -- defaults to current day unless otherwise specified
  time-Hour-qty            INTEGER (0..23)            DEFAULT 0,
  time-Minute-qty          INTEGER (0..59)            DEFAULT 0,
  time-Second-qty          INTEGER (0..60)            DEFAULT 0,
  secondFractions          CHOICE {
                             time-Deciseconds-qty       INTEGER (0..9),
                             time-Centiseconds-qty      INTEGER (0..99),
                             time-Milliseconds-qty      INTEGER (0..999),
                             ...
                           }   DEFAULT 0,
  timezone                 SEQUENCE {
                             time-TimeZoneHour-qty         INTEGER (-
13. .13)     DEFAULT 0,
                             time-TimeZoneMinute-qty       INTEGER (0. .59)
DEFAULT 0
                           } OPTIONAL
        -- defaults to UTC
}
```

B.2.9 Publication datagram structure

```
Publication ::= SEQUENCE {
  datexPublish-Guaranteed-bool           BOOLEAN,
  format                                 Publish-Format
}

Publish-Format ::= CHOICE {
  data                                   SEQUENCE OF PublicationData,
  datexPublish-FileName-txt              UTF8String (SIZE (0..2000))
}
-- Support for sending/receiving multiple publications in a single
-- datagram or file is optional; however, the ‘data’ field of the
-- Publish-Format structure must still be coded as a SEQUENCE OF
-- structure.

PublicationData ::= SEQUENCE {
   datexPublish-SubscribeSerial-nbr       INTEGER (0..4294967295),
   datexPublish-Serial-nbr                INTEGER (0..4294967295),
   datexPublish-LatePublicationFlag-bool  BOOLEAN,
   publicationType                        PublicationType
}

PublicationType : := CHOICE {
  datexPublish-Management-cd     ENUMERATED {
                                   temporarilySuspended,
                                   resume,
                                   terminate-other,
                                   terminate-dataNoLongerAvailable,
                                   terminate-publicationsBeingRejected,
                                   terminate-PendingShutdown,
                                   terminate-processingMgmt,
                                   terminate-bandwidthMgmt,
                                   terminate-accessDenied,
                                   unknownRequest,
                                   ...
                                 },
  publicationData                EndApplicationMessage
}

EndApplicationMessage ::= SEQUENCE {
  endApplication-Message-id      ISO14827-MESSAGE.&id,
  endApplication-Message-msg     ISO14827-MESSAGE.&MessageBody
}
```

B.2.10 Transfer done datagram structure

```
TransferDone ::= SEQUENCE {
  datexTransferDone-FileName-txt UTF8String (SIZE (0..2000)),
  datexTransferDone-Success-bool BOOLEAN
}
```

B.2.11 Accept datagram structure

```
Accept ::= SEQUENCE {
  datexAccept-Packet-nbr  INTEGER (0..4294967295),
  acceptType              CHOICE {
                            datexAccept-Login-id       OBJECT IDENTIFIER,
                            -- encoding rules
                            single-subscription        NULL,
                            datexAccept-Registered-nbr INTEGER
                               (0..4294967295),
                            -- the accepted value for the UpdateDelay
                            -- parameter
                            Publication                NULL
                          }
}
```

B.2.12 Reject datagram structure

```
Reject ::= SEQUENCE {
  datexReject-Packet-nbr  INTEGER (0..4294967295),
  rejectType              RejectType,
  alternateRequest        AlternateRequest    OPTIONAL
}

RejectType ::= CHOICE {
  datexReject-Login-cd         ENUMERATED {
                                 other,
                                 unknownDomainName,
                                 accessDenied,
                                 invalidNamePassword,
                                 timeoutTooSmall,
                                 timeoutTooLarge,
                                 heartbeatTooSmall,
                                 heartbeatTooLarge,
                                 sessionExists,
                                 maxSessionsReached,
                                 ...
                               },
  datexReject-Subscription-cd  ENUMERATED {
                                 other,
                                 unknownSubscriptionNbr,
                                 invalidTimes,
                                 frequencyTooSmall,
                                 frequencyTooLarge,
                                 invalidMode,
                                 publishFormatNotSupported,
                                 unknowSubscriptionMsgId,
                                 invalidSubscriptionMsgId,
                                 invalidSubscriptionContent,
                                 ...
                               },
  datexReject-Publication-cd   ENUMERATED {
                                 other,
                                 invalidPublishFormat,
                                 ...
                               },
  rejectPublicationData        SEQUENCE {
                                 datexReject-SubscriptionSerial-nbr
                                    INTEGER (0..4294967295),
                                 datexReject-PublicationSerial-nbr
                                    INTEGER (0..4294967295),
                                 datexReject-PublicationData-cd
                                    ENUMERATED {
                                      other,
                                      unknownSubscription,
                                      unknownPublicationNbr,
                                      unknownPublicationMsgId,
                                      invalidPublicationMsgId,
                                      invalidPublicationMsgContent,
                                      repeatedPublicationNbr,
                                      ...
                                    }
                               }
}

AlternateRequest ::= SubscriptionType
```

B.3 DatexDataPacket diagram

The structure of DatexDataPacket is shown in Figure B.1.

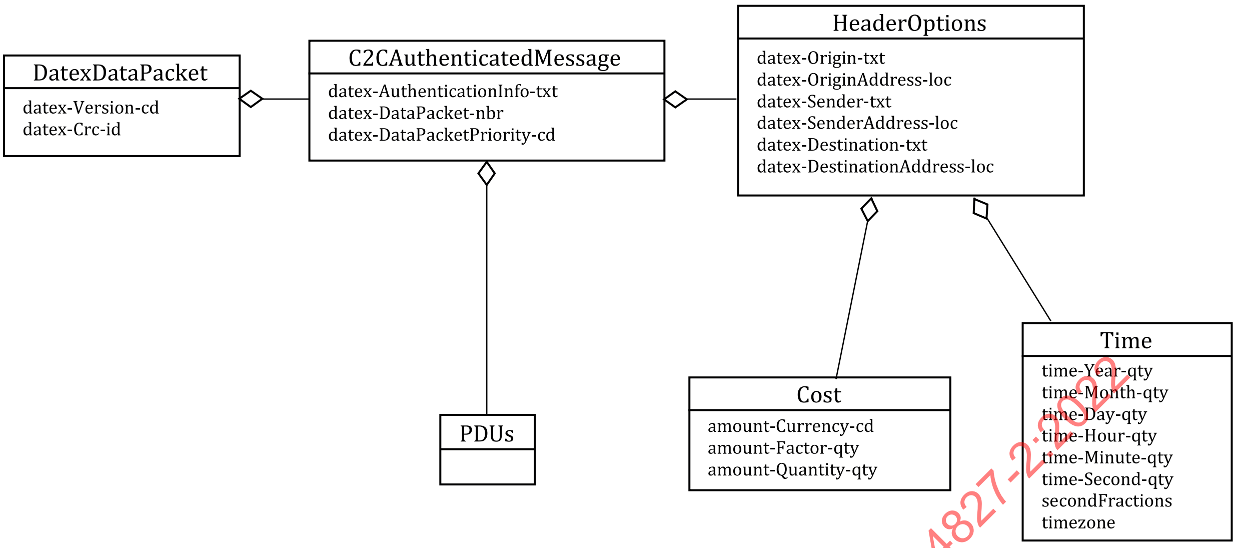


Figure B.1 — Structure of DatexDataPacket

The structure of PDUs is shown in Figure B.2.

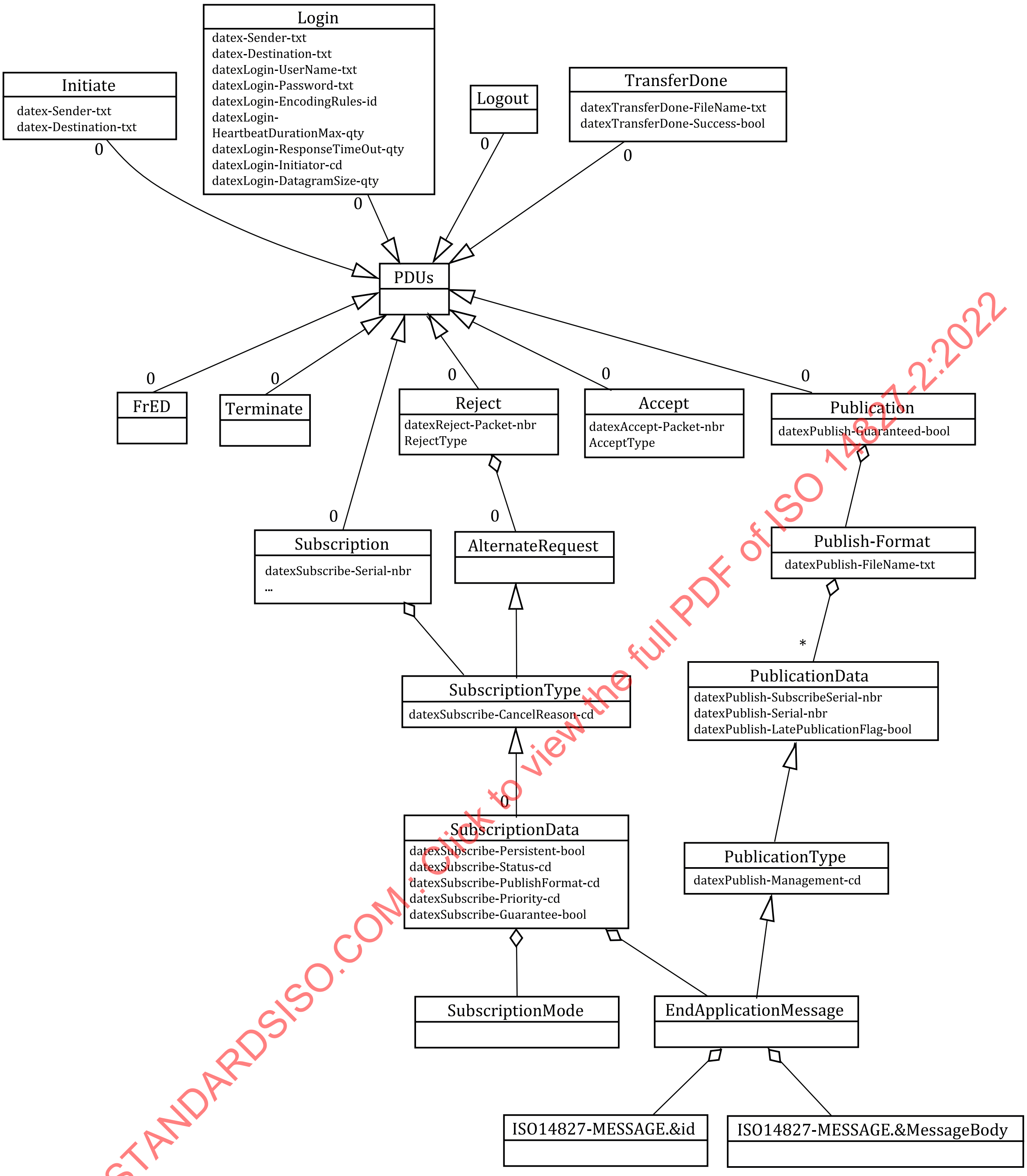

Login
datex-Sender-txt
datex-Destination-txt
datexLogin-UserName-txt
datexLogin-Password-txt
datexLogin-EncodingRules-id
datexLogin-HeartbeatDurationMax-qty
datexLogin-ResponseTimeOut-qty
datexLogin-Initiator-cd
datexLogin-DatagramSize-qty
Initiate
datex-Sender-txt
datex-Destination-txt
Logout
TransferDone
datexTransferDone-FileName-txt
datexTransferDone-Success-bool
0
0
0
0
PDUs
0
0
0
0
0
FrED
Terminate
Reject
datexReject-Packet-nbr
RejectType
Accept
datexAccept-Packet-nbr
AcceptType
Publication
datexPublish-Guaranteed-bool
0
0
Subscription
datexSubscribe-Serial-nbr
...
AlternateRequest
Publish-Format
datexPublish-FileName-txt
*
SubscriptionType
datexSubscribe-CancelReason-cd
PublicationData
datexPublish-SubscribeSerial-nbr
datexPublish-Serial-nbr
datexPublish-LatePublicationFlag-bool
0
SubscriptionData
datexSubscribe-Persistent-bool
datexSubscribe-Status-cd
datexSubscribe-PublishFormat-cd
datexSubscribe-Priority-cd
datexSubscribe-Guarantee-bool
PublicationType
datexPublish-Management-cd
SubscriptionMode
EndApplicationMessage
ISO14827-MESSAGE.&id
ISO14827-MESSAGE.&MessageBody

Figure B.2 — Structure of PDUs

Annex C
(normative)

Data dictionary

C.1 General

The data elements defined within this section are defined using the following ASN.1 Information Object Specification:

```
DatexDataDictionary ::= BEGIN

DATA-ELEMENT ::= CLASS {

    &name                   UTF8String (SIZE (0..255)),

    &namecontext            UTF8String (SIZE (0..40)),

    &definition             UTF8String (SIZE (0..65535)),

    &class                  UTF8String (SIZE (0..40)),

    &classScheme            UTF8String (SIZE (0..40)),

    &classSchemeVer         UTF8String (SIZE (0..40)),

    &keyword                UTF8String (SIZE (0..255)) OPTIONAL,

    &remarks                UTF8String (SIZE (0..2000)) OPTIONAL,

    &valueDomain            UTF8String (SIZE (0..255)),

    &DataType               ,

    &valueRule              UTF8String (SIZE (0..255)),

    &constraints            UTF8String (SIZE (0..2000)) OPTIONAL

}


WITH SYNTAX {

    DESCRIPTIVE-NAME        &name

    CONTEXT                 &namecontext

    DEFINITION              &definition

    CLASS                   &class

    CLASS SCHEME            &classScheme

    SCHEME VERSION          &classSchemeVer

    [KEYWORDS               &keyword]

    [REMARKS                &remarks]

    VALUE DOMAIN            &valueDomain

    DATA TYPE               &DataType
    VALUE RULE              &validValueRule

    [CONSTRAINTS            &constraints]

}
```

The included fields are defined to conform with those as specified in IEEE 1489-1999. Some fields required by IEEE 1489-1999 are not included in this document due to their redundancy.

C.2 Data element structure

C.2.1 AMOUNT_Currency_code-datex1

```
amount-Currency-cd DATA-ELEMENT ::= {

DESCRIPTIVE-NAME   "AMOUNT_Currency_code-datex1"

CONTEXT            "ITS"

DEFINITION         "The three letter currency code associated with the quantity."

CLASS              "Communication Networks"

CLASS SCHEME       "ITS Classification Scheme"

SCHEME VERSION     "980201"

VALUE DOMAIN       "Code-Currency"

DATA TYPE          OCTET STRING (SIZE (3))

VALUE RULE         "ISO 4217"

}
```

C.2.2 AMOUNT_Factor_quantity

```
amount-Factor-qty DATA-ELEMENT ::= {

DESCRIPTIVE-NAME   "AMOUNT_Factor_quantity"

CONTEXT            "ITS"

DEFINITION         "The 10^x factor applied to the value given in AMOUNT_Quantity-quantity.
                   For example, if the Currency is USD, the Factor is -3, and the Quantity
                   is 11; the amount being specified would be US$0.011 or 1.1 US cents."

CLASS              "Communication Networks"

CLASS SCHEME       "ITS Classification Scheme"

SCHEME VERSION     "980201"

VALUE DOMAIN       "Qty-unlimited"

DATA TYPE          INTEGER

VALUE RULE         “INTEGER”

}
```

C.2.3 AMOUNT_Quantity_quantity

```
amount-Quantity-qty DATA-ELEMENT ::= {

DESCRIPTIVE-NAME   “AMOUNT_Quantity_quantity”
CONTEXT             “ITS”
DEFINITION          “The quantity of units as specified by currency and factor.”
CLASS               “Communication Networks”
CLASS SCHEME        “ITS Classification Scheme”
SCHEME VERSION      “980201”
VALUE DOMAIN        “Qty-unlimited”
DATA TYPE           INTEGER
VALUE RULE          “INTEGER”
}
```

C.2.4 DATEX.ACCEPT_Login_id-oid

```
datexAccept-Login-id DATA-ELEMENT ::= {
DESCRIPTIVE-NAME    “DATEX.ACCEPT_Login_id-oid”
CONTEXT             “ITS”
DEFINITION          “Indicates the OID of the encoding rules being accepted for the session.”
CLASS               “Communication Networks”
CLASS SCHEME        “ITS Classification Scheme”
SCHEME VERSION      “980201”
VALUE DOMAIN        “Id-Object Identifier”
DATA TYPE           OBJECT IDENTIFIER
RULE                “OBJECT IDENTIFIER”
}
```

C.2.5 DATEX.ACCEPT_Packet_number-ulong

```
datexAccept-Packet-nbr DATA-ELEMENT ::= {
DESCRIPTIVE-NAME    “DATEX.ACCEPT_Packet_number-ulong”
CONTEXT             “ITS”
DEFINITION          “The DATEX.MESSAGE_DataPacket_number-ulong, as defined in C.2.19, of the
                    request which is being accepted.”
CLASS               “Communication Networks”
CLASS SCHEME        “ITS Classification Scheme”
SCHEME VERSION      “980201”
VALUE DOMAIN        “Number-ULONG”
DATA TYPE           INTEGER (0..4294967295)
VALUE RULE          “INTEGER (0..4294967295)”
}
```

C.2.6 DATEX.ACCEPT_Registered_number-ulong

```
datexAccept-Registered-nbr DATA-ELEMENT ::= {

DESCRIPTIVE-NAME   “DATEX.ACCEPT_Registered_number-ulong”

CONTEXT            “ITS”

DEFINITION         “Indicates the accepted value for the datexRegistered-UpdateDelay-qty
                   Parameter from the associated request.”

CLASS              “Communication Networks”

CLASS SCHEME       “ITS Classification Scheme”

SCHEME VERSION     “980201”

VALUE DOMAIN       “Number-ULONG”

DATA TYPE          INTEGER (0..4294967295)

VALUE RULE         “INTEGER (0..4294967295)”

}
```

C.2.7 DATEX.FrED_ConfirmPacket_number-ulong

```
datexFrED-ConfirmPacket-nbr DATA-ELEMENT ::= {

DESCRIPTIVE-NAME   “DATEX.FrED_ConfirmPacket_number-ulong”

CONTEXT            “ITS”

DEFINITION         “The datagram number being confirmed by the 'friendly exchange of
                   data'datagram. If the FrED is being used as a heartbeat, the value of this
                   field shall be zero (0).”

CLASS              “Communication Networks”

CLASS SCHEME       “ITS Classification Scheme”

SCHEME VERSION     “980201”

VALUE DOMAIN       “Number-ULONG”

DATA TYPE          INTEGER (0..4294967295)

VALUE RULE         “INTEGER (0..4294967295)”

}
```

C.2.8 DATEX.LOGIN_DatagramSize_quantity-ushort

```
datexLogin-DatagramSize-qty DATA-ELEMENT ::= {

DESCRIPTIVE-NAME   “DATEX.LOGIN_DatagramSize_quantity-ushort”

CONTEXT            “ITS”

DEFINITION         “The maximum datagram size that will be supported during the session.”

CLASS              “Communication Networks”

CLASS SCHEME       “ITS Classification Scheme”

SCHEME VERSION     “980201”

VALUE DOMAIN       “Qty-USHORT”
DATA TYPE          INTEGER (0..65535)

VALUE RULE         “INTEGER (0..65535)”

}
```

C.2.9 DATEX.LOGIN_EncodingRules_id-oids

```
datexLogin-EncodingRules-id DATA-ELEMENT ::= {

DESCRIPTIVE-NAME   “DATEX.LOGIN_EncodingRules_id-oids”

CONTEXT            “ITS”

DEFINITION         “A listing of the encoding rules that the Client supports.”

CLASS              “Communication Networks”

CLASS SCHEME       “ITS Classification Scheme”

SCHEME VERSION     “980201”

VALUE DOMAIN       “Id-Object Identifiers”

DATA TYPE          SEQUENCE OF OBJECT IDENTIFIER

VALUE RULE         “SEQUENCE OF OBJECT IDENTIFIER”

}
```

C.2.10 DATEX.LOGIN_HeartbeatDurationMax_quantity-ushort

```
datexLogin-HeartbeatDurationMax-qty DATA-ELEMENT ::= {

DESCRIPTIVE-NAME   “DATEX.LOGIN_HeartbeatDurationMax_quantity-ushort”

CONTEXT            “ITS”

DEFINITION         “The maximum duration, in seconds, allowed during the session without the
                   exchange of datagrams. If this time is exceeded without any datagrams being
                   received from the other system, the session shall be locally terminated
                   without exchanging any datagrams. The value zero shall indicate that
                   heartbeats are not being used.”

CLASS              “Communication Networks”

CLASS SCHEME       “ITS Classification Scheme”

SCHEME VERSION     “980201”

VALUE DOMAIN       “Qty-USHORT”

DATA TYPE          INTEGER (0..65535)

VALUE RULE         “INTEGER (0..65535)”

}
```

C.2.11 DATEX.LOGIN_Initiator_code-datex2

```
datexLogin-Initiator-cd DATA-ELEMENT ::= {

DESCRIPTIVE-NAME   “DATEX.LOGIN_Initiator_code-datex2”

CONTEXT            “ITS”

DEFINITION         “Indicates who initiated the Login request.”
CLASS              “Communication Networks”

CLASS SCHEME       “ITS Classification Scheme”

SCHEME VERSION     “980201”

VALUE DOMAIN       “Code-DATEX Initiator”

DATA TYPE          ENUMERATED {serverInitiated, clientInitiated, …}

VALUE RULE         “ENUMERATED {serverInitiated, clientInitiated, …}”

}
```

C.2.12 DATEX.LOGIN_Password_text-general

```
datexLogin-Password-txt DATA-ELEMENT ::= {

DESCRIPTIVE-NAME   “DATEX.LOGIN_Password_text-general”

CONTEXT            “ITS”

DEFINITION         “An optionally encrypted password for the login request. The password and
                   username shall be verified by the end application to ensure that the user
                   has appropriate rights. The encryption algorithm is not specified in this
                   document.”

CLASS              “Communication Networks”

CLASS SCHEME       “ITS Classification Scheme”

SCHEME VERSION     “980201”

VALUE DOMAIN       “Text-OctetString Unlimited”

DATA TYPE          OCTET STRING

VALUE RULE         “OCTET STRING”

}
```

C.2.13 DATEX.LOGIN_ResponseTimeOut_quantity-ubyte

```
datexLogin-ResponseTimeOut-qty DATA-ELEMENT ::= {

DESCRIPTIVE-NAME   “DATEX.LOGIN_ResponseTimeOut_quantity-ubyte”

CONTEXT            “ITS”

DEFINITION         “The time in seconds within which a system will expect to receive a
                   response to the transmission of a datagram. This same timer will apply to
                   both the Client and the Supplier (Server) and will apply to all datagrams
                   transmitted within the session where a response is required. The time is
                   measured from the return from the system write () call. If a system has
                   not received the appropriate response prior to this timer expiring, it
                   will assume the transmitted datagram was not received by the other end
                   and proceed as defined in 7.2.5”

CLASS              “Communication Networks”

CLASS SCHEME       “ITS Classification Scheme”

SCHEME VERSION     “980201”

VALUE DOMAIN       “Qty-UBYTE”

DATA TYPE          INTEGER (0..255)

VALUE RULE         “INTEGER (0..255)”
CONSTRAINT          "The value zero (0) is not allowed."

}
```

C.2.14 DATEX.LOGIN_UserName_text-general

```
datexLogin-UserName-txt DATA-ELEMENT ::= {

DESCRIPTIVE-NAME    "DATEX.LOGIN_UserName_text-general"

CONTEXT             "ITS"

DEFINITION          "An optionally encrypted username for the Login request. The encryption
                    algorithm is not specified in this document."

CLASS               "Communication Networks"

CLASS SCHEME        "ITS Classification Scheme"

SCHEME VERSION      "980201"

VALUE DOMAIN        "Text-OctetString Unlimited"

DATA TYPE           OCTET STRING

VALUE RULE          "OCTET STRING"

}
```

C.2.15 DATEX.LOGOUT_Reason_code-datex14

```
datexLogout-Reason-cd DATA-ELEMENT ::= {

DESCRIPTIVE-NAME    "DATEX.LOGOUT_Reason_code-datex14"

CONTEXT             "ITS"

DEFINITION          "The reason the logout is occurring."

CLASS               "Communication Networks"

CLASS SCHEME        "ITS Classification Scheme"

SCHEME VERSION      "980201"

VALUE DOMAIN        "Code-DATEX Terminate"

DATA TYPE           ENUMERATED {other, serverRequested, clientRequested, serverShutdown,
                    clientShutdown, serverCommProblems, clientCommProblems, …}

VALUE RULE          "ENUMERATED {other, serverRequested, clientRequested, serverShutdown,
                    clientShutdown, serverCommProblems, clientCommProblems, …}"

}
```

C.2.16 DATEX.MESSAGE_AuthenticationInformation_text-general255

```
datex-AuthenticationInfo-txt DATA-ELEMENT ::= {

DESCRIPTIVE-NAME    "DATEX.MESSAGE_AuthenticationInformation_text-general255"

CONTEXT             "ITS"
DEFINITION          "Authentication information which the two interconnected systems have
                    agreed to include within each message. This can be used to exchange a
                    fixed code or algorithmic code in order to provide additional password
                    protection. This may be zero length octet string, a fixed octet string,
                    or a variable octet string according to rules defined in the Interchange
                    Agreement (e.g. the parameters defined in Annex E)."

CLASS               "Communication Networks"

CLASS SCHEME        "ITS Classification Scheme"

SCHEME VERSION      "980201"

KEY WORD            "Authentication, DATEX"

VALUE DOMAIN        "Text-OctetString 255"

DATA TYPE           OCTET STRING (SIZE (0..255))

VALUE RULE          "OCTET STRING (SIZE (0..255))"

}
```

C.2.17 DATEX.MESSAGE_Crc_id-crc16

```
datex-Crc-id DATA-ELEMENT ::= {

DESCRIPTIVE-NAME    "DATEX.MESSAGE_Crc_id-crc16"

CONTEXT             "ITS"

DEFINITION          "The relatively unique code for the message byte stream which is used for
                    error checking. The value of the code is determined by applying the CRC16
                    algorithm defined in ISO 3309 on the byte stream encoding of datex-Data-
                    txt. For example, if BER encoding was used, the CRC would be calculated
                    on the identifier, length, and contents octets of the encoding."

CLASS               "Communication Networks"

CLASS SCHEME        "ITS Classification Scheme"

SCHEME VERSION      "980201"

KEYWORDS            "CRC, DATEX"

VALUE DOMAIN        "Id-Crc16"

DATA TYPE           OCTET STRING (SIZE (2))

VALUE RULE          "OCTET STRING (SIZE (2))"

}
```

C.2.18 DATEX.MESSAGE_Data_text-general

```
datex-Data-txt DATA-ELEMENT ::= {

DESCRIPTIVE-NAME    "DATEX.MESSAGE_Data_text-general"

CONTEXT             "ITS"

DEFINITION          "The optionally encrypted contents of the DATEX Data Packet as defined by
                    the structure C2CAuthenticatedMessage."

CLASS               "Communication Networks"

CLASS SCHEME        "ITS Classification Scheme"

SCHEME VERSION      "980201"
VALUE DOMAIN        "Text-OctetString Unlimited"

DATA TYPE           OCTET STRING

VALUE RULE          "OCTET STRING"

CONSTRAINTS         "The initial layout for this octet string is given by C2CAuthenticatedMessage;
                    however, this data may be encrypted prior to transmission over the
                    communications media."

}
```

C.2.19 DATEX.MESSAGE_DataPacket_number-ulong

```
datex-DataPacket-nbr DATA-ELEMENT ::= {

DESCRIPTIVE-NAME    "DATEX.MESSAGE_DataPacket_number-ulong"

CONTEXT             "ITS"

DEFINITION          "Indicates the datagram number for the datagram being sent. The first
                    datagram sent during session establishment shall have a number of zero.
                    Each subsequent datagram shall have its number incremented by one. The
                    client and supplier (server) shall maintain separate counters such that
                    the first datagram sent by the Supplier (server) and the first datagram
                    sent by the client shall both be zero."

CLASS               "Communication Networks"

CLASS SCHEME        "ITS Classification Scheme"

SCHEME VERSION      "980201"

VALUE DOMAIN        "Number-ULONG"

DATA TYPE           INTEGER (0..4294967295)

VALUE RULE          "INTEGER (0..4294967295)"

}
```

C.2.20 DATEX.MESSAGE_DataPacketPriority_code-datex11

```
datex-DataPacketPriority-cd DATA-ELEMENT ::= {

DESCRIPTIVE-NAME    "DATEX.MESSAGE_DataPacketPriority_code-datex11"

CONTEXT             "ITS"

DEFINITION          "The priority of the message. A system must process messages in the order
                    in which they are received, unless a later message has a higher priority.
                    Messages of higher priority may be processed before messages of lower
                    priority at the option of the implementation."

CLASS               "Communication Networks"

CLASS SCHEME        "ITS Classification Scheme"

SCHEME VERSION      "980201"

VALUE DOMAIN        "Code-DATEX Priority"

DATA TYPE           INTEGER (1..10)

VALUE RULE          "INTEGER (1..10)"

}
```

C.2.21 DATEX.MESSAGE_DataPacketTime_frame

```
datex-DataPacketTime DATA-ELEMENT ::= {
DESCRIPTIVE-NAME   “DATEX.MESSAGE_DataPacketTime_frame”
CONTEXT            “ITS”
DEFINITION         “The time at which the datagram is being generated.”
CLASS              “Communication Networks”
CLASS SCHEME       “ITS Classification Scheme”
SCHEME VERSION     “980201”
VALUE DOMAIN       “Time data structure per B.2.8”
DATA TYPE          Time
VALUE RULE         “Time data structure per B.2.8”
}
```

C.2.22 DATEX.MESSAGE_Destination_text-name

```
datex-Destination-txt DATA-ELEMENT ::= {
DESCRIPTIVE-NAME   “DATEX.MESSAGE_Destination_text-name”
CONTEXT            “ITS”
DEFINITION         “The domain name of the system which is supposed to receive the message.”
CLASS              “Communication Networks”
CLASS SCHEME       “ITS Classification Scheme”
SCHEME VERSION     “980201”
KEYWORDS           “DATEX, Destination”
VALUE DOMAIN       “Text-Name”
DATA TYPE          UTF8String (SIZE (0..40))
VALUE RULE         “UTF8String (SIZE (0..40))”
}
```

C.2.23 DATEX.MESSAGE_DestinationAddress_location-address

```
datex-DestinationAddress-loc DATA-ELEMEN ::= {
DESCRIPTIVE-NAME   “DATEX.MESSAGE_DestinationAddress_location-address”
CONTEXT            “ITS”
DEFINITION         “A unique address for the computer which is the intended recipient of the
                   message.”
CLASS              “Communication Networks”
CLASS SCHEME       “ITS Classification Scheme”
SCHEME VERSION     “980201”
VALUE DOMAIN       “Location-Address”
DATA TYPE           OCTET STRING

VALUE RULE          “AddressCode ::= CHOICE{
                      gis                   IMPLICIT [1] NumericString,
                      mhORName              IMPLICIT [2] MhORName,
                      --X.400
                      dn                    IMPLICIT [3] DistinguishedName,
                      -- X.500
                      isdnOrPhonenumber     IMPLICIT [4] E164Form,
                      rfc822Address         IMPLICIT [5] PrintableString,
                      pstnAddress           IMPLICIT [6] NumericString
                    }”
}
```

C.2.24 DATEX.MESSAGE_Origin_text-name

```
datex-Origin-txt DATA-ELEMENT ::= {

DESCRIPTIVE-NAME    “DATEX.MESSAGE_Origin_text-name”

CONTEXT             “ITS”

DEFINITION          “The domain name of the system that collected the data contained in the
                    end application message.”

CLASS               “Communication Networks”

CLASS SCHEME        “ITS Classification Scheme”

SCHEME VERSION      “980201”

VALUE DOMAIN        “Text-Name”

DATA TYPE           UTF8String (SIZE (0..40))

VALUE RULE          “UTF8String (SIZE (0..40))”

}
```

C.2.25 DATEX.MESSAGE_OriginAddress_location-address

```
datex-OriginAddress-loc DATA-ELEMENT ::= {

DESCRIPTIVE-NAME    “DATEX.MESSAGE_OriginAddress_location-address”

CONTEXT             “ITS”

DEFINITION          “A unique address of the system that collected the data contained in the
                    end-application.”

CLASS               “Communication Networks”

CLASS SCHEME        “ITS Classification Scheme”

SCHEME VERSION      “980201”

VALUE DOMAIN        “Location-Address”

DATA TYPE           OCTET STRING
VALUE RULE          “AddressCode ::= CHOICE{
                      gis                   IMPLICIT [1] NumericString,
                      mhORName              IMPLICIT [2] MhORName,
                      --X.400
                      dn                    IMPLICIT [3] DistinguishedName,
                      -- X.500
                      isdnOrPhonenumber     IMPLICIT [4] E164Form,
                      rfc822Address         IMPLICIT [5] PrintableString,
                      pstnAddress           IMPLICIT [6] NumericString
                    }”
}
```

C.2.26 DATEX.MESSAGE_Sender_text-name

```
datex-Sender-txt DATA-ELEMENT ::= {
DESCRIPTIVE-NAME    “DATEX.MESSAGE_Sender_text-name”
CONTEXT             “ITS”
DEFINITION          “The domain name of the system which is sending the message.”
CLASS               “Communication Networks”
CLASS SCHEME        “ITS Classification Scheme”
SCHEME VERSION      “980201”
VALUE DOMAIN        “Text-Name”
DATA TYPE           UTF8String (SIZE (0..40))
VALUE RULE          “UTF8String (SIZE (0..40))”
}
```

C.2.27 DATEX.MESSAGE_SenderAddress_location-address

```
datex-SenderAddress-loc DATA-ELEMENT ::= {
DESCRIPTIVE-NAME    “DATEX.MESSAGE_SenderAddress_location-address”
CONTEXT             “ITS”
DEFINITION          “A unique address for the computer which is sending the message.”
CLASS               “Communication Networks”
CLASS SCHEME        “ITS Classification Scheme”
SCHEME VERSION      “980201”
VALUE DOMAIN        “Location-Address”
DATA TYPE           OCTET STRING
VALUE RULE          "AddressCode ::= CHOICE{
                      gis                   IMPLICIT [1] NumericString,
                      mhORName              IMPLICIT [2] MhORName,
                      --X.400
                      dn                    IMPLICIT [3] DistinguishedName,
                      -- X.500
                      isdnOrPhonenumber     IMPLICIT [4] E164Form,
                      rfc822Address         IMPLICIT [5] PrintableString,
                      pstnAddress           IMPLICIT [6] NumericString
                    }"
}
```

C.2.28 DATEX.MESSAGE_Version_code-datex15

```
datex-Version-cd DATA-ELEMENT ::= {
DESCRIPTIVE-NAME    "DATEX.MESSAGE_Version_code-datex15"
CONTEXT             "ITS"
DEFINITION          "This is the DATEX-ASN version to which this message conforms"
CLASS               "Communication Networks"
CLASS SCHEME        "ITS Classification Scheme"
SCHEME VERSION      "980201"
VALUE DOMAIN        "Code-DATEX Version"
DATA TYPE           ENUMERATED {experimental, version-1, …}
VALUE RULE          "ENUMERATED {experimental, version-1, …}"
}
```

C.2.29 DATEX.PUBLISH_FileName_text-memo

```
datexPublish-FileName-txt DATA-ELEMENT ::= {
DESCRIPTIVE-NAME    "DATEX.PUBLISH_FileName_text-memo"
CONTEXT             "ITS"
DEFINITION          "The name of the file, including the path, which contains the SEQUENCE OF
                    PublicationData that the supplier (server) is attempting to publish. It
                    is recommended that filenames be sequentially numbered to ensure a high
                    degree of uniqueness."
CLASS               "Communication Networks"
CLASS SCHEME        "ITS Classification Scheme"
SCHEME VERSION      "980201"
VALUE DOMAIN        "Text-Memo"
DATA TYPE           UTF8String (SIZE (0..2000))
VALUE RULE          "UTF8String (SIZE (0..2000))"
}
```

C.2.30 DATEX.PUBLISH_Guaranteed_boolean

```
datexPublish-Guaranteed-bool DATA-ELEMENT ::= {

DESCRIPTIVE-NAME   "DATEX.PUBLISH_Guaranteed_boolean"

CONTEXT            "ITS"

DEFINITION         "Indicates whether or not the client receiving the publication (reply) is
                   required to acknowledge receipt of the datagram by issuing an 'Accept'
                   datagram. This shall be set to 'true' if DATEX.SUBSCRIPTION_Guarantee_bool
                   was set to 'true' in the associated subscription (request)."

CLASS              "Communication Networks"

CLASS SCHEME       "ITS Classification Scheme"

SCHEME VERSION     "980201"

VALUE DOMAIN       "Boolean"

DATA TYPE          BOOLEAN

VALUE RULE         "BOOLEAN"

}
```

C.2.31 DATEX.PUBLISH_LatePublicationFlag_boolean

```
datexPublish-LatePublicationFlag-bool DATA-ELEMENT ::= {

DESCRIPTIVE-NAME   "DATEX.PUBLISH_LatePublicationFlag_boolean"

CONTEXT            "ITS"

DEFINITION         "Indicates whether the publication is being sent after the expiration of
                   the UpdateDelay timer."

CLASS              "Communication Networks"

CLASS SCHEME       "ITS Classification Scheme"

SCHEME VERSION     "980201"

VALUE DOMAIN       "Boolean"

DATA TYPE          BOOLEAN

VALUE RULE         "BOOLEAN"

}
```

C.2.32 DATEX.PUBLISH_Management_code-datex3

```
datexPublish_Management_cd DATA-ELEMENT ::= {

DESCRIPTIVE-NAME   "DATEX.PUBLISH_Management_code-datex3"

CONTEXT            "ITS"

DEFINITION         "Indicates the status of the publication feature per one of the defined
                   enumerated values."

CLASS              "Communication Networks"

CLASS SCHEME       "ITS Classification Scheme"
SCHEME VERSION     "980201"

VALUE DOMAIN       "Code-DATEX Publication Type"

DATA TYPE          ENUMERATED { temporarilySuspended, resume, terminate-other, terminate-
                   dataNoLongerAvailable, terminate-publicationsBeingRejected, terminate-
                   PendingShutdown, terminate-processingMgmt, terminate-bandwidthMgmt,
                   terminate-accessDenied, unknownRequest, . . .}

VALUE RULE         "ENUMERATED {temporarilySuspended, resume, terminate-other, terminate-
                   dataNoLongerAvailable, terminate-publicationsBeingRejected, terminate-
                   PendingShutdown, terminate-processingMgmt, terminate-bandwidthMgmt,
                   terminate-accessDenied, unknownRequest, . . .}"

}
```

C.2.33 DATEX.PUBLISH_Serial_number-ulong

```
datexPublish-Serial-nbr DATA-ELEMENT ::= {

DESCRIPTIVE-NAME   "DATEX.PUBLISH_Serial_number-ulong"

CONTEXT            "ITS"

DEFINITION         "Indicates how many responses have been published to the associated
                   request, including this publication (reply). The first publication to a
                   given response shall receive a Serial of 1 and each subsequent publication
                   to the same subscription (request) shall have a Serial number which is
                   incremented by one."

CLASS              "Communication Networks"

CLASS SCHEME       "ITS Classification Scheme"

SCHEME VERSION     "980201"

VALUE DOMAIN       "Number-ULONG"

DATA TYPE          INTEGER (0..4294967295)

VALUE RULE         "INTEGER (0..4294967295)"

CONSTRAINTS        "The value 0 is reserved."

}
```

C.2.34 DATEX.PUBLISH_SubscribeSerial_number-ulong

```
datexPublish-SubscribeSerial-nbr DATA-ELEMENT ::= {

DESCRIPTIVE-NAME   "DATEX.PUBLISH_SubscribeSerial_number-ulong"

CONTEXT            "ITS"

DEFINITION         "The serial number of the subscription (request) which caused the generation
                   of the publication (reply)."

CLASS              "Communication Networks"

CLASS SCHEME       "ITS Classification Scheme"

SCHEME VERSION     "980201"

VALUE DOMAIN       "Number-ULONG"

DATA TYPE          INTEGER (0..4294967295)

VALUE RULE         "INTEGER (0..4294967295)"
CONSTRAINTS        "The value of zero (0) shall be reserved for unrequested emergency
                   publications as agreed to by the centres."

}
```

C.2.35 DATEX.REGISTERED_DaysOfWeek_code-DaysOfWeek

```
datexRegistered-DaysOfWeek-cd DATA-ELEMENT ::= {

DESCRIPTIVE-NAME   "DATEX.REGISTERED_DaysOfWeek_code-DaysOfWeek"

CONTEXT            "ITS"

DEFINITION         "The days of week on which the subscription (request) is activated on the
                   supplier (server). "

CLASS              "Communication Networks"

CLASS SCHEME       "ITS Classification Scheme"

SCHEME VERSION     "980201"

VALUE DOMAIN       "Code-Days Of Week"

DATA TYPE          BIT STRING {other, sunday, monday, tuesday, wednesday, thursday, friday,
                   saturday} (SIZE (8))

VALUE RULE         "BIT STRING {other, sunday, monday, tuesday, wednesday, thursday, friday,
                   saturday} (SIZE (8))"

}
```

C.2.36 DATEX.REGISTERED_Duration_quantity-ushort

```
datexRegistered-Duration-qty DATA-ELEMENT ::= {

DESCRIPTIVE-NAME   "DATEX.REGISTERED_Duration_quantity-ushort"

CONTEXT            "ITS"

DEFINITION         "The duration, in minutes, for which the subscription (request) remains
                   active after the Start Time."

CLASS              "Communication Networks"

CLASS SCHEME       "ITS Classification Scheme"

SCHEME VERSION     "980201"

VALUE DOMAIN       "Qty-USHORT"

DATA TYPE          INTEGER (0..65535)

VALUE RULE         "INTEGER (0..65535)"

}
```

C.2.37 DATEX.REGISTERED_EndDate_frame

```
datexRegistered-EndDate DATA-ELEMENT ::= {

DESCRIPTIVE-NAME   "DATEX.REGISTERED_EndDate_frame"

CONTEXT            "ITS"

DEFINITION         "The last date on which the subscription (request) may be activated."

CLASS              "Communication Networks"
CLASS SCHEME        “ITS Classification Scheme”

SCHEME VERSION      “980201”

VALUE DOMAIN        “Time data structure per B.2.8”

DATA TYPE           Time

VALUE RULE          “Time data structure per B.2.8”

}
```

C.2.38 DATEX.REGISTERED_EndTime_frame

```
datexRegistered-EndTime DATA-ELEMENT ::= {

DESCRIPTIVE-NAME    “DATEX.REGISTERED_EndTime_frame”

CONTEXT             “ITS”

DEFINITION          “The time of day at which the subscription (request) is deactivated.”

CLASS               “Communication Networks”

CLASS SCHEME        “ITS Classification Scheme”

SCHEME VERSION      “980201”

VALUE DOMAIN        “Time data structure per B.2.8”

DATA TYPE           Time

VALUE RULE          “Time data structure per B.2.8”

}
```

C.2.39 DATEX.REGISTERED_StartDate_frame

```
datexRegistered-StartDate DATA-ELEMENT ::= {

DESCRIPTIVE-NAME    “DATEX.REGISTERED_StartDate_frame”

CONTEXT             “ITS”

DEFINITION          “The first date on which the subscription (request) may be activated.”

CLASS               “Communication Networks”

CLASS SCHEME        “ITS Classification Scheme”

SCHEME VERSION      “980201”

VALUE DOMAIN        “Time data structure per B.2.8”

DATA TYPE           Time

VALUE RULE          “Time data structure per B.2.8”

}
```

C.2.40 DATEX.REGISTERED_StartTime_frame

```
datexRegistered-StartTime DATA-ELEMENT ::= {

DESCRIPTIVE-NAME    “DATEX.REGISTERED_StartTime_frame”

CONTEXT             “ITS”
DEFINITION         "The time of day at which the subscription (request) is activated, on those
                   days which are valid according to the StartDate, EndDate and DaysOfWeek
                   fields. If the date on which this subscription is received is a valid day
                   and the start time has already passed but the stop time has not passed,
                   the subscription shall immediately be activated. The value 00:00 shall
                   mean the start of the day."

CLASS              "Communication Networks"

CLASS SCHEME       "ITS Classification Scheme"

SCHEME VERSION     "980201"

VALUE DOMAIN       "Time data structure per B.2.8"

DATA TYPE          Time

VALUE RULE         "Time data structure per B.2.8"

}
```

C.2.41 DATEX.REGISTERED_UpdateDelay_quantity-ulong

```
datexRegistered-UpdateDelay-qty DATA-ELEMENT ::= {

DESCRIPTIVE-NAME   "DATEX.REGISTERED_UpdateDelay_quantity-ulong"

CONTEXT            "ITS"

DEFINITION         "The update interval for the subscription (request). If the mode field of
                   the SubscriptionData structure is 'periodic', a new publication (reply)
                   shall be produced every DATEX.REGISTERED_UpdateDelay_quantity-ulong seconds.
                   If the mode field of the SubscriptionData structure is 'event-driven',
                   a new publication datagram shall be generated within DATEX.REGISTERED_
                   UpdateDelay_quantity seconds from the time that a new record or record
                   update occurs, respectively, as defined by the message referenced in the
                   message field of the same SubscriptionData structure; in this case, the
                   value denotes a maximum latency."

CLASS              "Communication Networks"

CLASS SCHEME       "ITS Classification Scheme"

SCHEME VERSION     "980201"

VALUE DOMAIN       "Qty-ULONG"

DATA TYPE          INTEGER (0..4294967295)

VALUE RULE         "INTEGER (0..4294967295)"

}
```

C.2.42 DATEX.REJECT_Login_code-datex6

```
datexReject-Login-cd DATA-ELEMENT ::= {

DESCRIPTIVE-NAME   "DATEX.REJECT_Login_code-datex6 "

CONTEXT            "ITS"

DEFINITION         "The reason that the Login packet is being rejected."

CLASS              "Communication Networks"

CLASS SCHEME       "ITS Classification Scheme"

SCHEME VERSION     "980201"
VALUE DOMAIN        "Code-DATEX Reject Login"

DATA TYPE           ENUMERATED {other, unknownDomainName, accessDenied, invalidNamePassword,
                    timeoutTooSmall, timeoutTooLarge, heartbeatTooSmall, heartbeatTooLarge,
                    sessionExists, maxSessionsReached, …}

VALUE RULE          "other - used for any other reason

                    unknownDomainName - the Client or Supplier (Server) domain name in the
                    Login was unknown or invalid

                    accessDenied - the Supplier (Server) is denying access for some reason

                    invalidNamePassword – the Supplier (server) is denying access due to an
                    invalid name password pair.

                    timeoutTooSmall (TooLarge) - the timeout value in the Login was not within
                    a range that the Supplier (server) supports

                    heartbeatTooSmall (TooLarge) - the heartbeat in the Login was not within
                    a range that the Supplier (server) supports

                    sessionExists - a session already exists between the indicated Client
                    domain name and Supplier (server) domain name over the specified transport
                    profile; only one session is allowed between the pair over the same profile.

                    maxSessionsReached - the Supplier (server) can not support any more
                    sessions."

}
```

C.2.43 DATEX.REJECT_Packet_number-ulong

```
datexReject-Packet-nbr DATA-ELEMENT ::= {

DESCRIPTIVE-NAME    "DATEX.REJECT_Packet_number-ulong"

CONTEXT             "ITS"

DEFINITION          "The DATEX.MESSAGE_DataPacket_number-ulong, as defined in B.2.1, of the
                    request which is being rejected. For the rejection of specific PublicationData
                    structures within a Publication datagram, this number shall be set to zero."

CLASS               "Communication Networks"

CLASS SCHEME        "ITS Classification Scheme"

SCHEME VERSION      "980201"

VALUE DOMAIN        "Number-ULONG"

DATA TYPE           INTEGER (0..4294967295)

VALUE RULE          "INTEGER (0..4294967295)"

}
```

C.2.44 DATEX.REJECT_Publication_code-datex7

```
datexReject-Publication-cd DATA-ELEMENT ::= {

DESCRIPTIVE-NAME    "DATEX.REJECT_Publication_code-datex7"

CONTEXT             "ITS"

DEFINITION          "A reason why the Publication (reply) datagram was rejected."

CLASS               "Communication Networks"

CLASS SCHEME        "ITS Classification Scheme"
SCHEME VERSION      “980201”

VALUE DOMAIN        “Code-DATEX Reject Publication”

DATA TYPE           ENUMERATED {other, invalidPublishFormat, …}

VALUE RULE          “other - used for any other error
                    invalidPublishFormat – used if the selected publication format is invalid”
}
```

C.2.45 DATEX.REJECT_PublicationData_code-datex16

```
datexReject-PublicationData-cd DATA-ELEMENT ::= {

DESCRIPTIVE-NAME    “DATEX.REJECT_PublicationData_code-datex16”

CONTEXT             “ITS”

DEFINITION          “A reason why the PublicationData structure was rejected.”

CLASS               “Communication Networks”

CLASS SCHEME        “ITS Classification Scheme”

SCHEME VERSION      “980201”

VALUE DOMAIN        “Code-DATEX Reject Publication Data”

DATA TYPE           ENUMERATED {other, unknownSubscription, unknownPublicationNbr,
                    unknownPublicationMsgId, invalidPublicationMsgId, invalidPublicationMsgContent,
                    repeatedPublicationNbr, …}

VALUE RULE          “other - used for any other error
                    unknownSubscription - used if the subscription (request) is not recognized
                    unknownPublicationNbr – used if the publication is not recognized
                    unknownPublicationMsgId – used if the publication message identification
                    number is not recognized
                    invalidPublicationMsgId - used if the publication message identification
                    number is recognized, but is an invalid message identification number
                    invalidPublicationMsgContent - the content of the publication message is
                    recognized, but the content is invalid
                    repeatedPublicationNbr – a publication with this publication number has
                    already been received for this subscription.”
}
```

C.2.46 DATEX.REJECT_PublicationSerial_number-ulong

```
datexReject-PublicationSerial-nbr DATA-ELEMENT ::= {

DESCRIPTIVE-NAME    “DATEX.REJECT_PublicationSerial_number-ulong”

CONTEXT             “ITS”

DEFINITION          “The publication serial number of the PublicationData structure that
                    contained invalid data and is being rejected.”

CLASS               “Communication Networks”

CLASS SCHEME        “ITS Classification Scheme”

SCHEME VERSION      “980201”
VALUE DOMAIN        “Number-ULONG”

DATA TYPE           INTEGER (0..4294967295)

VALUE RULE          “INTEGER (0..4294967295)”

}
```

C.2.47 DATEX.REJECT_Subscription_code-datex8

```
datexReject-Subscription-cd DATA-ELEMENT ::= {

DESCRIPTIVE-NAME    “DATEX.REJECT_Subscription_code-datex8”

CONTEXT             “ITS”

DEFINITION          “A reason why the Subscription (request) was rejected.”

CLASS               “Communication Networks”

CLASS SCHEME        “ITS Classification Scheme”

SCHEME VERSION      “980201”

VALUE DOMAIN        “Code-DATEX Reject Subscription”

DATA TYPE           ENUMERATED {other, unknownSubscriptionNbr, invalidTimes, frequencyTooSmall,
                    frequencyTooLarge, invalidMode, publishFormatNotSupported,
                    unknownSubscriptionMsgId, invalidSubscriptionMsgId, invalidSubscriptionContent,
                    …}

VALUE RULE          “other - used for any other error

                    unknownSubscriptionNbr - used if the subscription (request) is known to
                    be too big to fit in a requested datagram

                    invalidTimes - used if the time of either the Request or event message is
                    not recognized by the Supplier (server)

                    frequencyTooSmall - used if the frequency is too small for one of the
                    fields in the request or event message within the subscription

                    frequencyTooLarge - - used if the frequency is too large for one of the
                    fields in the request or event message within the subscription

                    invalidMode – the subscription contained an invalid mode.

                    publishFormatNotSupported – the publish format requested within the
                    subscription is not supported

                    unknownSubscriptionMsgId – the subscription message identification is
                    unknown

                    invalidSubscriptionMsgId – the subscription message identification is
                    invalid

                    invalidSubscriptionContent – the content of the subscription message is
                    invalid”

}
```

C.2.48 DATEX.REJECT_SubscriptionSerial_number-ulong

```
datexReject-SubscriptionSerial-nbr DATA-ELEMENT ::= {

DESCRIPTIVE-NAME    “DATEX.REJECT_SubscriptionSerial_number-ulong”

CONTEXT             “ITS”
DEFINITION          "The subscription serial number of the PublicationData structure that
                    contained invalid data and is being rejected."

CLASS               "Communication Networks"

CLASS SCHEME        "ITS Classification Scheme"

SCHEME VERSION      "980201"

VALUE DOMAIN        "Number-ULONG"

DATA TYPE           INTEGER (0..4294967295)

VALUE RULE          "INTEGER (0..4294967295)"

}
```

C.2.49 DATEX.SUBSCRIBE_CancelReason_code-datex5

```
datexSubscribe-CancelReason-cd DATA-ELEMENT ::= {

DESCRIPTIVE-NAME    "DATEX.SUBSCRIBE_CancelReason_code-datex5"

CONTEXT             "ITS"

DEFINITION          "The reason that the subscription (request) is being cancelled."

CLASS               "Communication Networks"

CLASS SCHEME        "ITS Classification Scheme"

SCHEME VERSION      "980201"

VALUE DOMAIN        "Code-DATEX Cancel Subscription"

DATA TYPE           ENUMERATED {other, dataNotNeeded, errorsInPublication, pendingLogout,
                    processingMgmt, bandwidthMgmt, ...}

VALUE RULE
                    "other - not one of the standard reasons listed in this document

                    dataNotNeeded - indicates that the Client no longer requires the data

                    errorsInPublication - indicates that the Client is cancelling the
                    subscription (request) due to excessive publications that could not be
                    decoded or contained invalid data.

                    pendingLogout - indicates that the Client is in the process of gracefully
                    terminating the session

                    processingMgmt - indicates that the Client is cancelling the subscription
                    so that it may concentrate its processing resources on other tasks

                    bandwidthMgmt - indicates that the Client is cancelling the subscription
                    so that it may use the limited bandwidth for higher priority needs"

}
```

C.2.50 DATEX.SUBSCRIBE_Guarantee_boolean

```
datexSubscribe-Guarantee-bool DATA-ELEMENT ::= {

DESCRIPTIVE-NAME    "DATEX.SUBSCRIBE_Guarantee_boolean"

CONTEXT             "ITS"

DEFINITION          "Indicates whether or not the publication (reply) should be confirmed.
                    If the value is 'true', the supplier (server) shall set DATEX.PUBLISH_
                    Guaranteed_boolean of the publication datagram(s) to true; otherwise the
                    indicated field shall be set to false."
CLASS               "Communication Networks"

CLASS SCHEME        "ITS Classification Scheme"

SCHEME VERSION      "980201"

VALUE DOMAIN        "Boolean"

DATA TYPE           BOOLEAN

VALUE RULE          "BOOLEAN"

}
```

C.2.51 DATEX.SUBSCRIBE_Persistent_boolean

```
datexSubscribe-Persistent-bool DATA-ELEMENT ::= {

DESCRIPTIVE-NAME    "DATEX.SUBSCRIBE_Persistent_boolean"

CONTEXT             "ITS"

DEFINITION          "Indicates whether or not the subscription (request) will survive session
                    terminations."

CLASS               "Communication Networks"

CLASS SCHEME        "ITS Classification Scheme"

SCHEME VERSION      "980201"

VALUE DOMAIN        "Boolean"

DATA TYPE           BOOLEAN

VALUE RULE          "BOOLEAN"

}
```

C.2.52 DATEX.SUBSCRIBE_Priority_code-datex11

```
datexSubscribe-Priority-cd DATA-ELEMENT ::= {

DESCRIPTIVE-NAME    "DATEX.SUBSCRIBE_Priority_code-datex11"

CONTEXT             "ITS"

DEFINITION          "Indicates the relative priority of the subscription. A priority of 1 is
                    the highest priority and a priority of 10 is the lowest priority."

CLASS               "Communication Networks"

CLASS SCHEME        "ITS Classification Scheme"

SCHEME VERSION      "980201"

VALUE DOMAIN        "Code-DATEX Priority"

DATA TYPE           INTEGER (1..10)

VALUE RULE          "INTEGER (1..10)"

}
```

C.2.53 DATEX.SUBSCRIBE_PublishFormat_code-datex4

```
datexSubscribe-PublishFormat-cd DATA-ELEMENT ::= {
```

```
DESCRIPTIVE-NAME   "DATEX.SUBSCRIBE_PublishFormat_code-datex4"

CONTEXT            "ITS"

DEFINITION         "The format in which the corresponding publication(s) should be exchanged.
                   If the value is 'ftp' or 'tftp', the supplier (server) shall store the
                   publication (reply) in a file and transmit the filename in the 'publication'
                   datagram. If the value is 'dataPacket' the supplier (server) shall attempt
                   to transmit the publication data within the 'publication' datagram."

CLASS              "Communication Networks"

CLASS SCHEME       "ITS Classification Scheme"

SCHEME VERSION     "980201"

VALUE DOMAIN       "Code-DATEX Publish Format"

DATA TYPE          ENUMERATED {other, ftp, tftp, dataPacket, …}

VALUE RULE         "ENUMERATED {other, ftp, tftp, dataPacket, …}"

}
```

C.2.54 DATEX.SUBSCRIBE_Serial_number-ulong

```
datexSubscribe-Serial-nbr DATA-ELEMENT ::= {

DESCRIPTIVE-NAME   "DATEX.SUBSCRIBE_Serial_number-ulong"

CONTEXT            "ITS"

DEFINITION         "The serial number for the subscription (request) being submitted. Each
                   subscription shall have an associated serial number; the number may be
                   used to update or cancel previously sent subscriptions. New subscriptions
                   shall not use a serial number which is already in use by the Client/
                   Supplier (server) pair."

CLASS              "Communication Networks"

CLASS SCHEME       "ITS Classification Scheme"

SCHEME VERSION     "980201"

VALUE DOMAIN       "Number-ULONG"

DATA TYPE          INTEGER (0..4294967295)

VALUE RULE         "INTEGER (0..4294967295)"

CONSTRAINTS        "The value of zero shall not be used"

}
```

C.2.55 DATEX.SUBSCRIBE_Status_code-datex12

```
datexSubscribe-Status-cd DATA-ELEMENT ::= {

DESCRIPTIVE-NAME   "DATEX.SUBSCRIBE_Status_code-datex12"

CONTEXT            "ITS"
DEFINITION          "An indication of the status of this subscription (request) message. If the
                    value is 'new', the request number shall be unique for the client-supplier
                    (server) pair and shall cause a new subscription to be handled by the
                    Supplier (server). If the value is 'update', the supplier (server) shall
                    update the previously stored request, i.e. the existing request with the
                    same serial number, according to the new information. The update feature
                    shall not be used to change the message id of the message field of the
                    SubscriptionData structure. If such a change is necessary, the existing
                    subscription shall be cancelled and a new subscription submitted."

CLASS               "Communication Networks"

CLASS SCHEME        "ITS Classification Scheme"

SCHEME VERSION      "980201"

VALUE DOMAIN        "Code-DATEX Status"

DATA TYPE           ENUMERATED {new, update}

VALUE RULE          "ENUMERATED {new, update}"

}
```

C.2.56 DATEX.TERMINATE_Reason_code-datex14

```
datexTerminate-Reason-cd DATA-ELEMENT ::= {

DESCRIPTIVE-NAME    "DATEX.TERMINATE_Reason_code-datex14"

CONTEXT             "ITS"

DEFINITION          "Indicates who initiated the termination of the session and a reason."

CLASS               "Communication Networks"

CLASS SCHEME        "ITS Classification Scheme"

SCHEME VERSION      "980201"

VALUE DOMAIN        "Code-DATEX Terminate"

DATA TYPE           ENUMERATED {other, serverRequested, clientRequested, serverShutdown,
                    clientShutdown, serverCommProblems, clientCommProblems, …}

VALUE RULE          "ENUMERATED {other, serverRequested, clientRequested, serverShutdown,
                    clientShutdown, serverCommProblems, clientCommProblems, …}"

}
```

C.2.57 DATEX.TRANSFER.DONE_FileName_text-memo

```
datexTransferDone-FileName-txt DATA-ELEMENT ::= {

DESCRIPTIVE-NAME    "DATEX.TRANSFER.DONE_FileName_text-memo"

CONTEXT             "ITS"

DEFINITION          "The name of the file which the Client is no longer attempting to retrieve."

CLASS               "Communication Networks"

CLASS SCHEME        "ITS Classification Scheme"

SCHEME VERSION      "980201"

VALUE DOMAIN        "Text-Memo"

DATA TYPE           UTF8String (SIZE (0..2000))
VALUE RULE          “UTF8String (SIZE (0..2000))”

}
```

C.2.58 DATEX.TRANSFER.DONE_Success_boolean

```
datexTransferDone-Success-bool DATA-ELEMENT ::= {

DESCRIPTIVE-NAME    “DATEX.TRANSFER.DONE_Success_boolean”

CONTEXT             “ITS”

DEFINITION          “Indicates whether or not the file transfer was successful.”

CLASS               “Communication Networks”

CLASS SCHEME        “ITS Classification Scheme”

SCHEME VERSION      “980201”

VALUE DOMAIN        “Boolean”

DATA TYPE           BOOLEAN

VALUE RULE          “BOOLEAN”

}
```

C.2.59 END.APPLICATION_Message_id

```
endApplication-Message-id DATA-ELEMENT ::= {

DESCRIPTIVE-NAME    “END.APPLICATION_Message_id”

CONTEXT             “ITS”

DEFINITION          “The object identifier associated with the end application message that
                    is contained within this datagram. Annex A provides the generic structure
                    within which end application messages are defined.”

CLASS               “Communication Networks”

CLASS SCHEME        “ITS Classification Scheme”

SCHEME VERSION      “980201”

VALUE DOMAIN        “As defined by the End Application message”

DATA TYPE           ISO14827-MESSAGE.&id

VALUE RULE          “As defined by the End Application message”

}
```

C.2.60 END.APPLICATION_Message_msg

```
endApplication-Message-msg DATA-ELEMENT ::= {

DESCRIPTIVE-NAME    “END.APPLICATION_Message_msg”

CONTEXT             “ITS”

DEFINITION          “The encoded content of the end application message. Annex A provides the
                    generic structure within which end application messages are defined.”

CLASS               “Communication Networks”

CLASS SCHEME        “ITS Classification Scheme”
SCHEME VERSION     "980201"
VALUE DOMAIN       "As defined by the End Application message"
DATA TYPE          ISO14827-MESSAGE.&MessageBody
VALUE RULE         "As defined by the End Application message"
}
```

C.2.61 TIME_Centiseconds_quantity

```
time-Centiseconds-qty DATA-ELEMENT ::= {
DESCRIPTIVE-NAME   "TIME_Centiseconds_quantity"
CONTEXT            "ITS"
DEFINITION         "The hundredths of seconds of the time."
CLASS              "Communication Networks"
CLASS SCHEME       "ITS Classification Scheme"
SCHEME VERSION     "980201"
VALUE DOMAIN       "INTEGER (0..99)"
DATA TYPE          INTEGER (0..99)
VALUE RULE         "INTEGER (0..99)"
}
```

C.2.62 TIME_Day_quantity

```
time-Day-qty DATA-ELEMENT ::= {
DESCRIPTIVE-NAME   "TIME_Day_quantity"
CONTEXT            "ITS"
DEFINITION         "The day of the month."
CLASS              "Communication Networks"
CLASS SCHEME       "ITS Classification Scheme"
SCHEME VERSION     "980201"
VALUE DOMAIN       "INTEGER (1..31)"
DATA TYPE          INTEGER (1..31)
VALUE RULE         "INTEGER (1..31)"
}
```

C.2.63 TIME_Deciseconds_quantity

```
time-Deciseconds-qty DATA-ELEMENT ::= {
DESCRIPTIVE-NAME   "TIME_Deciseconds_quantity"
CONTEXT            "ITS"
DEFINITION         "The tenths of seconds of the time."
CLASS             “Communication Networks”
CLASS SCHEME      “ITS Classification Scheme”
SCHEME VERSION    “980201”
VALUE DOMAIN      “INTEGER (0..9)”
DATA TYPE         INTEGER (0..9)
VALUE RULE        “INTEGER (0..9)”
}
```

C.2.64 TIME_Hour_quantity

```
time-Hour-qty DATA-ELEMENT ::= {
DESCRIPTIVE-NAME  “TIME_Hour_quantity”
CONTEXT           “ITS”
DEFINITION        “The hour units of the time.”
CLASS             “Communication Networks”
CLASS SCHEME      “ITS Classification Scheme”
SCHEME VERSION    “980201”
VALUE DOMAIN      “INTEGER (0..23)”
DATA TYPE         INTEGER (0..23)
VALUE RULE        “INTEGER (0..23)”
}
```

C.2.65 TIME_Milliseconds_quantity

```
time-Milliseconds-qty DATA-ELEMENT ::= {
DESCRIPTIVE-NAME  “TIME_Milliseconds_quantity”
CONTEXT           “ITS”
DEFINITION        “The thousandths of seconds of the time.”
CLASS             “Communication Networks”
CLASS SCHEME      “ITS Classification Scheme”
SCHEME VERSION    “980201”
VALUE DOMAIN      “INTEGER (0..999)”
DATA TYPE         INTEGER (0..999)
VALUE RULE        “INTEGER (0..999)”
}
```

C.2.66 TIME_Minute_quantity

```
time-Minute-qty DATA-ELEMENT ::= {
DESCRIPTIVE-NAME  “TIME_Minute_quantity”
CONTEXT            “ITS”
DEFINITION         “The minute units of the time.”
CLASS              “Communication Networks”
CLASS SCHEME       “ITS Classification Scheme”
SCHEME VERSION     “980201”
VALUE DOMAIN       “INTEGER (0..59)”
DATA TYPE          INTEGER (0..59)
VALUE RULE         “INTEGER (0..59)”
}
```

C.2.67 TIME_Month_quantity

```
time-Month-qty DATA-ELEMENT ::= {
DESCRIPTIVE-NAME   “TIME_Month_quantity”
CONTEXT            “ITS”
DEFINITION         “The Month of the year.”
CLASS              “Communication Networks”
CLASS SCHEME       “ITS Classification Scheme”
SCHEME VERSION     “980201”
VALUE DOMAIN       “INTEGER (1..12)”
DATA TYPE          INTEGER (1..12)
VALUE RULE         “INTEGER (1..12)”
}
```

C.2.68 TIME_Second_quantity

```
time-Second-qty DATA-ELEMENT ::= {
DESCRIPTIVE-NAME   “TIME_Second_quantity”
CONTEXT            “ITS”
DEFINITION         “The full seconds of the time.”
CLASS              “Communication Networks”
CLASS SCHEME       “ITS Classification Scheme”
SCHEME VERSION     “980201”
VALUE DOMAIN       “INTEGER (0..60)”
DATA TYPE          INTEGER (0..60)
VALUE RULE         “INTEGER (0..60)”
}
```

C.2.69 TIME_TimeZoneHour_quantity

```
time-TimeZoneHour-qty DATA-ELEMENT ::= {
DESCRIPTIVE-NAME    “TIME_TimeZoneHour_quantity”
CONTEXT             “ITS”
DEFINITION          “The number of hours offset from UTC for the local time.”
CLASS               “Communication Networks”
CLASS SCHEME        “ITS Classification Scheme”
SCHEME VERSION      “980201”
VALUE DOMAIN        “INTEGER (-13..13)”
DATA TYPE           INTEGER (-13..13)
VALUE RULE          “INTEGER (-13..13)”
}
```

C.2.70 TIME_TimeZoneMinute_quantity

```
time-TimeZoneMinute-qty DATA-ELEMENT ::= {
DESCRIPTIVE-NAME    “TIME_TimeZoneMinute_quantity”
CONTEXT             “ITS”
DEFINITION          “The minutes unit of the offset from UTC.”
CLASS               “Communication Networks”
CLASS SCHEME        “ITS Classification Scheme”
SCHEME VERSION      “980201”
VALUE DOMAIN        “INTEGER (0..59)”
DATA TYPE           INTEGER (0..59)
VALUE RULE          “INTEGER (0..59)”
}
```

C.2.71 TIME_Year_quantity

```
time-Year-qty DATA-ELEMENT ::= {
DESCRIPTIVE-NAME    “TIME_Year_quantity”
CONTEXT             “ITS”
DEFINITION          “The Year.”
CLASS               “Communication Networks”
CLASS SCHEME        “ITS Classification Scheme”
SCHEME VERSION      “980201”
VALUE DOMAIN        “INTEGER (-32768..32767)”
DATA TYPE           INTEGER (-32768..32767)
VALUE RULE          “INTEGER (-32768..32767)”
}


END
```

Annex D
(normative)

Value domains

D.1 General

The value domains defined within this annex are defined using the following ASN.1 Information Object Specification:

```
VALUE-DOMAIN ::= CLASS {
&name               PrintableString (SIZE (0..255))
&namecontext        PrintableString (SIZE (0..40))
&definition         PrintableString (SIZE (0..65535))
&formula            PrintableString (SIZE (0..255)) OPTIONAL
&source             PrintableString (SIZE (0..255)) OPTIONAL
&keyword            PrintableString (SIZE (0..255)) OPTIONAL
&remarks            PrintableString (SIZE (0..2000)) OPTIONAL
&Type
&rule               PrintableString (SIZE (0..65535)) OPTIONAL
}

WITH SYNTAX {
NAME                &name
CONTEXT             &namecontext
DEFINITION          &definition
[FORMULA            &formula]
[SOURCE             &source]
[KEYWORDS           &keyword]
[REMARKS            &remarks]
DATA TYPE           &Type
[VALID VALUE RULE   &rule]
}
```

The fields are defined to conform with those specified in IEEE 1489-1999.

D.2 Value domain structure

D.2.1 Boolean

```
boolean          VALUE-DOMAIN ::= {
NAME             “Boolean”
CONTEXT          “ITS”
DEFINITION       “Indicates either 'true' or 'false'.”
FORMULA
SOURCE           “ISO 8824-1”
KEYWORDS
REMARKS          “In PER (ISO 8825-2), this is a single bit where False is zero and True is
                 one. Other encoding schemes may have different representations, such as any
                 non-zero value being True.”
TYPE             BOOLEAN
VALID RULE       “BOOLEAN”
}
```

D.2.2 Code-Currency

```
code-datex1      VALUE-DOMAIN ::= {
NAME             “Code-Currency”
CONTEXT          “ITS”
DEFINITION       “Indicates the three letter currency code”
FORMULA
SOURCE
KEYWORDS
REMARKS
TYPE             OCTET STRING (SIZE (3))
VALID RULE       “ISO 4217”
}
```

D.2.3 Code-DATEX initiator

```
code-datex2      VALUE-DOMAIN ::= {
NAME             “Code-DATEX Initiator“
CONTEXT          “ITS”
DEFINITION       “Indicates the initiator of an action”
FORMULA
SOURCE
KEYWORDS
REMARKS

TYPE                ENUMERATED {serverInitiated, clientInitiated, …}

VALID RULE          “ENUMERATED {serverInitiated, clientInitiated, …}

}
```

D.2.4 Code-DATEX publication type

```
code-datex3         VALUE-DOMAIN ::= {

NAME                “Code-DATEX Publication Type

CONTEXT             “ITS”

DEFINITION          “Indicates the type of response being sent.”

FORMULA

SOURCE

KEYWORDS

REMARKS

TYPE                ENUMERATED {temporarilySuspended, resume, terminate-other, terminate-
                    dataNoLongerAvailable, terminatepublicationsBeingRejected, terminate-
                    PendingShutdown, terminate-processingMgmt, terminate-bandwidthMgmt, terminate-
                    accessDenied, unknownRequest, …}

VALID RULE          “ENUMERATED {temporarilySuspended, resume, terminate-other, terminate-
                    dataNoLongerAvailable, terminatepublicationsBeingRejected, terminate-
                    PendingShutdown, terminate-processingMgmt, terminate-bandwidthMgmt, terminate-
                    accessDenied, unknownRequest, …}”

}
```

D.2.5 Code-DATEX publish format

```
code-datex4         VALUE-DOMAIN ::= {

NAME                “Code-DATEX Publish Format”

CONTEXT             “ITS”

DEFINITION          “Indicates the format in which to publish information”

FORMULA

SOURCE

KEYWORDS

REMARKS

TYPE                ENUMERATED {other, ftp, tftp, dataPacket, …}

VALID RULE          “ENUMERATED {other, ftp, tftp, dataPacket, …}”

}
```

D.2.6 Code-DATEX cancel subscription

```
code-datex5         VALUE-DOMAIN ::= {

NAME                “Code-DATEX Cancel Subscription”
```