INTERNATIONAL STANDARD

ISO 10161-1

Second edition
1997-06-15

Information and documentation — Open Systems Interconnection — Interlibrary Loan Application Protocol Specification —

Part 1:
Protocol specification

Information et documentation — Interconnexion de systèmes ouverts (OSI) — Spécification du protocole d'application pour les prêts entre bibliothèques —

Partie 1: Spécification du protocole

Reference number
ISO 10161-1:1997(E)

Contents

International Organization for Standardization
Case postale 56 • CH-1211 Genève 20 • Switzerland
Internet central@iso.ch
X.400 c=ch; a=400net; p=iso; o=isocs; s=central

Printed in Switzerland

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.

Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

International Standard ISO 10161-1 was prepared by Technical Committee ISO/TC 46, *Information and Documentation*, Subcommittee SC 4, *Computer applications in information and documentation.*

This second edition cancels and replaces the first edition (ISO 10161-1:1993), which has been technically revised. It includes amendments specified in ISO 10161, DAM 1 and the corrections specified in Defect Reports 1-23.

ISO 10161 consists of the following parts, under the general title *Information and documentation — Open Systems Interconnection — Interlibrary Loan Application Protocol Specification*:

— *Part 1: Protocol specification*

— *Part 2: Protocol implementation conformance statement (PICS) proforma*

Annexes A to D form an integral part of this part of ISO 10161. Annexes E to H are for information only.

Introduction

This part of ISO 10161 is one of a set of International Standards produced to facilitate the interconnection of computer systems. It is related to other International Standards in the set as defined by the Reference Model for Open Systems Interconnection (ISO 7498). The Reference Model subdivides the area of standardization for interconnection into a series of layers of specification, each of manageable size.

The aim of Open Systems Interconnection is to allow, with a minimum of technical agreement outside the interconnection standards, the interconnection of computer systems:

a. from different manufacturers,

b. under different managements,

c. of different levels of complexity,

d. of different ages.

This part of ISO 10161 provides a protocol specification for Interlibrary Loan (ILL) communication. The ILL protocol operates in the Application Layer and allows the parties involved in an ILL-transaction to progress through the ILL-transaction in an orderly and defined way.

The ILL protocol has been designed to support the ILL services defined in ISO 10160, the ILL Application Service Definition, which generally requires invocation of external delivery services to fulfill an ILL request. The ILL protocol carries information that permits both automatic and operator-mediated invocation of external delivery services.

This part of ISO 10161 is one of a number of related standards supporting the interconnection of library systems. These standards can be used by themselves or in a cooperative manner to support library applications requiring a mixture of communications services. For example, ISO 23950, which supports remote access to bibliographic databases, could be used in conjunction with the ILL protocol to obtain item identification information. The control and management of interactions among such bibliographic applications are local matters that are outside the scope of this International Standard.

Security and accounting issues as they relate to ILL operations are for further study.

The specification technique used in this part of ISO 10161 is consistent with techniques used in defining other OSI protocols. Within most of this document, the technique is self-explanatory. The Abstract Syntax of the ILL Application Protocol Data Units (APDUs) is defined by means of the ASN.1 specification technique specified in ISO/IEC 8824.

This part of ISO 10161 contains eight annexes. Annexes A to D are normative. Annex A specifies the the state tables for the ILL protocol. Annex B specifies the encoding rules for generating a transfer syntax compatible with EDIFACT as defined in ISO 9735. Annex C specifies the object identifiers assigned in this part of ISO 10161 and registration requirements. Annex D defines the registration procedures for ILL EXTERNAL data type definitions. Annex E is an example of an ILL EXTERNAL data type registry entry. Annex F describes the possible mappings of this protocol onto supporting services. Annex G describes possible methods of using a document delivery protocol in conjunction with the ILL protocol. Annex H is a bibliography.

This page intentionally left blank

Information and documentation — Open Systems Interconnection — Interlibrary Loan Application Protocol Specification — Part 1: Protocol specification

1 Scope

This part of ISO 10161 defines the protocol for an ILL application-service-element (ASE). It specifies the behaviour which must be exhibited by a system in order to take part in the provision of the ISO interlibrary loan service.

It provides a formal statement of the rules of behaviour of each of the two or more entities participating in an ILL transaction. It specifies:

a. the actions to be taken on receiving request service primitives issued by an ILL service-user;

b. the actions to be taken on receiving application-protocol-data-units (APDUs);

c. the actions to be taken as a result of events within the local system.

It provides a specification (in clause 9) of the abstract syntax required to convey the ILL protocol APDUs.

It states the conformance requirements to be met by implementors of this protocol (in clause 10).

The scope of the ILL protocol is restricted to the interconnection of systems; it does not specify or restrict the possible implementation of interfaces within a computer system. Computer systems may range from stand-alone workstations to mainframes.

This part of ISO 10161 is intended for use by libraries, information utilities such as union catalogue centres, and any other system which processes bibliographic information. These systems may participate in an interlibrary loan transaction in the role of requester (i.e. an initiator of ILL requests), responder (i.e. a provider of bibliographic material or information) and/or intermediary (i.e. an agent that acts on behalf of a requester to find suitable responders).

Various interworking topologies are supported, ranging from simple two-party interactions, to multi-party interactions.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO 10161. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreeements based on this part of ISO 10161 are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO/IEC 646:1991, *Information technology - ISO 7-bit coded character set for information interchange.*

NOTE - ISO/IEC 646:1991 supersedes ISO 646:1983. However, when this part of ISO 10161 was under development, the previous edition was valid and this part of ISO 10161 is therefore based on this edition, which is given below.

ISO 646:1983, *Information Processing Systems - ISO 7-bit coded character set for information interchange.*

ISO 2108:1992, *Information and documentation - International standard book numbering (ISBN).*

NOTE - ISO 2108:1992 supersedes ISO 2108:1978. However, when this part of ISO 10161 was under development, the previous edition was valid and this part of ISO 10161 is therefore based on this edition, which is given below.

ISO 2108:1978, *Documentation - International standard book numbering (ISBN).*

ISO 2709:1996, *Information and documentation - Format for Information Exchange.*

NOTE - ISO 2709:1996 supersedes ISO 2709:1981. However, when this part of ISO 10161 was under development, the previous edition was valid and this part of ISO 10161 is therefore based on this edition, which is given below.

ISO 2709:1981, *Documentation - Format for bibliographic information interchange on magnetic tape.*

ISO 3297:1986, *Documentation - International standard serial numbering (ISSN).*

ISO 4217:1995, *Codes for the representation of currencies and funds.*
NOTE - ISO 4217:1995 supersedes ISO 4217:1981. However, when this part of ISO 10161 was under development, the previous edition was valid and this part of ISO 10161 is therefore based on this edition, which is given below.

ISO 4217:1981, *Codes for the representation of currencies and funds.*

ISO/IEC 7498-1:1994, *Information technology - Open Systems Interconnection - Basic Reference Model: The Basic Model.*

ISO 7498-2:1989, *Information processing systems - Open Systems Interconnection - Basic Reference Model - Part 2: Security Architecture.*

ISO 7498-3:1989, *Information processing systems - Open Systems Interconnection - Basic Reference Model - Part 3: Naming and addressing.*

ISO/IEC 7498-4:1989, *Information processing systems - Open Systems Interconnection - Basic Reference Model - Part 4: Management framework.*
NOTE - ISO/IEC 7498-1, ISO 7498-2, ISO 7498-3 and ISO/IEC 7498-4 supersede ISO 7498:1984. However, when this part of ISO 10161 was under development, the previous edition was valid and this part of ISO 10161 is therefore based on this edition, which is given below.

ISO 7498:1984, *Information processing systems - Open Systems Interconnection - Basic Reference Model.*

ISO 8601:1988, *Data elements and interchange formats - Information interchange - Representation of dates and times.*

ISO/IEC 8822:1994, *Information technology - Open Systems Interconnection - Presentation service definition.*
NOTE - ISO/IEC 8822:1994 supersedes ISO 8822:1988. However, when this part of ISO 10161 was under development, the previous edition was valid and this part of ISO 10161 is therefore based on this edition, which is given below.

ISO 8822:1988, *Information processing systems - Open Systems Interconnection - Connection Oriented Presentation Service Definition.*

ISO/IEC 8824:1990, *Information technology - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1).*

ISO/IEC 8825:1990, *Information technology - Open Systems Interconnection - Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1).*

ISO 9735:1988, *Electronic data interchange for administration, commerce and transport (EDIFACT) - Application Level Syntax Rules.*

ISO/IEC 9834-1:1993, *Information technology - Open Systems Interconnection - Procedures for the operation of OSI Registration Authorities - Part 1: General procedures.*

ISO/IEC 9834-2:1993, *Information technology - Open Systems Interconnection - Procedures for the operation of OSI Registration Authorities - Part 2: Registration procedures for OSI document types.*

ISO/IEC 10021-4:1990, *Information Technology - Text Communication - Message-Oriented Text Interchange Systems (MOTIS) - Part 4: Message Transfer System: Abstract Service Definition and Procedures.*

ISO 10160:1997, *Information and documentation - Open Systems Interconnection - Interlibrary Loan Application Service Definition.*

3 Definitions

For the purposes of this part of ISO 10161, the following definitions apply.

3.1 Reference Model Definitions

This part of ISO 10161 is based on the concepts developed in ISO 7498:1984 and makes use of the following terms found in it. These terms are replicated here as a convenience to the reader.

3.1.1 **Application Layer:** The seventh and highest layer in the Reference Model for Open Systems Interconnection (OSI); it serves as the window between correspondent application-processes which are using the OSI to exchange meaningful information.

3.1.2 **application-entity:** The aspects of an application-process pertinent to OSI.

3.1.3 **application-process:** An element within a real open system which performs the information processing for a particular application.

3.1.4 **application-protocol-data-unit:** A unit of data specified in an application-protocol and consisting of application-protocol-information and possibly application-user-data.

3.1.5 **application-service-element:** That part of an application-entity which provides an OSI environment capability, using underlying services when appropriate.

3.1.6 **(N)-service:** A capability of the (N)-layer and the layers beneath it, which is provided to (N+1)-entities at the boundary between the (N)-layer and the (N+1)-layer.

NOTE — An application-service does not provide a capability to higher layer entities, but rather to application-processes.

3.1.7 **presentation-service:** A capability of the Presentation Layer and the layers beneath it, which is provided to application-entities at the boundary between the Presentation and the Application Layer.

3.1.8 **transfer syntax:** The concrete syntax used in the transfer of data between open systems.

3.2 Abstract Syntax Notation One Definitions

This part of ISO 10161 makes use of the following terms defined in ISO/IEC 8824:1990.

3.2.1 **data type; type:** A named set of values.

3.2.2 **simple type:** A type defined by directly specifying the set of its values.

3.2.3 **structured type:** A type defined by reference to one or more other types.

3.2.4 **component type:** One of the types referenced when defining a structured type.

3.2.5 **value:** A distinguished member of a set of values.

3.3 Presentation Service Definition

This part of ISO 10161 makes use of the following term defined in ISO 8822:1988.

3.3.1 **abstract syntax:** Those aspects of the rules used in the formal specification of data which are independent of the encoding technique to represent the data.

3.4 Application Layer Structure Definitions

This part of ISO 10161 makes use of the following terms defined in ISO/IEC 9545:1989.

3.4.1 **application-association:** A cooperative relationship between two application-entity-invocations for the purpose of communication of information and coordination of their joint operation. This relationship is formed by the exchange of application-protocol-control-information using the Presentation Service.

3.4.2 **application-context:** A set of rules shared in common by two application-entity-invocations governing their behavior in order to enable their cooperative operation.

NOTE — An application-context is a shared conceptual schema for the universe of discourse for communication.

3.4.3 **application-context-definition:** The description of an application-context.

3.4.4 **application-entity-invocation:** A specific utilization of part or all of the capabilities of a given application-entity in support of the communications requirements of an application-process-invocation.

3.4.5 **application-process-invocation:** A specific utilization of part or all of the capabilities of a given application-process in support of a specific occasion of information processing.

3.5 Service Convention Definitions

This part of ISO 10161 makes use of the following terms defined in ISO/TR 8509:1987.

3.5.1 **indication primitive:** A representation of an interaction in which a service-provider either:

a. indicates that it has, on its own initiative, invoked some procedure; or

b. indicates that a procedure has been invoked by the service-user at the peer service-access-point.

3.5.2 **non-confirmed service:** A distinct part of the total (N)-service which does not result in an explicit confirmation from the service-provider to the initiating service-user.

3.5.3 **provider-initiated service:** A distinct part of the total (N)-service which is initiated by the service-provider rather than the service-user.

3.5.4 **request primitive:** A representation of an interaction in which a service-user invokes some procedure.

3.5.5 **service primitive:** An abstract, implementation-independent representation of an interaction between service-user and the service-provider.

3.5.6 **service-provider:** An abstract of the totality of those entities which provide a service to peer service-users.

3.5.7 **service-user:** An entity in a single open system that makes use of a service.

3.6 ILL Definitions

For the purposes of this part of ISO 10161, the following definitions apply to the ASN.1 value reference names and values which are associated with simple data types, as specified in clause 9. Of these definitions, the following are repetitions of those found in ISO 8459-1:1988:

author
call number
client identifier
client-status
country
edition
place-of-publication
post-office-box
postal-code
publisher
sponsoring body
street-and-number
title
transportation-mode
volume-issue

NOTE — volume-issue is defined here, but volume and issue are defined separately in ISO 8459-1:1988.

3.6.1 **account-number:** The number of an account to which a credit or debit is made. A requester typically has been assigned a separate account for each responder. (Variation of ISO 8459-1)

3.6.2 **additional-no-letters:** (additional-numbers-letters) A number or code identifying an item.

3.6.3 **already-forwarded:** Responder indication that an ILL request has already been forwarded.

3.6.4 **already-tried-list:** List of institutions which have been approached but were unable to supply requested item.

3.6.5 **answer:** A code representing a yes or no response.

3.6.6 **at-bindery:** Title is owned but the requested item is at the bindery.

3.6.7 **author:** Name of the person or corporate body responsible for the intellectual or artistic content of an item, including composers, creators or originators of an item.

3.6.8 **author-of-article:** Author of an item which is a component part of another item.

3.6.9 **badly-structured-APDU:** The structure of a received APDU does not conform to the standard notation and encoding defined in ISO 8824 and 8825, or to the EDIFACT encoding defined in ISO 9735 and Annex B of this International Standard. For example, a received APDU does not match its stated length.

3.6.10 **being-processed-for-supply:** Item is being retrieved, copied, and/or packaged for delivery.

3.6.11 **call-number:** Notation assigned to an item indicating its physical location in the owner institution.

3.6.12 **can-send-CHECKED-IN:** An indication by the responder that it is capable of supplying the CHECKED-IN APDU.

3.6.13 **can-send-RECEIVED:** An indication by the requester that it is capable of supplying the RECEIVED APDU.

3.6.14 **can-send-RETURNED:** An indication by the requester that it is capable of supplying the RETURNED APDU.

3.6.15 **can-send-SHIPPED:** An indication by the responder that it is capable of supplying the SHIPPED APDU.

3.6.16 **cannot-send-onward:** An intermediary is unable to send on a request due to communication problems.

3.6.17 **chargeable-units:** The number of units supplied for which there is a charge.

3.6.18 **charges:** The responder's charges for the provision of the requested service.

3.6.19 **city:** A phrase used to identify a city, town or village. (Variation of ISO 8459-1)

3.6.20 **client-identifier:** Number or code used to identify the client uniquely.

3.6.21 **client-name:** Name of the person or institution for which the item has been requested. (Variation of ISO 8459-1)

3.6.22 **client-signature-required:** Responder's stipulation that the client must sign the signature sheet enclosed with the item.

3.6.23 **client-status:** Professional level or position of the client.

3.6.24 **conditions:** A code used to indicate the conditions under which an item may be borrowed.

3.6.25 **copyright-compliance:** Requester notation indicating the applicable copyright regulations or laws to which the requester is adhering.

3.6.26 **correlation-information:** Information that is used to correlate an error report with the service request to which the report relates.

3.6.27 **cost:** The amount asked, taken or billed by the responder for the service supplied.

3.6.28 **cost-estimate:** Estimate of the cost to provide the service requested.

3.6.29 **cost-exceeds-limit:** Responder indication that the minimum cost to supply the request is greater than the amount authorized.

3.6.30 **country:** A phrase used to identify a country.

3.6.31 **currency-code:** A code identifying the currency of an amount, according to ISO 4217:1981.

3.6.32 **current-state:** A code identifying the state of the ILL-transaction.

3.6.33 **date-checked-in:** The date on which a loaned item is received back by the responder.

3.6.34 **date-due:** The date by which the loaned item should be returned to the responder. Note that this should reflect the latest date-due.

3.6.35 **date-for-reply:** The date by which a reply should be returned to the responder.

3.6.36 **date-of-last-transition:** The date on which the last state transition occurred.

3.6.37 **date-of-most-recent-service:** The date when the most recent service event occurred at the system providing the status report. This is either a service invoked by the system providing the status report or a service reflected in a received APDU.

3.6.38 **date-of-service:** The date on which a service concerning an ILL-transaction is invoked.

3.6.39 **date-received:** The date when the item is received by the requester.

3.6.40 **date-requested:** The date the ILL request was initiated by the requester.

3.6.41 **date-returned:** The date when the item was returned to the responder.

3.6.42 **date-shipped:** The date when the item was shipped to the requester.

3.6.43 **delivery-service:** The delivery service or method used in transporting a requested item. Either physical or electronic delivery may be used.

3.6.44 **desired-due-date:** The proposed due date for the renewed loan.

3.6.45 **duplicate-transaction-id:** The transaction-id value of an ILL-REQUEST APDU is an illegal duplicate, i.e. the value is identical to one for an existing ILL-REQUEST received from the same requester.

3.6.46 **edition:** All the copies of an item produced from one master copy or substantially the same type image, having the same contents, and, in the case of non-book materials, issued by a particular publishing agency or group of such agencies.

3.6.47 **electronic-delivery:** Delivery of an electronic representation of a document via a telecommunications-based data transfer mechanism. Delivery via transfer of a tangible magnetic or optical medium is excluded.

3.6.48 **estimated-date-available:** Date when an item placed on hold is expected to become available.

3.6.49 **expiry-date:** The date on which an ILL-transaction expires automatically.

3.6.50 **expiry-flag:** An indication of whether an expiry date has been set for an ILL-transaction, and if so, whether that date is the "need-before-date", or some other date.

3.6.51 **extended-postal-delivery-address:** Additional information in the postal address necessary to identify the exact point of delivery, e.g. room and floor number in a large building.

3.6.52 **final-responder:** Institution which supplies a requested item. This term is used when it is necessary to distinguish between the responder of an ILL-transaction and the responder of an ILL-sub-transaction.

3.6.53 **forward-flag:** An indication whether a received ILL-REQUEST has been forwarded from an intermediary.

3.6.54 **forward-note:** Note added to the ILL-REQUEST by the responder when it is forwarded to a new responder.

3.6.55 **general-problem:** Code indicating a general problem with a received APDU detected by the ILL service-provider that is not related to either the transaction-id or permissible state transitions.

3.6.56 **ILL-APDU-type:** Code identifying the type of APDU received.

3.6.57 **ILL-service-type:** A code for the type of ILL service requested. These can be listed in a preferred order.

3.6.58 **ILL-transaction:** A single complete instance of the whole ILL cycle, including all of the actions, service primitives, and messages involved from the initial ILL-Request until the cycle is concluded, as with the return of the requested material.

3.6.59 **in-process:** Item has been received but is not yet ready for use.

3.6.60 **in-use/on-loan:** Item is owned but is currently being used by a client or is on loan to another institution.

3.6.61 **initial-requester:** Person or institution which initiates an ILL-transaction; this term is used when it is necessary to distinguish between the requester of an ILL-transaction and the requester of an ILL sub-transaction.

3.6.62 **initial-requester-address:** Information identifying the telecommunications service and address by which the initial-requester can be reached.

3.6.63 **initiator-of-most-recent service:** Identification information of the requester or responder who initiated the most recent service.

3.6.64 **institution-symbol:** Number(s), letter(s) or a code serving to identify unambiguously and in an abbreviated format a library, institution or corporation that is participating in an ILL-transaction, e.g. institution's national union catalogue symbol.

3.6.65 **insured-for:** Notation of the amount of insurance purchased against loss or damage of items.

3.6.66 **intermediary-id:** Identification information of an ILL-transaction intermediary.

3.6.67 **intermediary-problem:** Code indicating that the intermediary has a problem in processing the request.

3.6.68 **invalid-transaction-id:** The transaction-id value of an ILL-REQUEST is invalid, e.g. the value violates the assignment rules of this part of ISO 10161, or an unknown person-or-institution symbol or person-or-institution-name is encountered.

3.6.69 **ISBN:** The International Standard Book Number assigned to a monograph as prescribed by ISO 3297:1986.

3.6.70 **ISSN:** The International Standard Serial Number assigned to a serial title as prescribed by ISO 2108:1978.

3.6.71 **item-type:** A code identifying the bibliographic form in which the item has been produced.

3.6.72 **lacking:** The title is owned but not the component part or pages requested.

3.6.73 **lacks-copyright-compliance:** Compliance with applicable copyright regulations or laws must be indicated before copying can be done.

3.6.74 **level-of-service:** A code that indicates the level of search detail required or the duration of time within which a response is required. Note that this code reflects regional or national conventions.

3.6.75 **library-use-only:** Responder indication that the item may not be removed from the requesting institution.

3.6.76 **location-address:** Information identifying the telecommunication service and address or the postal address by which the institution that owns the item can be reached.

3.6.77 **location-id:** The symbol or name of the institution that owns the requested item.

3.6.78 **location-note:** Additional information that supplements or corrects the bibliographic data provided in the ILL-REQUEST or that clarifies the location provided.

3.6.79 **locations-not-found:** No potential responder has been identified.

3.6.80 **lost:** Item declared missing and/or withdrawn from the collection.

3.6.81 **mandatory-messaging-not-supported:** The responder does not provide the SHIPPED and/or CHECKED-IN message(s).

3.6.82 **maximum-cost:** Maximum amount that will be paid to obtain an ILL service. (Variation of ISO 8459-1).

3.6.83 **medium-characteristics:** Technical specifications of the physical form in which the requested item is to be supplied.

3.6.84 **medium-type:** A code identifying the medium in which the item has been produced. (Variation of ISO 8459-1)

3.6.85 **mistyped-APDU:** The structure of the APDU does not conform to the structure defined in this part of ISO 10161. For example, it contains a data type not defined for that version of the protocol.

3.6.86 **monetary-value:** The value of an amount.

3.6.87 **most-recent-service:** A code identifying the last service event occurring at the system providing the status report. This is either a service invoked by the system providing the status report or a service reflected in a received APDU. A status report sent in response to a status query will not indicate STATUS-QUERY as the most-recent-service because this would not be informative.

3.6.88 **most-recent-service-note:** The contents of the note parameter from the most recent service primitive.

3.6.89 **name-of-institution:** A word, phrase or abbreviation which identifies a library, institution or corporation. (Variation of ISO 8459-1)

3.6.90 **name-of-person:** A word or combination of words and/or initials by which an individual is regularly known or designated and which identifies the person participating in the ILL-transaction.

3.6.91 **national-bibliography-no:** Information identifying the national bibliography and the corresponding record number for the desired item, e.g. Library of Congress Card Number (LCCN).

3.6.92 **need-before-date:** The date by which the item or a reply is needed.

3.6.93 **no-of-units-per-medium:** The number of physical pieces shipped per supplied medium.

3.6.94 **no-reproduction:** Item may not be photocopied or mechanically reproduced in whole or in part.

3.6.95 **non-circulating:** Item is held but is not available for loan.

3.6.96 **not-available:** Due to some technical problem the service-user is temporarily unable to process the service request.

3.6.97 **not-found-as-cited:** The item identification information is believed by the responder to be either incomplete or incorrect.

3.6.98 **not-on-shelf:** Item is owned by the institution but is not charged out and is not on shelf.

3.6.99 **not-owned:** Title is not owned by the responder.

3.6.100 **note:** Additional information which is not covered by any other data elements.

3.6.101 **notification note:** Note added to the FORWARD-NOTIFICATION by the responder.

3.6.102 **on-hold:** Item has been requested by another institution or person and will be supplied to that institution or person as soon as it becomes available.

3.6.103 **on-order:** Item has been ordered but has not been received by the responder.

3.6.104 on-reserve: Item is owned but set aside for restricted use.

3.6.105 pagination: The numbering of the pages of an item or a component part of an item. (Variation of ISO 8459-1)

3.6.106 payment-provided: Requester statement that payment of responder charges has been authorized, is being sent, or will be enclosed with the returned material.

3.6.107 permission-to-chain: An indication granting permission to the responder to initiate a chained sub-transaction with another responder.

3.6.108 permission-to-change-send-to-list: An indication granting permission to the responder to alter the contents of the send-to-list. The nature of the permitted changes depends on the value of the type "preference".

3.6.109 permission-to-forward: An indication granting permission to the responder to forward the request to another responder.

3.6.110 permission-to-partition: An indication granting permission to the responder to initiate a partitioned sub-transaction with another responder.

3.6.111 person-symbol: Number(s), letter(s) or code serving to identify unambiguously and in an abbreviated format a person who is participating in an ILL-transaction.

3.6.112 physical-medium: See medium-type.

3.6.113 place-of-publication: Geographical location of the publisher, or failing this, of the printer, distributor or manufacturer.

3.6.114 place-on-hold: A request that a hold be placed on the item which is to be supplied as soon as it becomes available.

3.6.115 policy-problem: Responder indication that there is no policy in place to permit the completion of the request.

3.6.116 poor-condition: Item is owned but its physical condition prohibits lending or reproduction.

3.6.117 post-office-box: A box number assigned by the post office.

3.6.118 postal-code: A code which identifies a given area within a city or other geographical area.

3.6.119 preference: An indication of whether the institutions listed in the send-to-list are to be approached in the order of the list or in any order.

3.6.120 prepayment-required: Responder indication that prepayment is required prior to the processing of the ILL-transaction.

3.6.121 protocol-version-not-supported: An APDU has been received with a protocol-version-number component identifying an unsupported version of the protocol.

3.6.122 protocol-version-num: A number identifying the protocol version in use.

3.6.123 publication-date: The date of issue of a work as designated by the publisher of the work.

3.6.124 publication-date-of-components: The publication date assigned by the publisher to identify the unique bibliographic components of a work.

3.6.125 publisher: Person(s) or organization(s) responsible for the publication of an item.

3.6.126 reason-locs-provided: A code used to indicate the reason why locations are provided in response to an ILL request.

3.6.127 reason-no-report: A code used to indicate the reason why no report can be provided in response to a STATUS-QUERY.

3.6.128 reason-not-available: A code used to indicate the reason for item unavailability.

3.6.129 reason-unfilled: A code used to indicate the reason for not filling an ILL request.

3.6.130 reciprocal-agreement: Requester indication of a prior agreement regarding what may be supplied and under what conditions.

3.6.131 region: A phrase used to identify a province, state, region or locale.

3.6.132 renewable: An indication of whether the supplied item is renewable or not.

3.6.133 report-source: Code indicating whether the initiating source of the error report is the service-user or the service-provider.

3.6.134 report-type: An indication of whether a report is available and if so, whether it is a status report, error report or both.

3.6.135 requester-id: Identification information of the ILL-transaction requester.

3.6.136 requester-note: Note provided by the ILL-transaction requester.

3.6.137 requester-optional-messages: An indication of whether the requester is capable of supplying the RECEIVED and RETURNED optional messages and whether the SHIPPED and/or CHECKED-IN optional messages are required or desired from the responder.

3.6.138 requester-CHECKED-IN: An indication by the requester as to whether or not it requires or desires to receive the CHECKED-IN APDU.

3.6.139 requester-SHIPPED: An indication by the requester as to whether or not it requires or desires the SHIPPED APDU.

3.6.140 resource-limitation: The service-user is unable to perform the requested service due to resource limitations.

3.6.141 responder-address: Information identifying the telecommunications service and address by which the responder can be reached.

3.6.142 responder-id: Identification information of the ILL-transaction responder.

3.6.143 responder-note: Note provided by the ILL-transaction responder.

3.6.144 responder-optional-messages: Indication of whether the responder is capable of sending the SHIPPED and/or CHECKED optional messages (for diagnostic purposes) and whether the RECEIVED and/or RETURNED messages are required or desired from the requester.

3.6.145 responder-RECEIVED: An indication by the responder as to whether or not it requires or desires to receive the RECEIVED APDU.

3.6.146 responder-RETURNED: An indication by the responder as to whether or not it requires or desires to receive the RETURNED APDU.

3.6.147 responder-specific-result: A reason provided in response to an ILL request which is specific to the responder, i.e. not specified in this part of ISO 10161.

3.6.148 responder-specific-service: A service provided by a responder which is specific to the responder, i.e. not specified in this part of ISO 10161.

3.6.149 retry-date: The date after which a request may be retried.

3.6.150 retry-flag: Requester indication that the ILL-transaction is or is not a retry of a previous one.

3.6.151 return-insurance-required: Amount of insurance against loss or damage required by the responder for the return of a loaned item.

3.6.152 returned-via: Requester's method of shipment used to return the item.

3.6.153 security-problem: An indication that the recipient has encountered a security problem that prevents it from processing the service request. The possible reasons are outside the scope of this part of ISO 10161.

3.6.154 series-title-number: Name given to a number of separate publications related to one another by the fact that each bears a collective title applying to the group or subgroup as a whole as well as its own title, and its number within that group. (Variation of ISO 8459-1)

3.6.155 send-to-list: List of potential responders for forwarded, chained or partitioned ILL-transactions.

3.6.156 shipped-conditions: Conditions under which an item may be used.

3.6.157 shipped-service-type: A code for the type of ILL service provided.

3.6.158 shipped-via: Lender's method of shipment used to send the item.

3.6.159 special-collections-supervision-required: Indication by the responder that the item must be used within the special collections department or archives of the requester.

3.6.160 sponsoring-body: The corporate body or organization that issued the item or that is associated with its authorship.

3.6.161 street-and-number: A number and/or phrase used to identify the location of a building within a city or a rural area.

3.6.162 supplemental-item-description: Additional item description information that may be represented in a machine-readable format, e.g. MARC record.

3.6.163 supplier-id: Identification information of the supplier of the requested item when the supplier is different from the responder.

3.6.164 supply-medium-type: A code identifying the medium in which the item is required. This can be listed in preferred order.

3.6.165 system-no: A number providing system specific identification of a bibliographic record for a requested item.

3.6.166 telecom-service-address: Unique number or code assigned to an electronic mailbox or service or to a participant in a communications network.

3.6.167 telecom-service-identifier: Unique name or code of the telecommunication service used for the ILL-transaction.

3.6.168 time-of-service: Time at which a service is invoked.

3.6.169 title: Name of an item consisting of a word or group of words intended to identify it.

3.6.170 title-of-article: Title of an item which is a component part of another item.

3.6.171 transaction-group-qualifier: An alphanumeric string uniquely identifying a set of related ILL-transactions, e.g. a series of referrals or an ILL request and its subsequent retry. This qualifier is unique within the scope of the original ILL-transaction requester's system. In combination with the requester's id, this provides a universally unique identifier for the ILL-transaction group.

3.6.172 transaction-id-problem: Code indicating a problem with the transaction-id in a received APDU.

3.6.173 transaction-qualifier: An alphanumeric string identifying all services and messages associated with a single ILL-transaction. Note that this is a unique string assigned by the initial requester of the ILL-transaction and applied by the ILL partners to all subsequent services and messages associated with the ILL-transaction. In combination with the requester's id and the transaction-group-qualifier this provides a universally unique identification for the ILL-transaction.

3.6.174 transaction-type: A code that identifies the type of ILL-transaction.

3.6.175 transportation-mode: A physical or non-electronic means of transporting the requested item when represented or stored on a tangible medium.

3.6.176 unable-to-perform: Code indicating the reason why the service-user is unable to perform the requested service.

3.6.177 unknown-transaction-id: There is no ILL-transaction corresponding to the transaction-id value of a received APDU.

3.6.178 unrecognized-APDU: The type of the received APDU is not one of the APDUs defined in this part of ISO 10161.

3.6.179 verification-reference-source: An authoritative source of bibliographic information used to identify or locate an item / any source used to identify or locate an item. (Variation of ISO 8459-1).

3.6.180 volume-issue: Identifier of a physical unit of a serial or multi-volume monograph / number, letter or word identifying a unit of an item which is, or the volumes of which are, published in parts.

3.6.181 volume/issue-not-yet-available: Title is owned but the requested component part has not yet been received.

3.6.182 will-pay-fee: Requester notation indicating that the requester agrees to pay the applicable fee.

3.6.183 will-supply-results: Code for identifying additional information associated with the "will-supply" result in the response to an ILL request.

4 Abbreviations

ACSE	association control service element
APDU	application-protocol-data-unit
ASE	application-service-element
ASN.1	abstract syntax notation one
ASO	application service object
EDIFACT	Electronic Data Interchange For Administration, Commerce and Transport
ILL	interlibrary loan
ILLPM	ILL protocol machine
MHS	Message Handling System
MOTIS	Message Oriented Text Interchange System
MTS	Message Transfer System
OSI	open systems interconnection

5 Overview of the Protocol

5.1 Service Provision

The protocol specified in this part of ISO 10161 provides the services defined in ISO 10160. These services are listed in table 1.

Table 1 — ILL Services

SERVICE	TYPE
ILL-REQUEST	non-confirmed
FORWARD	non-confirmed
FORWARD-NOTIFICATION	provider-initiated
SHIPPED	non-confirmed
ILL-ANSWER	non-confirmed
CONDITIONAL-REPLY	non-confirmed
CANCEL	non-confirmed
CANCEL-REPLY	non-confirmed
RECEIVED	non-confirmed
RECALL	non-confirmed
RETURNED	non-confirmed
CHECKED-IN	non-confirmed
OVERDUE	non-confirmed
RENEW	non-confirmed
RENEW-ANSWER	non-confirmed
LOST	non-confirmed
DAMAGED	non-confirmed
MESSAGE	non-confirmed
STATUS-QUERY	non-confirmed
STATUS-OR-ERROR-REPORT	non-confirmed
EXPIRY	provider-initiated

5.2 Supporting Services Assumed

The ILL protocol is specified to potentially operate in both store-and-forward and connection-oriented modes. The specification of mappings to specific supporting services are to be provided in application-context-definitions and functional profiles. Annex F describes some possible mappings to supporting services.

5.3 Model

In the abstract, the operation of the ILL protocol is modelled by the interaction of ILL protocol machines (ILLPM). The ILLPMs communicate by exchanging ILL APDUs through the use of the abstract services "send APDU" and "receive APDU" at their lower boundary. At their upper boundary, the ILLPMs provide the services defined in ISO 10160.

An ILLPM is driven by the receipt of input events from its ILL service-user, supporting service provider or from an internal timer. The input events from the ILL service-user are request primitives and from the ILL supporting service they are received APDUs. The input event from the internal timer is timer expiry.

An ILLPM responds to input events by issuing output events to its supporting service and to its ILL service-user. The output events to the supporting service are the sending of ILL APDUs. The output events to its ILL service-user are ILL indication primitives.

The receipt of an input event, the generation of dependent actions, and the resultant output event are considered to be an indivisible action.

Logically there is a separate invocation of a set of ILLPMs for each ILL-transaction. Those invocations maintain the state information for a given ILL-transaction. The lifetime of an ILLPM invocation is as long as is required to complete the associated ILL transaction. This state information must be preserved across all instances of use of supporting services, for example, it must be maintained separate from any state information associated with underlying application-associations. Many ILL-transactions, and hence many ILLPMs, may be in existence simultaneously.

The ILL protocol machine expects that its APDUs will be transferred reliably without loss, alteration or addition of any information. It is tolerant to repeated APDUs. The protocol allows for user-initiated repetition of service requests to recover from lost, out-of-sequence or malformed APDUs. The mechanisms whereby such problems are detected and users are notified are outside the scope of this part of ISO 10161.

6 ILL APDUs

An ILL APDU is a unit of information which passes between two peer ILL ASEs involved in an ILL-transaction. This clause lists the APDUs used in the ILL application and describes the use and meaning of these APDUs.

ILL-REQUEST	used by the requester to request the loan, the location, or a photocopy of an item, or a cost estimate for a service from a library.
FORWARD-NOTIFICATION	used by the service provider to inform the requester that its request has been forwarded, and to whom.
SHIPPED	used by the responder to indicate that an item has been shipped.
ILL-ANSWER	used by the responder to send a response to the requester (possible responses are: CONDITIONAL, RETRY, UNFILLED, LOCATIONS-PROVIDED, WILL-SUPPLY, HOLD PLACED and ESTIMATE).
CONDITIONAL-REPLY	used by the requester to reply to an ILL-ANSWER with a status of CONDITIONAL. Possible answers are YES (we will meet the conditions) and NO (we do not agree to meet the conditions).
CANCEL	used by the requester to initiate cancellation of an ILL-transaction.
CANCEL-REPLY	used by the responder to respond to a CANCEL request.
RECEIVED	used by the requester to indicate that an item has been received.
RECALL	used by the responder to request the immediate return of an item.
RETURNED	used by the requester to indicate that a borrowed item has been returned.
CHECKED-IN	used by the responder to acknowledge the return of a borrowed item.
OVERDUE	used by the responder to notify the requester that an item is overdue.
RENEW	used by the requester to request the renewal of a borrowed item.
RENEW-ANSWER	used by the responder to respond to a RENEW request.
LOST	used by either the requester or the responder to notify the other that an item has been lost.
DAMAGED	used by either the requester or the responder to notify the other that an item has been damaged.
MESSAGE	used by either the requester or the responder to communicate with the other without affecting the state of the ILL-transaction.
STATUS-QUERY	used by either the requester or the responder to request the status of the ILL-transaction at the remote site.
STATUS-OR-ERROR-REPORT	used by either the requester or the responder to report the current state of a ILL-transaction, and any other relevant information available, or to report an error condition.
EXPIRED	used by the responder system to notify the requester of ILL-transaction expiry. The sending of this APDU is initiated by the ILL service-provider and not by the service-user.

7 Transaction Information

An ILL system must maintain the following information for each ILL-transaction:

— transaction identification

— protocol state

— protocol variables

— expiry timer

— request information

— history information.

7.1 Transaction Identification

An ILL APDU or service primitive is associated with an ILL-transaction by means of a transaction-id.

The transaction-id satisfies the following requirements:

a. uniqueness in the case of chaining and partitioning;

b. allows sub-transactions to be related to the parent ILL-transaction;

c. allows multiple ILL-transactions to be related together into a logical grouping, e.g. when a requester refers an ILL-request to many responders in turn; or a retry is to be related to the original ILL-request.

It consists of the following components:

initial-requester-id (optional)	identifies the ILL-transaction initiator;
transaction-group-qualifier (mandatory)	distinguishes a group of ILL-transactions from all other active ILL-transaction groups associated with the initial-requester;
transaction-qualifier (mandatory)	distinguishes an ILL-transaction from all other ILL-transactions within an ILL-transaction group;
sub-transaction-qualifier (optional)	distinguishes a sub-transaction from all other sub-transactions initiated by the intermediary.

Table 2 summarizes the use of the components of transaction-id.

Table 2 — Components of Transaction-Id

Simple Transaction

Initial-requester-id	Transaction-group qualifier	Transaction qualifier	Sub-transaction qualifier
optional	mandatory	mandatory	not used
set by requester	set by requester	set by requester	

Chained or Partioned Transaction

Initial-requester-id	Transaction-group qualifier	Transaction qualifier	Sub-transaction qualifier
mandatory	mandatory	mandatory	mandatory
set by intermediary	set by requester	set by requester	set by intermediary

The initial-requester-id identifies the initiator of the ILL-transaction. It may take on any of the possible representations identified in 7.5.1 for system-id, but if the person-or-institution-symbol representation is used, then the ILL-transaction is constrained to occur only within a domain where that symbol is unambiguous, e.g. within a country. When this component is assigned a value by the intermediary, the value is the same as "requester-id" type in the original ILL-REQUEST.

NOTE — The internal components of initial-requester-id may need to be further defined within an application profile definition.

For a simple two-party ILL-transaction, the initial-requester-id component need not be included as part of the transaction-id because this information is available already in the requester-id type of the ILL-REQUEST APDU. However, if the initial-requester-id is present in an ILL-REQUEST APDU, then it must be present with the same value in all other messages associated with that ILL-transaction. Also, the sub-transaction-qualifier is not needed.

The transaction-group-qualifier is a mandatory component of the transaction-id. It is unique within the scope of the initial-requester. The initial-requester is responsible for assigning values to this qualifier to satisfy this rule. A transaction-group-qualifier can be reused only when there are no ILL-transactions active within that group and there is little probability that new related transactions will be initiated. The transaction-group-qualifier can be used to relate multiple ILL-transactions, e.g. in the case of referrals or in the case of retries where it is desired to distinguish the retries from the original ILL-transaction.

The transaction-qualifier uniquely identifies an ILL-transaction within a transaction group. It is a mandatory component of a transaction-id.

For a sub-transaction, all components are needed. The initial-requester-id together with the transaction-group-qualifier and the transaction-qualifier ensure that the ILL-transaction is unique within the domain of uniqueness of the initial-requester-id. The sub-transaction-qualifier together with the requester-id information conveyed in a sub-transaction ILL-REQUEST APDU ensure sub-transaction uniqueness.

In the case of chained ILL-transactions, there may be a sequence of sub-transactions linked in a chain. In such cases, each sub-transaction is considered to be a sub-transaction of the original ILL-transaction, so each intermediary which initiates a new sub-transaction replaces the current sub-transaction-qualifier with a new one which is unique within the scope of the intermediary. All sub-transactions of a particular ILL-transaction can be distinguished on the basis of the combination of the transaction-id and the requester-id.

7.2 Protocol States

The ILL protocol states are identical to the ILL-transaction states defined in the ILL Service Definition.

7.2.1 Requester States

The requester state is the state of processing of an ILL-transaction at the requester. It may be one of the following:

IDLE	The ILL-transaction has not started.
PENDING	A request has been made and the item is expected from the responder; or a message has been received indicating that the item will be supplied or has been placed on hold; or that the request has been forwarded to another institution.
NOT-SUPPLIED	The ILL-transaction has reached a stage where the request cannot be filled by the responder.
CONDITIONAL	The ILL-transaction has reached a stage where the request can only be filled if the requester agrees to meet specified conditions.
CANCEL-PENDING	The requester has initiated cancellation of the ILL-transaction but no response has been received from the responder.
CANCELLED	The ILL-transaction has been cancelled by the responder.
SHIPPED	The item has been shipped to the requester.
RECEIVED	The item has been received from the responder.
RENEW/PENDING	A request has been made for the renewal of the item.
RENEW/OVERDUE	A request has been made for the renewal of an item which is overdue.
OVERDUE	The requester has been notified that the item is overdue.
NOT-RECEIVED/OVERDUE	The responder has sent an overdue notification for an item that has not yet been received.
RECALL	The item has been recalled by the responder.
RETURNED	The item has been shipped back to the responder.
LOST	The item has been lost.

7.2.2 Responder States

The responder state is the state of processing of an ILL-transaction at the responder. It may be one of the following:

IDLE	The responder has not received a request.
IN-PROCESS	A request has been received and is being processed by the responder; the item has not been shipped.
FORWARD	The request has been forwarded to another institution.
NOT-SUPPLIED	The responder has responded to a request with an ILL-ANSWER of RETRY, UNFILLED, LOCATIONS-PROVIDED or ESTIMATE; or the ILL-transaction has expired.
CONDITIONAL	The request can only be filled if the requester agrees to meet specified conditions.
CANCEL-PENDING	The requester has initiated cancellation of the ILL-transaction but no response has been provided by the responder.
CANCELLED	The ILL-transaction has been cancelled by the responder.
SHIPPED	The item has been shipped to the requester.
RENEW/PENDING	A request has been made for the renewal of the item.
RENEW/OVERDUE	A request has been made for the renewal of an item which is overdue.
OVERDUE	The responder has informed the requester that the item is overdue.
RECALL	The item has been recalled by the responder.
CHECKED-IN	The item has been received back from the requester.
LOST	The item has been lost.

7.2.3 Terminal States

For the requester, responder and intermediary, there are certain states, known as terminal states which , when reached, will not result in any further transitions for a given ILL-transaction. The only exception is a transition to another terminal state.

An ILL-transaction would normally be maintained in a terminal state for a certain length of time before the ILL-transaction information is made inaccessible to the peer or is deleted. This length of time is a local management decision, or subject to agreement by implementors. Note, however, that the requirement to respond to a Status-Query request, and the requirement to relay messages suggests that this length of time may have to be sufficient to allow the information to be accessible for the

maximum loan period plus renewal periods and delivery time. For non-returnable items, the length of time may have to be sufficient to allow the requester to determine that an expected item will not be received and to invoke the Status-Query or Lost services.

The possible terminal states for the requester are:

— NOT-SUPPLIED
— CANCELLED
— RECEIVED (if a non-returnable item is received)
— RETURNED
— LOST

The possible terminal states for the responder are:

— NOT-SUPPLIED
— CANCELLED
— FORWARD
— SHIPPED (if a non-returnable item is shipped)
— CHECKED-IN
— LOST

The terminal state for a particular ILL-transaction will depend on its circumstances. For example, when a photocopy is provided, SHIPPED is the terminal state for the responder, while RECEIVED is the terminal state for the requester.

The possible terminal states for the intermediary are:

— NOT SUPPLIED
— FORWARD
— CANCELLED
— SHIPPED

7.2.4 Intermediary States

An intermediary involved in a chained or partitioned ILL-transaction plays both the role of responder (in its interactions with the requester) and requester (in its interactions with the responder). It maintains separate state information for each of these sets of interactions.

For unsuccessful sub-transactions, the terminal states for the intermediary requester are NOT-SUPPLIED and CANCELLED; for the intermediary responder, they are NOT-SUPPLIED, CANCELLED and FORWARD.

For a successful ILL-transaction, the terminal state for an intermediary in both the roles of requester and responder is the SHIPPED state.

7.3 Protocol Variables

The following protocol variables affect ILL protocol behaviour. These variables are maintained by the ILLPM and are set according to parameters of service primitives or received APDUs. An intermediary maintains the appropriate protocol variables separately for its requester and responder roles.

RETURN	used to indicate whether a shipped item is required to be returned to the responder. It takes on the values TRUE or FALSE.
FWD	used to indicate whether or not a request may be forwarded by the responder. It takes on the values TRUE or FALSE.
PART	used by an intermediary to indicate whether an ILL-transaction can be or has been partitioned. It takes on the values TRUE or FALSE.
CHAIN	used by an intermediary to indicate whether an ILL-transaction can be or has been chained. It takes on the values TRUE or FALSE.
SEQUENCE-TIME-STAMP	used to preserve the time stamp of the last received APDU. This variable is set to the value of the type "date-time-of-this-service" in the received APDU, and serves to detect out-of-sequence APDUs.
REPEAT-TIME-STAMP	used to preserve the time stamp of the last received APDU which caused a state change. This variable is set to the value of the type "date-time-of-this-service", or if the received APDU is itself a repeated one, then this protocol variable is set to the value of the type "date-time-of-original-service".

CURRENT-PARTNER-ID	used to preserve the identification of the current partner for the ILL-transaction for the purpose of APDU sequence validation. For the requester, it is set initially to the value of the parameter "responder identification" in an ILL-REQUEST service primitive. Thereafter it is updated to the value of "responder identification" in a received APDU when that value is not equal to the value of CURRENT-PARTNER-ID and is not one of the values of PREVIOUS-PARTNER-IDS. Similarly, for the responder, it is set initially to the value of the field "requester-id" in a received ILL-REQUEST and is updated to the value of the same field in subsequent APDUs.
PREVIOUS-PARTNER-IDS	used to preserve the identification of previous partners for the ILL-transaction. Whenever the value of CURRENT-PARTNER-ID is changed, the previous value is added to this protocol variable. Since the current partner identification may change more than once in the course of an ILL-transaction, e.g. as a result of multiple instances of forwarding, this protocol variable may consist of a sequence of values.

7.4 Expiry Timer

The following timer must be maintained by the responder if it supports the EXPIRY service:

EXPIRY specifies the date of ILL-transaction expiry.

This information is provided by the requester in the ILL-REQUEST.

7.5 Request Information

The entire contents of the original ILL-REQUEST must be preserved to support ILL-transaction forwarding, partitioning and chaining. An intermediary may however change the contents of an ILL-REQUEST when the request is forwarded or a sub-transaction is initiated. For example, it may correct bibliographic information provided, or it may change the permission-to-chain to FALSE in a sub-transaction when the value in the original ILL-REQUEST was TRUE.

An intermediary should always change the value of requester-id to its own when a sub-transaction is initiated. Note that the identity of the initial requester is preserved in the transaction-id.

7.5.1 System-id

The ILL-REQUEST information includes the identification of the requester and responder, both of which are of type system-id, which consists of one or more of the following components:

person-or-institution-symbol	Number(s), letter(s) or a code serving to identify unambiguously to the responder, in an abbreviated format, a person or institution which is participating in an ILL-request.
name-of-person-or-institution	a word, phrase or abbreviation which identifies a person, library, institution or corporation

A person-or-institution-symbol may have an internal structure, e.g. to indicate an institution's affiliation with a particular utility, consortium, etc., or to identify individual workstations (i.e. application processes) within an institution. Any such structure is outside the scope of this part of ISO 10161 and is the responsibility of the appropriate authorities.

Since the scope of a person-or-institution-symbol is limited, ILL-transactions which use only this information as part of system-id are constrained to operate only within the domain of the symbol. ILL-transactions which involve persons or institutions outside that domain must include "name-of-person-or-institution" as part of the system-id.

The name-of-person-or-institution represents a free-form description of a person or institution title.
The requester and responder shall use a constant value for a system-id throughout an ILL-transaction. Thus, if only a person-or-institution-symbol is assigned initially by the requester, then only a symbol shall be used thereafter and the value of the symbol shall not change. If both a person-and-institution-symbol and a name-of-person-or-institution are assigned initially by the requester, then both the symbol and the name shall be used thereafter and the value of the symbol and of the name shall not change.

7.6 History Information

The following additional history information is provided as part of a STATUS-OR-ERROR-REPORT for an ongoing ILL-transaction and, therefore, must be maintained throughout the lifetime of an ILL-transaction:

— date-of-last-transition
— most-recent-service
— date-of-most-recent-service
— initiator-of-most-recent-service
— shipped-service-type
— transaction-results
— most-recent-service-note

The "most-recent-service" indicates which ILL service primitive was last invoked or which APDU (indication) last received by this ILL service-user. Out-of-sequence service indications are to be considered as valid service indications for the purpose of the "most-recent-service" component of the history report. An exception is when a STATUS-OR-ERROR-REPORT is sent in response to a STATUS-QUERY, in which case the "most-recent-service" is the one prior to the STATUS-QUERY.

The "date-of-most-recent-service" indicates the date of the "most-recent-service".

The "initiator-of-most-recent-service" identifies the requester or responder who initiated the most recent service.

The "shipped-service-type" should reflect the most current information in the "shipped-service-type" parameter from the SHIPPED or RECEIVED service or from the SHIPPED or RECEIVED APDU, e.g. if a requester has received a SHIPPED APDU and then invokes a RECEIVED.request, then the value in the RECEIVED.request is used. This information may not be available for all ILL-transactions.

The "transaction-results" should reflect the value of the "transaction-results" parameter contained in the ILL-ANSWER. This information may not be available for all ILL transactions.

The "most-recent-service-note" provides the contents of the note parameter from the most recent service primitive.

If status information is requested for a terminated ILL-transaction, no history information may be available.

8 Elements of Procedure

8.1 Events and Actions

This clause describes the allowable events and actions for a single ILL-transaction. The requester, responder and intermediary must react to various events by taking specific actions. An event is defined to be the receipt of an APDU from another ILL ASE or the receipt of a service primitive from the local ILL service user. An action can take the form of the issuance of a service primitive to the local ILL service user, or the transmission of one or more APDUs to a remote ILL ASE.

The following subclauses identify which events and actions are allowable for the requester, responder and intermediary and which, by implication, are not.

8.1.1 Requester Events

This subclause describes the allowable events for a requester.

The events relating to service primitive requests from the ILL service user are:

— ILL-REQUEST.request

— CONDITIONAL-REPLY.request:
answer = YES

— CONDITIONAL-REPLY.request:
answer = NO

— CANCEL.request

— RECEIVED.request

— RETURNED.request

— RENEW.request

— LOST.request

— DAMAGED.request

— MESSAGE.request

— STATUS-QUERY.request

— STATUS-OR-ERROR-REPORT.request

The events relating to ILL APDUs received from the remote ILL ASE are the following:

— FORWARD-NOTIFICATION

— SHIPPED

— ILL-ANSWER:
Request-result = CONDITIONAL

— ILL-ANSWER: Request-result = RETRY

— ILL-ANSWER: Request-result = UNFILLED

— ILL-ANSWER:
Request-result = LOCATIONS-PROVIDED

— ILL-ANSWER:
Request-result = WILL-SUPPLY
— ILL-ANSWER:
Request-result = HOLD-PLACED
— ILL-ANSWER:
Request-result = ESTIMATE
— CANCEL-REPLY
— RECALL
— CHECKED-IN
— OVERDUE
— RENEW-ANSWER: answer = YES
— RENEW-ANSWER: answer = NO
— LOST
— DAMAGED
— MESSAGE
— STATUS-QUERY
— STATUS-OR-ERROR-REPORT
— EXPIRED

8.1.2 Requester Actions

The following service primitives can be issued to the ILL service user:

— FORWARD-NOTIFICATION.indication
— SHIPPED.indication
— ILL-ANSWER.indication:
Request-result = CONDITIONAL
— ILL-ANSWER.indication: Request-result = RETRY
— ILL-ANSWER.indication: Request-result = UNFILLED
— ILL-ANSWER.indication:
Request-result = LOCATIONS-PROVIDED
— ILL-ANSWER.indication:
Request-result = WILL-SUPPLY
— ILL-ANSWER.indication:
Request-result = HOLD-PLACED
— ILL-ANSWER.indication:
Request-result = ESTIMATE
— CANCEL-REPLY.indication
— RECALL.indication
— CHECKED-IN.indication
— OVERDUE.indication
— RENEW-ANSWER.indication:
answer = YES
— RENEW-ANSWER.indication: answer = NO
— LOST.indication
— DAMAGED.indication
— MESSAGE.indication
— STATUS-QUERY.indication
— STATUS-OR-ERROR-REPORT.indication
— EXPIRED.indication

The following ILL APDUs can be sent to the remote ILL ASE:

— ILL-REQUEST
— CONDITIONAL-REPLY:
answer = YES
— CONDITIONAL-REPLY:
answer = NO
— CANCEL
— RECEIVED - This action is optional.
— RETURNED - This action is optional.
— RENEW
— LOST
— DAMAGED
— MESSAGE
— STATUS-QUERY
— STATUS-OR-ERROR-REPORT

8.1.3 Responder Events

This subclause describes the allowable events for a responder.

The events relating to service primitive requests from the ILL service-user are:

— FORWARD.request
— SHIPPED.request
— ILL-ANSWER.request: Request-result = CONDITIONAL
— ILL-ANSWER.request: Request-result = RETRY
— ILL-ANSWER.request: Request-result = UNFILLED
— ILL-ANSWER.request: Request-result = LOCATIONS-PROVIDED
— ILL-ANSWER.request: Request-result = WILL-SUPPLY
— ILL-ANSWER.request: Request-result = HOLD-PLACED
— ILL-ANSWER.request: Request-result = ESTIMATE
— CANCEL-REPLY.request
— RECALL.request
— CHECKED-IN.request
— OVERDUE.request
— RENEW-ANSWER.request: answer = YES
— RENEW-ANSWER.request: answer = NO
— LOST.request
— DAMAGED.request
— MESSAGE.request
— STATUS-QUERY.request
— STATUS-OR-ERROR-REPORT.request

The events relating to ILL APDUs received from the remote ILL ASE are the following:

— ILL-REQUEST
— CONDITIONAL-REPLY: answer = YES
— CONDITIONAL-REPLY: answer = NO
— CANCEL
— RECEIVED
— RETURNED
— RENEW

— LOST
— DAMAGED
— MESSAGE
— STATUS-QUERY
— STATUS-OR-ERROR-REPORT

The following local events can occur within the ILL service provider:

— EXPIRY TIMEOUT

8.1.4 Responder Actions

The following ILL service primitives can be issued to the ILL service user:

— ILL-REQUEST.indication
— CONDITIONAL-REPLY.indication: answer = YES
— CONDITIONAL-REPLY.indication: answer = NO
— CANCEL.indication
— RECEIVED.indication
— RETURNED.indication
— RENEW.indication
— LOST.indication
— DAMAGED.indication
— MESSAGE.indication
— STATUS-QUERY.indication
— STATUS-OR-ERROR-REPORT.indication
— EXPIRED.indication

The following ILL APDUs can be sent to the remote ILL ASE:

— ILL-REQUEST
— SHIPPED -This action is optional.
— FORWARD-NOTIFICATION
— ILL-ANSWER: Request-result = CONDITIONAL
— ILL-ANSWER : Request-result = RETRY
— ILL-ANSWER: Request-result = UNFILLED
— ILL-ANSWER: Request-result = LOCATIONS-PROVIDED
— ILL-ANSWER: Request-result = WILL-SUPPLY
— ILL-ANSWER: Request-result = HOLD-PLACED
— ILL-ANSWER: Request-result = ESTIMATE
— CANCEL-REPLY
— RECALL
— CHECKED-IN - This action is optional.
— OVERDUE
— RENEW-ANSWER: answer = YES
— RENEW-ANSWER: answer = NO
— LOST
— DAMAGED
— MESSAGE
— STATUS-QUERY
— STATUS-OR-ERROR-REPORT
— EXPIRED

8.1.5 Intermediary Events and Actions

This subclause describes the allowable events for an intermediary. An intermediary participating in a chained or partitioned ILL-transaction plays both the roles of requester and responder.

In the role of requester, the intermediary events and actions are the same as for a requester, although the states may differ.

In the role of responder, the intermediary events and actions are the same as for a responder, although the states may differ. The one difference is that the SHIPPED APDU is mandatory.

8.2 Procedural Rules for All Parties

8.2.1 Sending and Receiving APDUs

With the exception of the FORWARD service, each request service primitive results in the preparation and transmission of the APDU of the corresponding name.

In the case of the FORWARD.request, two APDUs are prepared: an ILL-REQUEST APDU is sent to the new responder, while a FORWARD-NOTIFICATION APDU is sent to the requester.

The receipt of a valid APDU results in a corresponding indication service primitive.

APDUs are prepared and sent only upon the explicit request of the ILL service user, with the exception of the EXPIRED APDU which is sent by the responder's system in the case of timer expiry.

8.2.2 Transaction Phases

An ILL-transaction can have two phases: processing and tracking. The processing phase is mandatory for all ILL-transactions while the tracking phase is applicable only for ILL-transactions where a returnable item, e.g. a monograph, is supplied.

The processing phase for the requester includes all events and actions up to and including the receipt of the requested item. This phase normally terminates in the RECEIVED state.

The processing phase for the responder includes all events and actions up to and including the shipping of the requested item. This phase normally terminates in the SHIPPED state.

For the intermediary requester, the processing phase includes all events and actions up to and including receipt of the SHIPPED indication; for the intermediary responder, the processing phase includes all events up to and including issuing the SHIPPED request. For both the intermediary requester and responder, the processing phase normally terminates in the SHIPPED state.

The tracking phase includes all events and actions after shipping and receipt of a returnable item, including renewals, overdues and item return.

The existence of a tracking phase for an ILL-transaction is indicated by the RETURN protocol variable. A value of TRUE indicates that the tracking phase is applicable and the associated procedures must be followed. A value of FALSE indicates that the tracking phase is not required and the corresponding events are not permitted. There is no tracking phase for ILL-transactions involving non-returnable items, e.g. a photocopy.

The responder sets the RETURN variable when the triggering event is SHIPPED.request. If the service data element shipped-service-type is equal to LOAN then RETURN is set to TRUE; if the service data element shipped-service-type is equal to COPY/NON-RETURNABLE then RETURN is set to FALSE.

The requester sets the RETURN variable when the triggering event is RECEIVED.request. If the service data element shipped-service-type is equal to LOAN then RETURN is set to TRUE, if the service data element shipped-service-type is equal to COPY/NON-RETURNABLE then RETURN is set to FALSE.

8.2.3 Optional Messages

8.2.3.1 Simple Transactions

For a simple ILL-transaction, four of the ILL protocol actions are optional:

— send SHIPPED APDU;

— send RECEIVED APDU;

— send RETURNED APDU; and

— send CHECKED-IN APDU.

An application-entity-invocation may send optional APDUs whenever it wants to, and in addition is obliged to send them in certain situations.

The initiator of an ILL-transaction can inform the responder of what it is capable of supplying and what is requires in the way of optional messages within the ILL-REQUEST.

The ILL-REQUEST can specify:

a. Whether the requester is capable of sending RECEIVED.
b. Whether the requester is capable of sending RETURNED.
c. Whether the requester requires SHIPPED.
d. Whether the requester requires CHECKED-IN.
e. Whether the requester desires SHIPPED; this choice is meaningful only if choice c. above is NO.
f. Whether the requester desires CHECKED-IN; this choice is meaningful only if choice d. above is NO.

Correspondingly, the ILL-ANSWER and the SHIPPED can specify:

a. Whether the responder is capable of sending SHIPPED.
b. Whether the responder is capable of sending CHECKED-IN.
c. Whether the responder requires RECEIVED.
d. Whether the responder requires RETURNED.
e. Whether the responder desires RECEIVED; this choice is meaningful only if choice c. above is NO.
f. Whether the responder desires RETURNED; this choice is meaningful only if choice d. above is NO.

When a responder receives an ILL-REQUEST that indicates either:

a. that the requester cannot send a message that the responder requires, or
b. that the requester requires a message that the responder cannot send,

then the responder may send an ILL-ANSWER -UNFILLED. If a responder chooses to supply the requested item, it does so on the understanding that the RECEIVED and RETURNED messages will not be sent.

In all cases where a message is not required, it may or may not be sent, whether or not it is desired. The receipt of an optional APDU that was not requested is not a protocol error and, except for the SHIPPED APDU, does not cause a state change.

8.2.3.2 Chained and Partitioned Transactions

For chained and partitioned ILL-transactions, the SHIPPED APDU is mandatory, both between the responder and the intermediary and between the intermediary and the requester.

The APDUs RECEIVED, RETURNED and CHECKED-IN are optional. However, an intermediary, when acting as a requester, must be capable of generating RECEIVED and/or RETURNED if so required by the responder, and, when acting as responder, must be capable of generating CHECKED-IN if so required by the requester.

8.2.4 Send-to-list

The "send-to-list" identifies potential destinations for forwarding, chaining or partitioning. Each entry in the list specifies a responder-id and optionally an account number and a system-address. The ILL-transaction initiator may always supply entries for this list, may never fill it, or may rely on an intermediary to add entries. The protocol does not limit the number of entries in the send-to-list, nor does it disallow repeated entries, with the condition that such repeated entries cannot be used when forwarding (see 8.2.5).

Intermediaries may alter this list if "permission-to-change-send-to-list" is TRUE. Changes to this list may be in the form of additions or deletions.

The interpretation of this list is governed by the "preference" type.

The value "ordered" indicates that the order of preference for forwarding, etc., is that specified in the "send-to-list". Changes to the list, when permitted, can only be in the form of additions to the end of the list.

The value "unordered" indicates that any member of the list may be selected without preference. Changes to the list, when permitted, can be in any form.

8.2.5 Already-tried-list

This list identifies the institutions to which an ILL-request has already been sent. Such sites must be excluded from any subsequent forwarding, i.e. it is not permissible for a responder who has forwarded an ILL-request to receive a subsequent ILL-request with the same transaction-id. This list imposes no other constraints on ILL-transaction processing.

This list is updated with the responder's system-id whenever an ILL-request is forwarded, or a sub-transaction for a chained or partitioned ILL-transaction is initiated. In the original ILL-REQUEST data may be contained in the already-tried-list. Any addition to the list should be placed at the end of the list.

8.2.6 Control of Renewals

The SHIPPED, OVERDUE and RENEW-ANSWER APDUs with the same transaction-id all indicate whether a loan is renewable. This information is provided to the requester who is expected not to initiate a RENEW.request unless the item is renewable. It is not an error if a RENEW.request is made when the item is not renewable.

8.2.7 APDU Sequence Validation

With the exception of the MESSAGE, STATUS-QUERY, STATUS-OR-ERROR-REPORT and DAMAGED APDUs, the recipient validates all received APDUs with the same transaction-id and from the same originator for correct sequencing, based on the value of the type "date-time-of-this-service" in each APDU. The value of the "service date and time" parameter must be distinct for each service request made by the same party (requester, responder or intermediary) for a particular transaction.

This value is compared against the value of the protocol variable SEQUENCE-TIME-STAMP. If the SEQUENCE-TIME-STAMP has a value equal to or greater than the value in the received APDU, the received APDU causes no state change, the SEQUENCE-TIME-STAMP is not updated, and an indication service primitive is issued to the user. No checking is performed for repeated APDUs (see 8.2.8). If the SEQUENCE-TIME-STAMP has a value less than the value of the type "date-time-of-this-service" in the received APDU, the APDU is accepted, a state change is made if appropriate and the SEQUENCE-TIME-STAMP is updated. Sequence validation is performed after system validation and before any other processing by the recipient ILLPM.

Sequence validation is performed only for APDUs from the same originator. For the requester, and the intermediary requester, this is determined by comparing the protocol variable CURRENT-PARTNER-ID with a field in some or all APDUs that are received. In a simple transaction, the comparison is with the "responder-id" field in every APDU that is received, except in the case of the FORWARD-NOTIFICATION APDU where the comparison is with the “intermediary-id” field. In a chained or partitioned transaction, the comparison is with the "intermediary-id" field of any FORWARD-NOTIFICATION or SHIPPED APDU that is received. The CURRENT-PARTNER-ID protocol variable is given an initial value as described in 7.3. When an APDU arrives from a different responder, for example a SHIPPED indication from an institution to which an ILL-request has been forwarded, then no sequence check is performed and the PREVIOUS-PARTNER-IDS protocol variable is checked to determine whether the APDU arrived from a party already involved in the ILL-transaction. If the received APDU is from a new responder, then

a. the received APDU is treated as an in-sequence message and is processed accordingly;
b. the SEQUENCE-TIME-STAMP protocol variable is set to the time stamp of the received APDU; and
c. the CURRENT-PARTNER-ID value is added to PREVIOUS-PARTNER-IDS and is updated to reflect the value of "responder-id" in the received APDU.

If the responder is a previous one, then the received APDU is treated in the same way as an out-of-sequence one, and SEQUENCE-TIME-STAMP, CURRENT-PARTNER-ID and PREVIOUS-PARTNER-IDS are not updated.

For the responder, and the intermediary responder, sequence validation is performed by comparing, for each received APDU, the value of the field "requester-id" with the protocol variable CURRENT-PARTNER-ID. When an APDU arrives from a different requester, for example a RECEIVED indication from an institution to which an ILL-request has been partitioned, then no sequence check is performed and the PREVIOUS-PARTNER-IDS protocol variable is checked to determine whether the APDU arrived from a party already involved in the ILL-transaction. If the received APDU is from a new requester, then

a. the received APDU is treated as an in-sequence message and is processed accordingly;
b. the SEQUENCE-TIME-STAMP protocol variable is set to the time stamp of the received APDU; and
c. the CURRENT-PARTNER-ID value is added to PREVIOUS-PARTNER-IDS and is updated to reflect the value of "requester-id" in the received APDU.

If the requester is a previous one, then the received APDU is treated in the same way as an out-of-sequence one, and SEQUENCE-TIME-STAMP, CURRENT-PARTNER-ID and PREVIOUS-PARTNER-IDS are not updated.

Note that with the exception of the (ILL-REQUEST, CANCEL) sequence, the ILL protocol is resilient to all out-of-sequence combinations, in the sense that the second (out-of-sequence) APDU in the sequence will be accepted by the protocol even if it is received first. This avoids the need for special rules for each possible situation as long as the second APDU received never causes a state change.

8.2.8 Repeated APDUs

It is possible to repeat a particular service request one or more times without causing a protocol error. Examples of situations where a service request may need repeating include:

— an OVERDUE request where multiple overdue notices are sent before action is taken;
— an ILL-REQUEST, CANCEL, RENEW or CONDITIONAL request when no response to the preceding request has been received;
— an ILL-REQUEST, or any other request, when a problem was detected with the underlying communications service that may have prevented delivery of the corresponding APDU.

Only the most recent service request which caused a state change in the originating system can be repeated. Service requests which never cause a state change, i.e. MESSAGE, STATUS-QUERY, STATUS-OR-ERROR-REPORT and DAMAGED, are not repeated; each service request is a new one.

If an ILL-ANSWER(CONDITIONAL) is followed by CONDITIONAL-REPLY(YES), the responder may send another ILL-ANSWER(CONDITIONAL), with an additional condition. This is not considered to be a repeated service request.

A repeated service request is identified by providing a value for the type "date-time-of-original-service" when the repeated request is made. This date and time is that of the original request that is being repeated. When a service request is repeated, only the "service-date-time" and "note" parameters may have different values. No state change is effected in the system where the request repeat is initiated.

The recipient of a repeated APDU deals with it differently according to whether the original or a preceding repeated APDU was already received.

If no previous APDU has been received, as indicated by different values for "date-time-of-original-service" and the REPEAT-TIME-STAMP, then the APDU is treated as original with the corresponding state change and service indication. Also, the REPEAT-TIME-STAMP is updated to equal the "date-time-of-original-service". If a previous APDU has been received, as indicated by equality of the "date-time-of-original-service" and "REPEAT-TIME-STAMP", then no state change is made. However, an indication service primitive is issued to the service-user because of the possibility that the "note" field might have new information. The recipient of a repeated service indication shall repeat its earlier response, if one has already been made.

The mechanism whereby the decision is made to repeat a service request is outside the scope of this part of ISO 10161.

Note that the EXPIRY service cannot be repeated because it is a provider-initiated service.

8.2.9 Retries

When a previous ILL-transaction or sub-transaction has terminated with a transaction result of RETRY, UNFILLED, LOCATIONS-PROVIDED or ESTIMATE, it is possible to initiate a new transaction as an explicit retry at a later date. When the ILL-transaction is a retry of a previous one, the "Retry-Flag" of the ILL-REQUEST APDU is set to TRUE.

For the initial requester a retry is a new transaction, and so the ILL-transaction-qualifier must be different from that used in the original request but the ILL-transaction-group-qualifier must be the same (to enable the responder or intermediary to relate the retry to the previous ILL-transaction).

For an intermediary a retry is a new sub-transaction, and so the sub-transaction-qualifier must be different from that used in the original request, but both the ILL-transaction-group-qualifier and the ILL-transaction-qualifier must be the same (to enable the responder or next intermediary to relate the retry to the previous sub-transaction).

8.2.10 Transaction Expiry

The requester, at the time of an ILL-REQUEST, may choose to set a time limit on the lifetime of the ILL-transaction. This time limit is indicated in one of two ways:

a. by providing a value for the "need-before-date" type and setting the "expiry-flag" to "NEED-BEFORE-DATE"; or

b. by providing a value for the type "expiry-date" and setting the "expiry-flag" to "OTHER-DATE".

If either of these two conditions is satisfied, then the "EXPIRY" timer is set to the specified expiry date at the responder when the ILL-REQUEST APDU is received.

If no time limit is to be set for ILL-transaction expiry then "expiry-flag" is given the value "NO EXPIRY".

If no response (in the form of an ILL-Answer or Shipped service) is initiated before the value of the "EXPIRY" timer becomes equal to the current calendar date, then an EXPIRED APDU is sent by the responder to the requester. An EXPIRED.indication service primitive is issued at both the requester and responder and the ILL-transaction enters the NOT-SUPPLIED state.

NOTE 1 — These two possibilities for setting an expiry date for an ILL-transaction give the requester the flexibility of associating or not associating expiry semantics with the "need-before-date". An example of the use of an expiry date other than "need-before-date" would be to allow a shorter time to reply so that other potential responders could be contacted before the need-before-date.

NOTE 2 — In cases where the "expiry-flag" is set to "OTHER-DATE", it is still possible to provide a value for "need-before-date" but it has no expiry semantics.

NOTE 3 — As an implementation consideration to protect against the possible loss of an EXPIRED APDU, the requester may also maintain an expiry timer. If this timer should expire, this could trigger the user to send a STATUS-QUERY to the responder. Expiry of the requester's timer should not result in automatic expiry of the ILL-transaction, as the responder may not actually have timed out (e.g. a SHIPPED message may be in transit or may have been lost).

If an ILL-ANSWER.request with the result CONDITIONAL is issued by the responder, then the EXPIRY timer is reset to the value of the type "date-for-reply". If this timer expires, then the same actions described above take place. If no value is present for this type, the EXPIRY time is unaffected, i.e. disabled if previously disabled or value left intact if previously set.

NOTE — The "date-for-reply" in the ILL-ANSWER(CONDITIONAL) may be sooner than the date originally set for expiry by the requester.

If the responder receives a CONDITIONAL-REPLY with answer YES, then the EXPIRY timer is reset to its original value. The following events at the responder will disable the EXPIRY timer:

— ILL-ANSWER.request with result not equal to CONDITIONAL

— SHIPPED.request

— FORWARD.request

— receive CANCEL APDU

NOTE — If an ILL-ANSWER with the results WILL-SUPPLY or HOLD-PLACED is received with a date that is later than the "need-before-date" or "expiry-date" indicated on the original ILL-REQUEST, then it is up to the requester to decide whether to wait or to cancel the request.

8.2.11 Transaction Cancellation

A requester can initiate cancellation of an ILL-transaction at any time while in the PENDING state.

Once a responder has received a CANCEL.indication, it must respond with a CANCEL-REPLY.request, with the following exception.

If the responder has already issued a SHIPPED.request, FORWARD.request, or an ILL-ANSWER.request with a result of RETRY, UNFILLED, LOCATIONS-PROVIDED or ESTIMATE, or has received an EXPIRED.indication, then the CANCEL.indication is ignored and no CANCEL-REPLY.request is issued.

When the responder issues the CANCEL-REPLY.request with answer=YES, then the APDU CANCEL-REPLY is sent and the responder enters the terminal state CANCELLED. When the APDU CANCEL-REPLY is received, the requester issues a CANCEL-REPLY.indication and enters the terminal state CANCELLED.

When the responder issues the CANCEL-REPLY.request with answer=NO, then the APDU CANCEL-REPLY is sent and the responder enters the IN-PROCESS state. When the APDU CANCEL-REPLY is received, the requester issues a CANCEL-REPLY.indication and enters the PENDING state.

8.2.12 Lifetime of ILL-Transaction Information

The length of time that ILL-transaction information is maintained by a system once an ILL-transaction reaches a terminal state is a local matter that is outside the scope of this part of ISO 10161. The process whereby this information is made unavailable is termed transaction closure or transaction removal.

A system must respond to all STATUS-QUERY.requests. If the relevant ILL-transaction is not available, then the STATUS-OR-ERROR-REPORT will indicate one of the following two reasons:

— "information not available - temporary"

— "information not available - permanent"

The first condition is used to signal a temporary unavailability of information, for example due to a storage system failure. The second condition is used once an ILL-transaction has been closed.

8.2.13 Protocol Errors

Any events not listed in the protocol tables of annex A are not valid and are considered to be protocol errors. With the exception specified in 8.2.14, incorrectly formatted APDUs or APDUs with invalid data are also considered to be protocol errors.

When a protocol error is detected, no state change occurs and a STATUS-OR-ERROR-REPORT APDU describing the nature of the error is sent by the ILLPM.

If a protocol error is reported in response to an initial ILL-Request APDU, the requester may force the transaction to a terminal state such as CANCELLED or NOT-SUPPLIED. Alternatively the requester may re-issue the APDU, possibly in a different version.

8.2.14 Rules for Extensibility

All syntactical errors in received APDUs are considered to be protocol errors except for an unknown value of a known parameter (other than protocol version), which does not cause a protocol error.

An unknown value of protocol-version does constitute a protocol error.

8.2.15 Responder-specific Information

The ILL protocol makes allowance for conveying responder-specific information to describe a requested service or the result of an ILL request.

Responder-specific information may be provided to supplement the standardized values for ILL-service-type and results-explanation defined in this part of ISO 10161. In such cases, any standardized value other than "responder-specific" is used for ILL-service-type or result reason and the supplementary responder-specific information is provided in the separate type responder-specific-service or responder-specific-results, respectively.

Alternatively, responder-specific information may be provided to supplant the standardized values for ILL-service-type and results-explanation defined in this part of ISO 10161. In such cases, the standardized value "responder-specific" is used for ILL-service-type or result reason and the responder-specific information must be provided in the separate type responder-specific-service or responder-specific-results, respectively. The use of the ASN.1 type EXTERNAL to define responder-specific information is described in annex D.

8.2.16 Account-number Information

When a request is forwarded, chained or partitioned, the account-number in Cost-Info-Type is replaced by the requesting institution's account number with the next institution to which the request is sent as contained in the corresponding entry in the Send-to-List-Type. If no such entry is available, this field is left empty.

8.2.17 Supplemental-item-description

In addition to the item identification information identified in this part of ISO 10161, it is possible to provide supplemental information that is different either in nature or in format. For example, information could be provided in ISO 2709 format, or in bar code format for quick item identification. Supplemental item description can be supplied in the ILL-REQUEST, SHIPPED, ILL-ANSWER, RECEIVED and RETURNED requests.

This information may be provided by the initial-requester or added later by responders or intermediaries (e.g. as a result of bibliographic checking).

8.2.18 Send Message

This parameter is an abstract service parameter that does not result in a corresponding protocol parameter.

8.3 Procedural Rules for Intermediaries

All ILL-transactions are initially simple. When given permission by the requester, a responder may choose to forward, chain or partition an ILL-transaction, or it may establish distinct ILL-transactions.

Intermediaries are subject to the rules stated in this subclause, in addition to the rules specified in the preceding subclause.

8.3.1 Transaction Forwarding

The ILL protocol supports unlimited forwarding of an ILL-request, with the only constraint being that a request cannot be forwarded to the same responder twice within the same ILL-transaction group.

A responder can supply two different types of notes when forwarding an ILL-REQUEST: A forward-note in the forwarded ILL-REQUEST and a notification-note in the FORWARD-NOTIFICATION.

Once forwarding occurs, the responder no longer participates in the ILL-transaction. From that point, only the FORWARD, STATUS-REQUEST, STATUS-OR-ERROR-REPORT and MESSAGE services can be invoked by the responder. If, while in the PENDING state, a requester sends a repeated ILL-REQUEST, a CANCEL, MESSAGE or STATUS-QUERY APDU to the responder, and the responder has already forwarded the request to a new responder, the intermediary (first responder) will send a STATUS-OR-ERROR-REPORT with an error-report value of "Already-Forwarded" with information in the note field as to where the request was forwarded.

The Boolean protocol variable FWD is used by the responder to indicate whether or not a request may be forwarded. It is set upon receipt of an ILL-REQUEST APDU to the value conveyed by the type "permission-to-forward". If FWD is TRUE, then forwarding is permitted. If FWD is FALSE, then forwarding is not permitted.

The "permission-to-forward" type can also be set to FALSE at the time a request is forwarded when it is desirable to constrain further forwarding.

Forwarding may be mixed with ILL-transaction chaining and/or partitioning.

NOTE — Forwarding is not intended to be used when a requested item is expected to become available in the future; in such cases, an ILL-ANSWER with a value of RETRY and a specified retry date is the appropriate response.

8.3.2 Transaction Chaining

The ILL protocol supports unlimited chaining of an ILL-transaction.

Chained ILL-transactions are identified by the value of "transaction-type" in the ILL-REQUEST APDU.

When ILL-transaction chaining occurs, the responder becomes an intermediary and initiates a sub-transaction with a new responder. When the sub-transaction is initiated, the CHAIN variable remains TRUE while the PART variable is set to FALSE.

The intermediary takes on the role of requester for the sub-transaction and keeps the role of responder for the main ILL-transaction. The sub-transaction is distinguished from the main ILL-transaction so that the former can be unsuccessful without affecting the latter.

The intermediary ILL-service user is responsible for coordinating events on the main and the sub-transactions. Conceptually, this coupling is in terms of service primitives: a service primitive indication on one ILL-transaction is mapped onto the corresponding service primitive request on the other, e.g. the receipt of a SHIPPED.indication is mapped onto a SHIPPED.request to be sent to the requester.

The intermediary may alter the values of service parameters as it performs the mapping between main and sub-transactions, except where specifically disallowed. If it alters the value of the expiry timer, the expiry date for the sub-transaction must be equal to or less than the expiry date for the original request.

Coupling between the main and the sub-transaction is such that main ILL-transaction events are always mapped onto the corresponding sub-transaction event. Similarly, all sub-transaction events, except the ILL-ANSWER.indication with values RETRY, UNFILLED, or LOCATIONS-PROVIDED, the EXPIRED.indication and the FORWARD-NOTIFICATION.indication, are mapped onto the corresponding main ILL-transaction event.

In the case of an ILL-ANSWER.indication with one of the values specified above, it may be mapped onto a main ILL-transaction ILL-ANSWER.request, or it may not at the option of the intermediary. An example of the latter is the case where the intermediary, upon receipt of a negative response from one responder, chooses to initiate a new sub-transaction with another responder rather than report immediately a negative answer to the main ILL-transaction requester.

An ILL-ANSWER.indication with a value of CONDITIONAL is to be mapped onto the main ILL-transaction ILL-ANSWER.request.

A subsequent CONDITIONAL-REPLY with the value NO is to be mapped onto the sub-transaction. Note that in this case the intermediary may not initiate a new sub-transaction with another responder.

In the case of the FORWARD-NOTIFICATION.indication and EXPIRED. indication, no mapping to any main ILL-transaction event is performed. These service primitives have implications only for the sub-transaction. If an intermediary requester in the CANCEL-PENDING state receives a FORWARD-NOTIFICATION.indication from the responder and as a result returns to the PENDING state, it must issue a CANCEL.request to the new responder.

The intermediary participates in both the main and sub-transactions throughout their lifetime, but performs no state transitions during the tracking phase of such ILL-transactions, i.e. the intermediary acts in a pass-through mode. Within the intermediary, the SHIPPED state is the terminal state for both a successful main ILL-transaction and a successful sub-transaction. Subsequent state transitions only occur for the requester and responder. The state transition rules for the intermediary differ from those of the requester and responder in certain states. These intermediary specific rules are reflected in separate tables in annex A.

The Boolean protocol variable CHAIN is used to indicate whether or not an ILL-transaction can be chained. It is set upon receipt of an ILL-REQUEST APDU to the value conveyed by the type "permission-to-chain". If CHAIN is TRUE, then chaining is permitted. The intermediary may choose to modify the value of the "permission-to-chain" type when it initiates a sub-transaction.

When the intermediary initiates a sub-transaction, it always indicates a requirement for the SHIPPED message in the "requester-optional-messages" parameter of the ILL-REQUEST service. All other components of this parameter have the same value as supplied in the original ILL-REQUEST.

When the intermediary responds to the requester with the APDU SHIPPED, the "responder-optional-messages" parameter must take on the value provided in the SHIPPED APDU received from the responder.

8.3.3 Transaction Partitioning

The ILL protocol supports unlimited partitioning of an ILL-transaction.

Partitioned ILL-transactions are identified by the value of "transaction-type" in the ILL-REQUEST APDU.

When ILL-transaction partitioning occurs, the responder becomes an intermediary and initiates a sub-transaction with a new responder. When the sub-transaction is initiated, the PART variable remains TRUE while the CHAIN variable is set to FALSE.

The intermediary participates in the main ILL-transaction only during the processing phase, i.e. until a SHIPPED APDU is received from the responder and passed on to the requester. The sub-transaction has only a processing phase. Within the intermediary, the SHIPPED state is the terminal state for both a successful main ILL-transaction and a successful sub-transaction.

The tracking phase normally involves direct interaction between the requester and responder and by-passes the intermediary. However, if the intermediary receives a MESSAGE, STATUS-QUERY or STATUS-OR-ERROR-REPORT APDU after it is in a terminal state, the intermediary does not respond but passes on the APDU.

The identifiers of the main and sub-transactions differ only in the presence of a sub-transaction qualifier. Upon completion of the processing phase, the responder drops the sub-transaction qualifier from the ILL-transaction identifier (see 7.1) and uses the remainder when interacting with the requester.

During the processing phase, the main and sub-transactions are coupled in the same manner as chained ILL-transactions.

The Boolean protocol variable PART is used to indicate whether or not an ILL-transaction can be partitioned. It is set upon receipt of an ILL-REQUEST APDU to the value conveyed by the type "permission-to-partition". If PART is TRUE, then partitioning is permitted.

When the intermediary initiates a sub-transaction, it always indicates a requirement for the SHIPPED message in the "requester-optional-messages" parameter of the ILL-REQUEST service. All other components of this parameter have the same value as supplied in the original ILL-REQUEST.

When the intermediary responds to the requester with the APDU SHIPPED, the "responder-optional-messages" parameter must take on the value provided in the SHIPPED APDU received from the responder. The requester sends any such requested APDUs directly to the responder, not to the intermediary.

8.3.4 Mixed Forwarding, Chaining and Partitioning

Forwarding, chaining and partitioning may be mixed in any fashion within an ILL-transaction. The rules that apply differ according to the particular combination being used.

The following cases are distinguished:

a. **Chaining followed by forwarding:** the recipient of a forwarded ILL-REQUEST interacts throughout the ILL-transaction with the intermediary identified by the "requester-id" parameter of the received ILL-REQUEST.

b. **Forwarding followed by chaining:** the initiator of the chained sub-transaction interacts directly with the original-requester after it has received the forwarded ILL-REQUEST. The recipient of the chained ILL-REQUEST interacts with the chaining intermediary, as with a normal chained ILL-transaction.

c. **Partitioning followed by forwarding:** the recipient of a forwarded ILL-REQUEST interacts throughout the processing phase with the intermediary identified by the "requester-id" parameter of the received ILL-REQUEST. During the tracking phase, it interacts directly with the initial-requester.

d. **Forwarding followed by partitioning:** the initiator of the partitioned sub-transaction interacts directly with the original-requester during the processing phase, after it has received the forwarded ILL-REQUEST. The recipient of the partitioned ILL-REQUEST interacts with the partitioning intermediary during the processing phase, and with the initial-requester during the tracking phase, as with a normal chained ILL-transaction.

e. **Chaining followed by partitioning:** when partitioning follows chaining, the rules for partitioning dominate the rules for chaining. The recipient of the partitioned ILL-REQUEST interacts during the processing phase with the intermediary identified by the "requester-id" parameter of the received ILL-REQUEST. If this phase is completed with the sending of a SHIPPED APDU, then the "transaction type" parameter must be returned to the initial-requester with the value "partitioned" and not "chained". The value of the "responder-id" parameter shall be the identifier of the responder to which the request was partitioned. During the tracking phase, the initial-requester interacts directly with that responder.

f. **Partitioning followed by chaining:** Unlike the preceding case, the recipient to which the ILL-REQUEST is chained is not aware of the partitioning and therefore cannot amend its behaviour accordingly. It therefore never interacts directly with the initial-requester. The processing phase proceeds normally, with the exception of the handling of the SHIPPED APDU. When the chaining intermediary, i.e. the one to which the ILL-REQUEST was partitioned, receives the SHIPPED.indication from the final-responder before passing it on, it sets the "supplier-id" parameter, if absent, to the value of "responder-id" and sets "responder-id" to the identification of that intermediary. The "transaction-type" parameter is set to "partitioned". In this way, the SHIPPED.indication received by the initial-requester will indicate that subsequent interactions during the tracking phase shall be directly with the chaining intermediary. However, the "supplier-id" will identify the actual supplier of the item. The chaining intermediary remains responsible for chaining all interactions with the initial-requester on to the final-responder, and vice versa.

   Changes to the "transaction-type", "supplier-id" and "responder-id" parameters are made only when the "transaction-type" parameter does not already have the value "partitioned". This rule ensures that it is always the final intermediary in a sequence of partitionings that is identified to the initial-requester.

g. **Forwarding, chaining and partitioning:** No additional rules apply. The behaviour of any given combination is determined by the repeated application of the rules for the pairwise combinations stated above.

8.3.4.1 Distinct ILL-Transactions

When a system establishes distinct ILL-transactions as an alternative to acting as an intermediary, no specific rules of procedure for linking the distinct ILL-transactions are established by this part of ISO 10161.

Note, however, that such a dual-role system must employ some mechanism to track the progress of the two transactions to ensure that they reach terminal states; this could be achieved in various ways, for example:

a. For all items, either a SHIPPED APDU must be supplied by the responder or the item must be shipped via the intermediary.

b. For returnable items, either a CHECKED-IN APDU must be supplied by the responder or the item must be returned via the intermediary.

This style of operation, since it involves distinct simple ILL-transactions, has no protocol implications, and is not described further in this part of ISO 10161.

9 Abstract Syntax

9.1 ASN.1 Specification of ILL APDUs

This clause describes the abstract syntax of the ILL APDUs defined within the ILL protocol as listed in clause 6. The ILL APDUs are defined using the ASN.1 notation defined in ISO 8824 and its Addendum 1.

Each APDU is defined as a structured type where a type is a named set of values. A structured type is defined by reference to one or more other types which may themselves be structured types or simple types. A simple type is defined by directly specifying the set of its values.

When defining the structured types, ASN.1 specifies which of the component types making up the structured type are optional, and which are mandatory, and the allowable values of the types.

Other information, for example whether a string is fixed or variable in size, and default values are also provided.

The comments included within the ASN.1 specification constitute part of the standard.

Some of the optional types are given default values. If a value for the type does not exist within the encoded APDU, the default value is assumed. If a type does not have a default value, and a value for that type does not exist within the encoded APDU, then no value is associated with that type.

If a structured type is mandatory, but is made up only of optional component types, then at least one of the optional types must be present, e.g. system-id.

If a structured type is optional, but is constructed from a mandatory type, then the component type is mandatory only if the structured type is present, e.g. supply-medium-info-type.

The ILL APDUs are defined in 9.1.1 as structured types. Subclause 9.1.2 lists the types from which these structured types are constructed and have not been defined in 9.1.1.

9.1.1 ILL APDUs

```
-- use of parameters and types is to be consistent with their definitions in clause 3

ISO-10161-ILL-1 DEFINITIONS EXPLICIT TAGS ::=

BEGIN
-- ISO-10161-ILL-1 refers to the ILL ISO standard 10161 version 1
-- and version 2 as specified in ISO standard 10161 Amendment 1

ILL-APDU ::= CHOICE {
    ILL-Request,
    Forward-Notification,
    Shipped,
    ILL-Answer,
    Conditional-Reply,
    Cancel,
    Cancel-Reply,
    Received,
    Recall,
    Returned,
    Checked-In,
    Overdue,
    Renew,
    Renew-Answer,
    Lost,
    Damaged,
    Message,
    Status-Query,
    Status-Or-Error-Report,
    Expired
    }

-- The tagging scheme used in the APDU definitions is as follows:
-- each named type that is a component type of an APDU definition is assigned a different tag
-- that is unique within the set of APDU definitions.  Where these component types themselves
-- have structure, the tagging within such type definitions has only local scope.

ILL-Request ::= [APPLICATION 1] SEQUENCE {
    protocol-version-num        [0]  IMPLICIT INTEGER {
                                        version-1 (1),
                                        version-2 (2)
                                        },
    transaction-id              [1]  IMPLICIT Transaction-Id,
    service-date-time           [2]  IMPLICIT Service-Date-Time,
    requester-id                [3]  IMPLICIT System-Id OPTIONAL,
                                -- mandatory when using store-and-forward communications
                                -- optional when using connection-oriented communications
    responder-id                [4]  IMPLICIT System-Id OPTIONAL,
                                -- mandatory when using store-and-forward communications
                                -- optional when using connection-oriented communications
    transaction-type            [5]  IMPLICIT Transaction-Type DEFAULT 1,
    delivery-address            [6]  IMPLICIT Delivery-Address OPTIONAL,
    delivery-service                 Delivery-Service OPTIONAL,
    billing-address             [8]  IMPLICIT Delivery-Address OPTIONAL,
    iLL-service-type            [9]  IMPLICIT SEQUENCE SIZE (1..5) OF
                                     ILL-Service-Type,
                                -- this sequence is a list, in order of preference
    responder-specific-service  [10] EXTERNAL OPTIONAL,
                                -- use direct reference style
    requester-optional-messages [11] IMPLICIT Requester-Optional-Messages-Type,
    search-type                 [12] IMPLICIT Search-Type OPTIONAL,
    supply-medium-info-type     [13] IMPLICIT SEQUENCE SIZE (1..7) OF
```

```
                                        Supply-Medium-Info-Type OPTIONAL,
                                    -- this sequence is a list, in order of preference,
                                    -- with a maximum number of 7 entries
    place-on-hold                   [14] IMPLICIT Place-On-Hold-Type DEFAULT 3,
    client-id                       [15] IMPLICIT Client-Id OPTIONAL,
    item-id                         [16] IMPLICIT Item-Id,
    supplemental-item-description   [17] IMPLICIT Supplemental-Item-Description OPTIONAL,
    cost-info-type                  [18] IMPLICIT Cost-Info-Type OPTIONAL,
    copyright-compliance            [19] ILL-String OPTIONAL,
    third-party-info-type           [20] IMPLICIT Third-Party-Info-Type OPTIONAL,
                                    -- mandatory when initiating a FORWARD service or an
                                    -- ILL-REQUEST service for a partitioned ILL sub-
                                    -- transaction or when initiating an ILL-REQUEST service for
                                    -- an ILL sub-transaction if the received ILL-REQUEST
                                    -- included an "already-tried-list";optional otherwise
    retry-flag                      [21] IMPLICIT BOOLEAN DEFAULT FALSE,
    forward-flag                    [22] IMPLICIT BOOLEAN DEFAULT FALSE,
    requester-note                  [46] ILL-String OPTIONAL,
    forward-note                    [47] ILL-String OPTIONAL,
    iLL-request-extensions          [49] IMPLICIT SEQUENCE OF Extension OPTIONAL
    }

Forward-Notification ::= [APPLICATION 2] SEQUENCE {
    protocol-version-num            [0]  IMPLICIT INTEGER {
                                            version-1 (1),
                                            version-2 (2)
                                            },
    transaction-id                  [1]  IMPLICIT Transaction-Id,
    service-date-time               [2]  IMPLICIT Service-Date-Time,
    requester-id                    [3]  IMPLICIT System-Id OPTIONAL,
                                    -- mandatory when using store-and-forward communications
                                    -- optional when using connection-oriented communications
    responder-id                    [4]  IMPLICIT System-Id,
                                    -- mandatory in this APDU
    responder-address               [24] IMPLICIT System-Address OPTIONAL,
    intermediary-id                 [25] IMPLICIT System-Id,
    notification-note               [48] ILL-String OPTIONAL,
    forward-notification-extensions [49] IMPLICIT SEQUENCE OF Extension OPTIONAL
    }

Shipped ::= [APPLICATION 3] SEQUENCE {
    protocol-version-num            [0]  IMPLICIT INTEGER {
                                            version-1 (1),
                                            version-2 (2)
                                            },
    transaction-id                  [1]  IMPLICIT Transaction-Id,
    service-date-time               [2]  IMPLICIT Service-Date-Time,
    requester-id                    [3]  IMPLICIT System-Id OPTIONAL,
                                    -- mandatory when using store-and-forward communications
                                    -- optional when using connection-oriented communications
    responder-id                    [4]  IMPLICIT System-Id OPTIONAL,
                                    -- mandatory when using store-and-forward communications
                                    -- optional when using connection-oriented communications
    responder-address               [24] IMPLICIT System-Address OPTIONAL,
    intermediary-id                 [25] IMPLICIT System-Id OPTIONAL,
    supplier-id                     [26] IMPLICIT System-Id OPTIONAL,
    client-id                       [15] IMPLICIT Client-Id OPTIONAL,
    transaction-type                [5]  IMPLICIT Transaction-Type DEFAULT 1,
    supplemental-item-description   [17] IMPLICIT Supplemental-Item-Description OPTIONAL,
    shipped-service-type            [27] IMPLICIT Shipped-Service-Type,
    responder-optional-messages     [28] IMPLICIT Responder-Optional-Messages-Type
                                            OPTIONAL,
    supply-details                  [29] IMPLICIT Supply-Details,
```

```
        return-to-address               [30] IMPLICIT Postal-Address OPTIONAL,
        responder-note                  [46] ILL-String OPTIONAL,
        shipped-extensions              [49] IMPLICIT SEQUENCE OF Extension OPTIONAL
        }

ILL-Answer ::= [APPLICATION 4] SEQUENCE {
        protocol-version-num            [0]  IMPLICIT INTEGER {
                                                version-1 (1),
                                                version-2 (2)
                                                },
        transaction-id                  [1]  IMPLICIT Transaction-Id,
        service-date-time               [2]  IMPLICIT Service-Date-Time,
        requester-id                    [3]  IMPLICIT System-Id OPTIONAL,
                                        -- mandatory when using store-and-forward communications
                                        -- optional when using connection-oriented communications
        responder-id                    [4]  IMPLICIT System-Id OPTIONAL,
                                        -- mandatory when using store-and-forward communications
                                        -- optional when using connection-oriented communications
        transaction-results             [31] IMPLICIT Transaction-Results,
        results-explanation             [32] CHOICE {
                                                conditional-results     [1] Conditional-Results,
                                                -- chosen if transaction-results=CONDITIONAL
                                                retry-results           [2] Retry-Results,
                                                -- chosen if transaction-results=RETRY
                                                unfilled-results        [3] Unfilled-Results,
                                                --chosen if transaction-results=UNFILLED
                                                locations-results       [4] Locations-Results,
                                                -- chosen if transaction-results=LOCATIONS-PROVIDED
                                                will-supply-results     [5] Will-Supply-Results,
                                                -- chosen if transaction-results=WILL-SUPPLY
                                                hold-placed-results     [6] Hold-Placed-Results,
                                                -- chosen if transaction-results=HOLD-PLACED
                                                estimate-results        [7] Estimate-Results
                                                -- chosen if transaction-results=ESTIMATE
                                                } OPTIONAL,
                                        -- optional if transaction-results equals RETRY, UNFILLED,
                                        -- WILL-SUPPLY or HOLD-PLACED;
                                        -- required if transaction-results equals CONDITIONAL,
                                        -- LOCATIONS-PROVIDED or ESTIMATE
        responder-specific-results      [33] EXTERNAL OPTIONAL,
                                        -- this type is mandatory if results-explanation
                                        -- chosen for any result
                                        -- has the value "responder-specific".
        supplemental-item-description   [17] IMPLICIT Supplemental-Item-Description OPTIONAL,
        send-to-list                    [23] IMPLICIT Send-To-List-Type OPTIONAL,
        already-tried-list              [34] IMPLICIT Already-Tried-List-Type OPTIONAL,
        responder-optional-messages     [28] IMPLICIT Responder-Optional-Messages-Type
                                             OPTIONAL,
        responder-note                  [46] ILL-String OPTIONAL,
        ill-answer-extensions           [49] IMPLICIT SEQUENCE OF Extension OPTIONAL
        }

Conditional-Reply ::= [APPLICATION 5] SEQUENCE {
        protocol-version-num            [0]  IMPLICIT INTEGER {
                                                version-1 (1),
                                                version-2 (2)
                                                },
        transaction-id                  [1]  IMPLICIT Transaction-Id,
        service-date-time               [2]  IMPLICIT Service-Date-Time,
        requester-id                    [3]  IMPLICIT System-Id OPTIONAL,
                                        -- mandatory when using store-and-forward communications
                                        -- optional when using connection-oriented communications
        responder-id                    [4]  IMPLICIT System-Id OPTIONAL,
```

```
                                    -- mandatory when using store-and-forward communications
                                    -- optional when using connection-oriented communications
        answer                      [35] IMPLICIT BOOLEAN,
        requester-note              [46] ILL-String OPTIONAL,
        conditional-reply-extensions [49] IMPLICIT SEQUENCE OF Extension OPTIONAL
        }

    Cancel ::= [APPLICATION 6] SEQUENCE {
        protocol-version-num        [0]  IMPLICIT INTEGER {
                                           version-1 (1),
                                           version-2 (2)
                                           },
        transaction-id              [1]  IMPLICIT Transaction-Id,
        service-date-time           [2]  IMPLICIT Service-Date-Time,
        requester-id                [3]  IMPLICIT System-Id OPTIONAL,
                                    -- mandatory when using store-and-forward communications
                                    -- optional when using connection-oriented communications
        responder-id                [4]  IMPLICIT System-Id OPTIONAL,
                                    -- mandatory when using store-and-forward communications
                                    -- optional when using connection-oriented communications
        requester-note              [46] ILL-String OPTIONAL,
        cancel-extensions           [49] IMPLICIT SEQUENCE OF Extension OPTIONAL
        }

    Cancel-Reply ::= [APPLICATION 7] SEQUENCE {
        protocol-version-num        [0]  IMPLICIT INTEGER {
                                           version-1 (1),
                                           version-2 (2)
                                           },
        transaction-id              [1]  IMPLICIT Transaction-Id,
        service-date-time           [2]  IMPLICIT Service-Date-Time,
        requester-id                [3]  IMPLICIT System-Id OPTIONAL,
                                    -- mandatory when using store-and-forward communications
                                    -- optional when using connection-oriented communications
        responder-id                [4]  IMPLICIT System-Id OPTIONAL,
                                    -- mandatory when using store-and-forward communications
                                    -- optional when using connection-oriented communications
        answer                      [35] IMPLICIT BOOLEAN,
        responder-note              [46] ILL-String OPTIONAL,
        cancel-reply-extensions     [49] IMPLICIT SEQUENCE OF Extension OPTIONAL
        }

    Received ::= [APPLICATION 8] SEQUENCE {
        protocol-version-num        [0]  IMPLICIT INTEGER {
                                           version-1 (1),
                                           version-2 (2)
                                           },
        transaction-id              [1]  IMPLICIT Transaction-Id,
        service-date-time           [2]  IMPLICIT Service-Date-Time,
        requester-id                [3]  IMPLICIT System-Id OPTIONAL,
                                    -- mandatory when using store-and-forward communications
                                    -- optional when using connection-oriented communications
        responder-id                [4]  IMPLICIT System-Id OPTIONAL,
                                    -- mandatory when using store-and-forward communications
                                    -- optional when using connection-oriented communications
        supplier-id                 [26] IMPLICIT System-Id OPTIONAL,
        supplemental-item-description [17] IMPLICIT Supplemental-Item-Description OPTIONAL,
        date-received               [36] IMPLICIT ISO-Date,
        shipped-service-type        [27] IMPLICIT Shipped-Service-Type,
        requester-note              [46] ILL-String OPTIONAL,
        received-extensions         [49] IMPLICIT SEQUENCE OF Extension OPTIONAL
        }
```

```
Recall ::= [APPLICATION 9] SEQUENCE {
    protocol-version-num        [0]  IMPLICIT INTEGER {
                                         version-1 (1),
                                         version-2 (2)
                                         },
    transaction-id              [1]  IMPLICIT Transaction-Id,
    service-date-time           [2]  IMPLICIT Service-Date-Time,
    requester-id                [3]  IMPLICIT System-Id OPTIONAL,
                                -- mandatory when using store-and-forward communications
                                -- optional when using connection-oriented communications
    responder-id                [4]  IMPLICIT System-Id OPTIONAL,
                                -- mandatory when using store-and-forward communications
                                -- optional when using connection-oriented communications
    responder-note              [46] ILL-String OPTIONAL,
    recall-extensions           [49] IMPLICIT SEQUENCE OF Extension OPTIONAL
    }

Returned ::= [APPLICATION 10] SEQUENCE {
    protocol-version-num        [0]  IMPLICIT INTEGER {
                                         version-1 (1),
                                         version-2 (2)
                                        },
    transaction-id              [1]  IMPLICIT Transaction-Id,
    service-date-time           [2]  IMPLICIT Service-Date-Time,
    requester-id                [3]  IMPLICIT System-Id OPTIONAL,
                                -- mandatory when using store-and-forward communications
                                -- optional when using connection-oriented communications
    responder-id                [4]  IMPLICIT System-Id OPTIONAL,
                                -- mandatory when using store-and-forward communications
                                -- optional when using connection-oriented communications
    supplemental-item-description [17] IMPLICIT Supplemental-Item-Description OPTIONAL,
    date-returned               [37] IMPLICIT ISO-Date,
    returned-via                [38] Transportation-Mode OPTIONAL,
    insured-for                 [39] IMPLICIT Amount OPTIONAL,
    requester-note              [46] ILL-String OPTIONAL,
    returned-extensions         [49] IMPLICIT SEQUENCE OF Extension OPTIONAL
    }

Checked-In ::= [APPLICATION 11] SEQUENCE {
    protocol-version-num        [0]  IMPLICIT INTEGER {
                                         version-1 (1),
                                         version-2 (2)
                                         },
    transaction-id              [1]  IMPLICIT Transaction-Id,
    service-date-time           [2]  IMPLICIT Service-Date-Time,
    requester-id                [3]  IMPLICIT System-Id OPTIONAL,
                                -- mandatory when using store-and-forward communications
                                -- optional when using connection-oriented communications
    responder-id                [4]  IMPLICIT System-Id OPTIONAL,
                                -- mandatory when using store-and-forward communications
                                -- optional when using connection-oriented communications
    date-checked-in             [40] IMPLICIT ISO-Date,
    responder-note              [46] ILL-String OPTIONAL,
    checked-in-extensions       [49] IMPLICIT SEQUENCE OF Extension OPTIONAL
    }

Overdue ::= [APPLICATION 12] SEQUENCE {
    protocol-version-num        [0]  IMPLICIT INTEGER {
                                         version-1 (1),
                                         version-2 (2)
                                         },
    transaction-id              [1]  IMPLICIT Transaction-Id,
    service-date-time           [2]  IMPLICIT Service-Date-Time,
```

```
298         requester-id                [3]  IMPLICIT System-Id OPTIONAL,
299                                     -- mandatory when using store-and-forward communications
300                                     -- optional when using connection-oriented communications
301         responder-id                [4]  IMPLICIT System-Id OPTIONAL,
302                                     -- mandatory when using store-and-forward communications
303                                     -- optional when using connection-oriented communications
304         date-due                    [41] IMPLICIT Date-Due,
305         responder-note              [46] ILL-String OPTIONAL,
306         overdue-extensions          [49] SEQUENCE OF Extension OPTIONAL
307         }

308     Renew ::= [APPLICATION 13] SEQUENCE {
309         protocol-version-num        [0]  IMPLICIT INTEGER {
310                                              version-1 (1),
311                                              version-2 (2)
312                                              },
313         transaction-id              [1]  IMPLICIT Transaction-Id,
314         service-date-time           [2]  IMPLICIT Service-Date-Time,
315         requester-id                [3]  IMPLICIT System-Id OPTIONAL,
316                                     -- mandatory when using store-and-forward communications
317                                     -- optional when using connection-oriented communications
318         responder-id                [4]  IMPLICIT System-Id OPTIONAL,
319                                     -- mandatory when using store-and-forward communications
320                                     -- optional when using connection-oriented communications
321         desired-due-date            [42] IMPLICIT ISO-Date OPTIONAL,
322         requester-note              [46] ILL-String OPTIONAL,
323         renew-extensions            [49] IMPLICIT SEQUENCE OF Extension OPTIONAL
324         }

325     Renew-Answer ::= [APPLICATION 14] SEQUENCE {
326         protocol-version-num        [0]  IMPLICIT INTEGER {
327                                              version-1 (1),
328                                              version-2 (2)
329                                              },
330         transaction-id              [1]  IMPLICIT Transaction-Id,
331         service-date-time           [2]  IMPLICIT Service-Date-Time,
332         requester-id                [3]  IMPLICIT System-Id OPTIONAL,
333                                     -- mandatory when using store-and-forward communications
334                                     -- optional when using connection-oriented communications
335         responder-id                [4]  IMPLICIT System-Id OPTIONAL,
336                                     -- mandatory when using store-and-forward communications
337                                     -- optional when using connection-oriented communications
338         answer                      [35] IMPLICIT BOOLEAN,
339         date-due                    [41] IMPLICIT Date-Due OPTIONAL,
340         responder-note              [46] ILL-String OPTIONAL,
341         renew-answer-extensions     [49] IMPLICIT SEQUENCE OF Extension OPTIONAL
342         }

343     Lost ::= [APPLICATION 15] SEQUENCE {
344         protocol-version-num        [0]  IMPLICIT INTEGER {
345                                              version-1 (1),
346                                              version-2 (2)
347                                              },
348         transaction-id              [1]  IMPLICIT Transaction-Id,
349         service-date-time           [2]  IMPLICIT Service-Date-Time,
350         requester-id                [3]  IMPLICIT System-Id OPTIONAL,
351                                     -- mandatory when using store-and-forward communications
352                                     -- optional when using connection-oriented communications
353         responder-id                [4]  IMPLICIT System-Id OPTIONAL,
354                                     -- mandatory when using store-and-forward communications
355                                     -- optional when using connection-oriented communications
356         note                        [46] ILL-String OPTIONAL,
357         lost-extensions             [49] IMPLICIT SEQUENCE OF Extension OPTIONAL
```

```
358         }

359     Damaged ::= [APPLICATION 16] SEQUENCE {
360         protocol-version-num        [0]  IMPLICIT INTEGER {
361                                             version-1 (1),
362                                             version-2 (2)
363                                             },
364         transaction-id              [1]  IMPLICIT Transaction-Id,
365         service-date-time           [2]  IMPLICIT Service-Date-Time,
366         requester-id                [3]  IMPLICIT System-Id OPTIONAL,
367                                     -- mandatory when using store-and-forward communications
368                                     -- optional when using connection-oriented communications
369         responder-id                [4]  IMPLICIT System-Id OPTIONAL,
370                                     -- mandatory when using store-and-forward communications
371                                     -- optional when using connection-oriented communications
372         damaged-details             [51] IMPLICIT Damaged-Details OPTIONAL,
373                                     -- this parameter may only be present in APDU's with a
374                                     -- protocol-version-num value of 2 or greater
375         note                        [46] ILL-String OPTIONAL,
376         damaged-extensions          [49] IMPLICIT SEQUENCE OF Extension OPTIONAL
377         }

378     Message ::= [APPLICATION 17] SEQUENCE {
379         protocol-version-num        [0]  IMPLICIT INTEGER {
380                                             version-1 (1),
381                                             version-2 (2)
382                                             },
383         transaction-id              [1]  IMPLICIT Transaction-Id,
384         service-date-time           [2]  IMPLICIT Service-Date-Time,
385         requester-id                [3]  IMPLICIT System-Id OPTIONAL,
386                                     -- mandatory when using store-and-forward communications
387                                     -- optional when using connection-oriented communications
388         responder-id                [4]  IMPLICIT System-Id OPTIONAL,
389                                     -- mandatory when using store-and-forward communications
390                                     -- optional when using connection-oriented communications
391         note                        [46] ILL-String,
392         message-extensions          [49] IMPLICIT SEQUENCE OF Extension OPTIONAL
393         }

394     Status-Query ::= [APPLICATION 18] SEQUENCE {
395         protocol-version-num        [0]  IMPLICIT INTEGER {
396                                             version-1 (1),
397                                             version-2 (2)
398                                             },
399         transaction-id              [1]  IMPLICIT Transaction-Id,
400         service-date-time           [2]  IMPLICIT Service-Date-Time,
401         requester-id                [3]  IMPLICIT System-Id OPTIONAL,
402                                     -- mandatory when using store-and-forward communications
403                                     -- optional when using connection-oriented communications
404         responder-id                [4]  IMPLICIT System-Id OPTIONAL,
405                                     -- mandatory when using store-and-forward communications
406                                     -- optional when using connection-oriented communications
407         note                        [46] ILL-String OPTIONAL,
408         status-query-extensions     [49] IMPLICIT SEQUENCE OF Extension OPTIONAL
409         }

410     Status-Or-Error-Report ::= [APPLICATION 19] SEQUENCE {
411         protocol-version-num        [0]  IMPLICIT INTEGER {
412                                             version-1 (1),
413                                             version-2 (2)
414                                             },
415         transaction-id              [1]  IMPLICIT Transaction-Id,
416         service-date-time           [2]  IMPLICIT Service-Date-Time,
```

```
        requester-id                    [3]  IMPLICIT System-Id OPTIONAL,
                                        -- mandatory when using store-and-forward communications
                                        -- optional when using connection-oriented communications
        responder-id                    [4]  IMPLICIT System-Id OPTIONAL,
                                        -- mandatory when using store-and-forward communications
                                        -- optional when using connection-oriented communications
        reason-no-report                [43] IMPLICIT Reason-No-Report OPTIONAL,
                                        -- mandatory if no report is present;
                                        -- not present otherwise
        status-report                   [44] IMPLICIT Status-Report OPTIONAL,
        error-report                    [45] IMPLICIT Error-Report OPTIONAL,
        note                            [46] ILL-String OPTIONAL,
        status-or-error-report-extensions [49] IMPLICIT SEQUENCE OF Extension OPTIONAL
        }

    Expired ::= [APPLICATION 20] SEQUENCE {
        protocol-version-num            [0]  IMPLICIT INTEGER {
                                                version-1 (1),
                                                version-2 (2)
                                                },
        transaction-id                  [1]  IMPLICIT Transaction-Id,
        service-date-time               [2]  IMPLICIT Service-Date-Time,
        requester-id                    [3]  IMPLICIT System-Id OPTIONAL,
                                        -- mandatory when using store-and-forward communications
                                        -- optional when using connection-oriented communications
        responder-id                    [4]  IMPLICIT System-Id OPTIONAL,
                                        -- mandatory when using store-and-forward communications
                                        -- optional when using connection-oriented communications
        expired-extensions              [49] IMPLICIT SEQUENCE OF Extension OPTIONAL
        }
```

9.1.2 Types

```
    Account-Number ::= ILL-String

    Already-Forwarded ::= SEQUENCE {
        responder-id                    [0]  IMPLICIT System-Id,
        responder-address               [1]  IMPLICIT System-Address OPTIONAL
        }

    Already-Tried-List-Type ::= SEQUENCE OF System-Id

    Amount ::= SEQUENCE {
        currency-code                   [0]  IMPLICIT PrintableString (SIZE (3)) OPTIONAL,
                                        -- values defined in ISO 4217-1981
        monetary-value                  [1]  IMPLICIT AmountString (SIZE (1..10))
        }

    AmountString ::= PrintableString (FROM ("1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9"|"0"|" "|"."|","))

    Client-Id ::= SEQUENCE {
        client-name                     [0]  ILL-String OPTIONAL,
        client-status                   [1]  ILL-String OPTIONAL,
        client-identifier               [2]  ILL-String OPTIONAL
        }

    Conditional-Results ::= SEQUENCE {
        conditions                      [0]  IMPLICIT ENUMERATED {
                                                cost-exceeds-limit              (13),
                                                charges                         (14),
                                                prepayment-required             (15),
                                                lacks-copyright-compliance      (16),
                                                library-use-only                (22),
```

```
                                    no-reproduction                         (23),
                                    client-signature-required               (24),
                                    special-collections-supervision-required (25),
                                    other                                   (27),
                                    responder-specific                      (28),
                                    proposed-delivery-service               (30)
                                    },
        date-for-reply              [1]  IMPLICIT ISO-Date OPTIONAL,
        locations                   [2]  IMPLICIT SEQUENCE OF Location-Info OPTIONAL,
        proposed-delivery-service        Delivery-Service OPTIONAL
                                    -- this parameter specifies a proposed delivery service the
                                    -- acceptance of which is a condition of supply.  It may be a
                                    -- physical service or an electronic service.  This parameter
                                    -- may only be present in APDUs with a
                                    -- protocol-version-num value of 2 or greater
        }

    Cost-Info-Type ::= SEQUENCE {
        account-number              [0]  Account-Number OPTIONAL,
        maximum-cost                [1]  IMPLICIT Amount OPTIONAL,
        reciprocal-agreement        [2]  IMPLICIT BOOLEAN DEFAULT FALSE,
        will-pay-fee                [3]  IMPLICIT BOOLEAN DEFAULT FALSE,
        payment-provided            [4]  IMPLICIT BOOLEAN DEFAULT FALSE
        }

    Current-State ::= ENUMERATED {
        nOT-SUPPLIED                (1),
        pENDING                     (2),
        iN-PROCESS                  (3),
        fORWARD                     (4),
        cONDITIONAL                 (5),
        cANCEL-PENDING              (6),
        cANCELLED                   (7),
        sHIPPED                     (8),
        rECEIVED                    (9),
        rENEW-PENDING               (10),
        nOT-RECEIVED-OVERDUE        (11),
        rENEW-OVERDUE               (12),
        oVERDUE                     (13),
        rETURNED                    (14),
        cHECKED-IN                  (15),
        rECALL                      (16),
        lOST                        (17),
        uNKNOWN                     (18)
        }

    Damaged-Details ::= SEQUENCE {
        document-type-id            [0]  IMPLICIT OBJECT IDENTIFIER OPTIONAL,
                                    -- identifies an OSI document type registered in accordance
                                    -- with ISO 9834-2, for use in an automated environment
        damaged-portion                  CHOICE {
                                      complete-document [1]        IMPLICIT NULL,
                                      specific-units    [2]        IMPLICIT SEQUENCE
                                                                       OF INTEGER
                                      -- the nature and extent of a "unit" is implicit in the
                                      -- value of document-type-id if one is supplied
                                      }
        }

    Date-Due ::= SEQUENCE {
        date-due-field              [0]  IMPLICIT ISO-Date,
        renewable                   [1]  IMPLICIT BOOLEAN DEFAULT TRUE
        }
```

```
Delivery-Address ::= SEQUENCE {
    postal-address              [0]  IMPLICIT Postal-Address OPTIONAL,
    electronic-address          [1]  IMPLICIT System-Address OPTIONAL
    }

Delivery-Service ::= CHOICE {
    physical-delivery           [7]  Transportation-Mode,
    electronic-delivery         [50] IMPLICIT SEQUENCE OF Electronic-Delivery-Service
                                -- electronic-delivery may only be present in APDUs
                                -- with a protocol-version-num value of 2 or greater
    }

Electronic-Delivery-Service ::= SEQUENCE {
    -- the first four parameters are intended to be used in an automated
    -- environment
    e-delivery-service          [0]  IMPLICIT SEQUENCE {
                                -- identifies the kind of electronic delivery service, e.g.
                                -- MOTIS IPM,FTAM, etc., using the assigned object
                                -- identifier for the standard e.g. {joint-iso-ccitt mhs-motis
                                -- ipms}
                                    e-delivery-mode         [0] IMPLICIT OBJECT IDENTIFIER,
                                    e-delivery-parameters   [1] ANY DEFINED BY e-delivery-mode
                                    } OPTIONAL,
    document-type               [1]  IMPLICIT SEQUENCE {
                                    document-type-id    [2] IMPLICIT OBJECT IDENTIFIER,
                                -- identifies an OSI document type registered in accordance
                                -- with ISO 9834-2
                                    document-type-parameters    [3] ANY DEFINED BY
                                    document-type-id } OPTIONAL,
                                -- any parameters relating to the registered document type
    e-delivery-description      [4]  ILL-String OPTIONAL,
                                -- holds a human readable name or description of the
                                -- required electronic delivery service and document type;
                                -- this may also be used to identify an electronic delivery
                                -- service for which there is no object identifier.
                                -- This parameter may be present instead of, or in addition
                                -- to, the previous 4 parameters
    e-delivery-details          [5]  CHOICE {
                                    e-delivery-address  [0] IMPLICIT System-Address,
                                    e-delivery-id       [1] IMPLICIT System-Id
                                    },
    name-or-code                [6]  ILL-String OPTIONAL,
                                -- holds a human-readable identifier or correlation
                                -- information for the document as shipped, e.g. a directory
                                -- and/or file name or message-id
    delivery-time               [7]  IMPLICIT ISO-Time OPTIONAL
                                -- holds the requester's preferred delivery time or
                                -- the responder's proposed or actual delivery time
    }

Error-Report ::= SEQUENCE {
    correlation-information     [0]  ILL-String,
    report-source               [1]  IMPLICIT Report-Source,
    user-error-report           [2]  User-Error-Report OPTIONAL,
                                -- mandatory if report-source is "user"; not present otherwise
    provider-error-report       [3]  Provider-Error-Report OPTIONAL
                                -- mandatory if report-source is "provider"; not
                                -- present otherwise
    }

Estimate-Results ::= SEQUENCE {
    cost-estimate               [0]  ILL-String,
    locations                   [1]  IMPLICIT SEQUENCE OF Location-Info OPTIONAL
```

```
588         }

589     Extension ::= SEQUENCE {
590         identifier                      [0]  IMPLICIT INTEGER,
591         critical                        [1]  IMPLICIT BOOLEAN DEFAULT FALSE,
592         item                            [2]  ANY DEFINED BY identifier
593         }

594     General-Problem ::= ENUMERATED {
595         unrecognized-APDU               (1),
596         mistyped-APDU                   (2),
597         badly-structured-APDU           (3),
598         protocol-version-not-supported  (4),
599         other                           (5)
600         }

601     History-Report ::= SEQUENCE {
602         date-requested                  [0]  IMPLICIT ISO-Date OPTIONAL,
603         author                          [1]  ILL-String OPTIONAL,
604         title                           [2]  ILL-String OPTIONAL,
605         author-of-article               [3]  ILL-String OPTIONAL,
606         title-of-article                [4]  ILL-String OPTIONAL,
607         date-of-last-transition         [5]  IMPLICIT ISO-Date,
608         most-recent-service             [6]  IMPLICIT ENUMERATED {
609                                                 iLL-REQUEST                 (1),
610                                                 fORWARD                     (21),
611                                                 fORWARD-NOTIFICATION        (2),
612                                                 sHIPPED                     (3),
613                                                 iLL-ANSWER                  (4),
614                                                 cONDITIONAL-REPLY           (5),
615                                                 cANCEL                      (6),
616                                                 cANCEL-REPLY                (7),
617                                                 rECEIVED                    (8),
618                                                 rECALL                      (9),
619                                                 rETURNED                    (10),
620                                                 cHECKED-IN                  (11),
621                                                 oVERDUE                     (12),
622                                                 rENEW                       (13),
623                                                 rENEW-ANSWER                (14),
624                                                 lOST                        (15),
625                                                 dAMAGED                     (16),
626                                                 mESSAGE                     (17),
627                                                 sTATUS-QUERY                (18),
628                                                 sTATUS-OR-ERROR-REPORT      (19),
629                                                 eXPIRED                     (20)
630                                                 },
631         date-of-most-recent-service     [7]  IMPLICIT ISO-Date,
632         initiator-of-most-recent-service [8] IMPLICIT System-Id,
633         shipped-service-type            [9]  IMPLICIT Shipped-Service-Type OPTIONAL,
634                                         -- If the information is available, i.e. if a SHIPPED or
635                                         -- RECEIVED APDU has been sent or received, then the
636                                         -- value in this parameter shall be supplied.
637                                         -- Value must contain the most current information, e.g. if a
638                                         -- requester has received a SHIPPED APDU and then
639                                         -- invokes a RECEIVED.request, then the value from the
640                                         -- RECEIVED.request is used
641         transaction-results             [10] IMPLICIT Transaction-Results OPTIONAL,
642                                         -- If the information is available, i.e. if an ILL-ANSWER
643                                         -- APDU has been sent or received, then the value in this
644                                         -- parameter shall be supplied.
645         most-recent-service-note        [11] ILL-String OPTIONAL
646                                         -- If the information is available, i.e. if a note has been
647                                         -- supplied in the most recent service primitive, then the
```

```
                                    -- value in this parameter shall be supplied.
    }
Hold-Placed-Results ::= SEQUENCE {
    estimated-date-available        [0]  IMPLICIT ISO-Date,
    hold-placed-medium-type         [1]  IMPLICIT Medium-Type OPTIONAL,
    locations                       [2]  IMPLICIT SEQUENCE OF Location-Info OPTIONAL
    }

ILL-APDU-Type ::= ENUMERATED {
    iLL-REQUEST                     (1),
    fORWARD-NOTIFICATION            (2),
    sHIPPED                         (3),
    iLL-ANSWER                      (4),
    cONDITIONAL-REPLY               (5),
    cANCEL                          (6),
    cANCEL-REPLY                    (7),
    rECEIVED                        (8),
    rECALL                          (9),
    rETURNED                        (10),
    cHECKED-IN                      (11),
    oVERDUE                         (12),
    rENEW                           (13),
    rENEW-ANSWER                    (14),
    lOST                            (15),
    dAMAGED                         (16),
    mESSAGE                         (17),
    sTATUS-QUERY                    (18),
    sTATUS-OR-ERROR-REPORT          (19),
    eXPIRED                         (20)
    }

ILL-Service-Type ::= ENUMERATED {
    loan                            (1),
    copy-non-returnable             (2),
    locations                       (3),
    estimate                        (4),
    responder-specific              (5)
    }

ILL-String ::= CHOICE {
    GeneralString,
    -- may contain any ISO registered G (graphic) and C
    -- (control) character set
    EDIFACTString
    }
    -- may not include leading or trailing spaces
    -- may not consist only of space (" ") or non-printing
    -- characters

Intermediary-Problem ::= ENUMERATED {
    cannot-send-onward              (1)
    }

ISO-Date ::= VisibleString
    -- conforms to ISO 8601
    -- length = 8
    -- fixed
    -- YYYYMMDD

ISO-Time ::= VisibleString
    -- conforms to ISO 8601
    -- length = 6,
    -- fixed
```

```
705         -- HHMMSS
706         -- local time of person or institution invoking service

707     Item-Id ::= SEQUENCE {
708         item-type                       [0]  IMPLICIT ENUMERATED {
709                                                  monograph  (1),
710                                                  serial     (2),
711                                                  other      (3)
712                                                  } OPTIONAL,
713         held-medium-type                [1]  IMPLICIT Medium-Type OPTIONAL,
714         call-number                     [2]  ILL-String OPTIONAL,
715         author                          [3]  ILL-String OPTIONAL,
716         title                           [4]  ILL-String OPTIONAL,
717         sub-title                       [5]  ILL-String OPTIONAL,
718         sponsoring-body                 [6]  ILL-String OPTIONAL,
719         place-of-publication            [7]  ILL-String OPTIONAL,
720         publisher                       [8]  ILL-String OPTIONAL,
721         series-title-number             [9]  ILL-String OPTIONAL,
722         volume-issue                    [10] ILL-String OPTIONAL,
723         edition                         [11] ILL-String OPTIONAL,
724         publication-date                [12] ILL-String OPTIONAL,
725         publication-date-of-component   [13] ILL-String OPTIONAL,
726         author-of-article               [14] ILL-String OPTIONAL,
727         title-of-article                [15] ILL-String OPTIONAL,
728         pagination                      [16] ILL-String OPTIONAL,
729         national-bibliography-no        [17] EXTERNAL OPTIONAL,
730         iSBN                            [18] ILL-String (SIZE (10)) OPTIONAL,
731                                         -- must conform to ISO 2108-1978
732         iSSN                            [19] ILL-String (SIZE (8)) OPTIONAL,
733                                         -- must conform to ISO 3297-1986
734         system-no                       [20] EXTERNAL OPTIONAL,
735         additional-no-letters           [21] ILL-String OPTIONAL,
736         verification-reference-source   [22] ILL-String OPTIONAL
737         }

738     Location-Info ::= SEQUENCE {
739         location-id                     [0]  IMPLICIT System-Id,
740         location-address                [1]  IMPLICIT System-Address OPTIONAL,
741         location-note                   [2]  ILL-String OPTIONAL
742         }

743     Locations-Results ::= SEQUENCE {
744         reason-locs-provided            [0]  IMPLICIT Reason-Locs-Provided OPTIONAL,
745         locations                       [1]  IMPLICIT SEQUENCE OF Location-Info
746         }

747     Medium-Type ::= ENUMERATED {
748         printed                         (1),
749         microform                       (3),
750         film-or-video-recording         (4),
751         audio-recording                 (5),
752         machine-readable                (6),
753         other                           (7)
754         }

755     Name-Of-Person-Or-Institution ::= CHOICE {
756         name-of-person                  [0]  ILL-String,
757         name-of-institution             [1]  ILL-String
758         }

759     Person-Or-Institution-Symbol ::= CHOICE {
760         person-symbol                   [0]  ILL-String,
761         institution-symbol              [1]  ILL-String
```

```
762         }

763     Place-On-Hold-Type ::= ENUMERATED {
764         yes                                 (1),
765         no                                  (2),
766         according-to-responder-policy       (3)
767         }

768     Postal-Address ::= SEQUENCE {
769         name-of-person-or-institution       [0]  Name-Of-Person-Or-Institution OPTIONAL,
770         extended-postal-delivery-address    [1]  ILL-String OPTIONAL,
771         street-and-number                   [2]  ILL-String OPTIONAL,
772         post-office-box                     [3]  ILL-String OPTIONAL,
773         city                                [4]  ILL-String OPTIONAL,
774         region                              [5]  ILL-String OPTIONAL,
775         country                             [6]  ILL-String OPTIONAL,
776         postal-code                         [7]  ILL-String OPTIONAL
777         }

778     Provider-Error-Report ::= CHOICE {
779         general-problem                 [0]  IMPLICIT General-Problem,
780         transaction-id-problem          [1]  IMPLICIT Transaction-Id-Problem,
781         state-transition-prohibited     [2]  IMPLICIT State-Transition-Prohibited
782         }

783     Reason-Locs-Provided ::= ENUMERATED {
784         in-use-on-loan                  (1),
785         in-process                      (2),
786         lost                            (3),
787         non-circulating                 (4),
788         not-owned                       (5),
789         on-order                        (6),
790         volume-issue-not-yet-available  (7),
791         at-bindery                      (8),
792         lacking                         (9),
793         not-on-shelf                    (10),
794         on-reserve                      (11),
795         poor-condition                  (12),
796         cost-exceeds-limit              (13),
797         on-hold                         (19),
798         other                           (27),
799         responder-specific              (28)
800         }

801     Reason-No-Report ::= ENUMERATED {
802         temporary                       (1),
803         permanent                       (2)
804         }

805     Reason-Unfilled ::= ENUMERATED {
806         in-use-on-loan                  (1),
807         in-process                      (2),
808         lost                            (3),
809         non-circulating                 (4),
810         not-owned                       (5),
811         on-order                        (6),
812         volume-issue-not-yet-available  (7),
813         at-bindery                      (8),
814         lacking                         (9),
815         not-on-shelf                    (10),
816         on-reserve                      (11),
817         poor-condition                  (12),
818         cost-exceeds-limit              (13),
```

```
819         charges                                         (14),
820         prepayment-required                             (15),
821         lacks-copyright-compliance                      (16),
822         not-found-as-cited                              (17),
823         locations-not-found                             (18),
824         on-hold                                         (19),
825         policy-problem                                  (20),
826         mandatory-messaging-not-supported               (21),
827         expiry-not-supported                            (22),
828         requested-delivery-services-not-supported       (23),
829         preferred-delivery-time-not-possible            (24),
830         other                                           (27),
831         responder-specific                              (28)
832         }

833     Report-Source ::= ENUMERATED {
834         user                            (1),
835         provider                        (2)
836         }

837     Requester-Optional-Messages-Type ::= SEQUENCE {
838         can-send-RECEIVED               [0]  IMPLICIT BOOLEAN,
839         can-send-RETURNED               [1]  IMPLICIT BOOLEAN,
840         requester-SHIPPED               [2]  IMPLICIT ENUMERATED {
841                                                 requires        (1),
842                                                 desires         (2),
843                                                 neither         (3)
844                                                 },
845         requester-CHECKED-IN            [3]  IMPLICIT ENUMERATED {
846                                                 requires        (1),
847                                                 desires         (2),
848                                                 neither         (3)
849                                                 }
850         }

851     Responder-Optional-Messages-Type ::= SEQUENCE {
852         can-send-SHIPPED                [0]  IMPLICIT BOOLEAN,
853         can-send-CHECKED-IN             [1]  IMPLICIT BOOLEAN,
854         responder-RECEIVED              [2]  IMPLICIT ENUMERATED {
855                                                 requires        (1),
856                                                 desires         (2),
857                                                 neither         (3)
858                                                 },
859         responder-RETURNED              [3]  IMPLICIT ENUMERATED {
860                                                   requires      (1),
861                                                   desires       (2),
862                                                   neither       (3)
863                                                 }
864         }

865     Retry-Results ::= SEQUENCE {
866         reason-not-available            [0]  IMPLICIT ENUMERATED {
867                                                 in-use-on-loan                  (1),
868                                                 in-process                      (2),
869                                                 on-order                        (6),
870                                                 volume-issue-not-yet-available  (7),
871                                                 at-bindery                      (8),
872                                                 cost-exceeds-limit              (13),
873                                                 charges                         (14),
874                                                 prepayment-required             (15),
875                                                 lacks-copyright-compliance      (16),
876                                                 not-found-as-cited              (17),
877                                                 on-hold                         (19),
```

```
878                                         other                                   (27),
879                                         responder-specific                      (28)
880                                         } OPTIONAL,
881         retry-date                  [1] IMPLICIT ISO-Date OPTIONAL,
882         locations                   [2] IMPLICIT SEQUENCE OF Location-Info OPTIONAL
883         }

884     Search-Type ::= SEQUENCE {
885         level-of-service            [0] ILL-String (SIZE (1)) OPTIONAL,
886         need-before-date            [1] IMPLICIT ISO-Date OPTIONAL,
887         expiry-flag                 [2] IMPLICIT ENUMERATED {
888                                             need-Before-Date        (1),
889                                             other-Date              (2),
890                                             no-Expiry               (3)
891                                             } DEFAULT 3,
892                                             -- value of "need-Before-Date" indicates that
893                                             -- need-before-date also specifies transaction expiry
894                                             -- date
895         expiry-date                 [3] IMPLICIT ISO-Date OPTIONAL
896                                     -- alternative expiry date can be used only when expiry-flag
897                                     -- is set to "Other-Date"
898         }

899     Security-Problem ::= ILL-String

900     Send-To-List-Type ::= SEQUENCE OF SEQUENCE {
901         system-id                   [0] IMPLICIT System-Id,
902         account-number              [1] Account-Number OPTIONAL,
903         system-address              [2] IMPLICIT System-Address OPTIONAL
904         }

905     Service-Date-Time ::= SEQUENCE {
906         date-time-of-this-service   [0] IMPLICIT SEQUENCE {
907                                             date        [0]     IMPLICIT ISO-Date,
908                                             time        [1]     IMPLICIT ISO-Time OPTIONAL
909                                             -- mandatory for 2nd and subsequent services
910                                             -- invoked for a given
911                                             -- ILL-transaction on the same day
912                                             },
913         date-time-of-original-service [1] IMPLICIT SEQUENCE {
914                                             date        [0]     IMPLICIT ISO-Date,
915                                             time        [1]     IMPLICIT ISO-Time OPTIONAL
916                                             } OPTIONAL
917         }

918     Shipped-Service-Type ::= ILL-Service-Type (loan | copy-non-returnable)
919         -- subtype of ILL-Service-Type

920     State-Transition-Prohibited ::= SEQUENCE {
921         aPDU-type                   [0] IMPLICIT ILL-APDU-Type,
922         current-state               [1] IMPLICIT Current-State
923         }

924     Status-Report ::= SEQUENCE {
925         user-status-report          [0] IMPLICIT History-Report,
926         provider-status-report      [1] IMPLICIT Current-State
927         }

928     Supplemental-Item-Description ::= SEQUENCE OF EXTERNAL
929         -- the syntax of supplementary item description information is defined outside this standard

930     Supply-Details ::= SEQUENCE {
931         date-shipped                [0] IMPLICIT ISO-Date OPTIONAL,
```

```
932         date-due                        [1]  IMPLICIT Date-Due OPTIONAL,
933         chargeable-units                [2]  IMPLICIT INTEGER (1..9999) OPTIONAL,
934         cost                            [3]  IMPLICIT Amount OPTIONAL,
935         shipped-conditions              [4]  IMPLICIT ENUMERATED {
936                                                  library-use-only                    (22),
937                                                  no-reproduction                     (23),
938                                                  client-signature-required           (24),
939                                                  special-collections-supervision-required (25),
940                                                  other                               (27)
941                                                  } OPTIONAL,
942         shipped-via                             CHOICE {
943                                                  physical-delivery      [5]    Transportation-Mode,
944                                                  electronic-delivery    [50]   IMPLICIT
945                                                                                Electronic-Delivery-Service
946                                                  } OPTIONAL,
947                                             -- electronic-delivery may only be present in APDUs with a
948                                             -- protocol-version-num value of 2 or greater
949         insured-for                     [6]  IMPLICIT Amount OPTIONAL,
950         return-insurance-required       [7]  IMPLICIT Amount OPTIONAL,
951         no-of-units-per-medium          [8]  IMPLICIT SEQUENCE OF Units-Per-Medium-Type
952                                         OPTIONAL
953         }

954     Supply-Medium-Info-Type ::= SEQUENCE {
955         supply-medium-type              [0]  IMPLICIT Supply-Medium-Type,
956         medium-characteristics          [1]  ILL-String OPTIONAL
957         }

958     Supply-Medium-Type ::= ENUMERATED {
959         printed                         (1),
960         photocopy                       (2),
961         microform                       (3),
962         film-or-video-recording         (4),
963         audio-recording                 (5),
964         machine-readable                (6),
965         other                           (7)
966         }

967     System-Address ::= SEQUENCE {
968         telecom-service-identifier      [0]  ILL-String OPTIONAL,
969         telecom-service-address         [1]  ILL-String OPTIONAL
970         }

971     System-Id ::= SEQUENCE {
972         --at least one of the following must be present
973         person-or-institution-symbol    [0]  Person-Or-Institution-Symbol OPTIONAL,
974         name-of-person-or-institution   [1]  Name-Of-Person-Or-Institution OPTIONAL
975         }

976     Third-Party-Info-Type ::= SEQUENCE {
977         permission-to-forward           [0]  IMPLICIT BOOLEAN DEFAULT FALSE,
978         permission-to-chain             [1]  IMPLICIT BOOLEAN DEFAULT FALSE,
979         permission-to-partition         [2]  IMPLICIT BOOLEAN DEFAULT FALSE,
980         permission-to-change-send-to-list [3] IMPLICIT BOOLEAN DEFAULT FALSE,
981         initial-requester-address       [4]  IMPLICIT System-Address OPTIONAL,
982                                         -- mandatory when initiating a FORWARD service or an
983                                         -- ILL-REQUEST service for a partitioned ILL
984                                         -- sub-transaction; optional otherwise
985         preference                      [5]  IMPLICIT ENUMERATED {
986                                                  ordered      (1),
987                                                  unordered    (2)
988                                                  } DEFAULT 2,
989         send-to-list                    [6]  IMPLICIT Send-To-List-Type OPTIONAL,
```

```
990         already-tried-list              [7]  IMPLICIT Already-Tried-List-Type OPTIONAL
991                                         -- mandatory when initiating a FORWARD service, or when
992                                         -- initiating an ILL-REQUEST service for an ILL
993                                         -- sub-transaction if the received ILL-REQUEST included an
994                                         -- "already-tried-list"; optional otherwise
995         }

996     Transaction-Id ::= SEQUENCE {
997         initial-requester-id            [0]  IMPLICIT System-Id OPTIONAL,
998                                         -- mandatory for sub-transactions; not called
999                                         -- "requester-id" to distinguish id of initial-requester
1000                                        --from id of requester of sub-transaction if there is one
1001        transaction-group-qualifier     [1]  ILL-String,
1002        transaction-qualifier           [2]  ILL-String,
1003        sub-transaction-qualifier       [3]  ILL-String OPTIONAL
1004                                        -- mandatory for sub-transactions
1005        }

1006    Transaction-Id-Problem ::= ENUMERATED {
1007        duplicate-transaction-id        (1),
1008        invalid-transaction-id          (2),
1009        unknown-transaction-id          (3)
1010        }

1011    Transaction-Results ::= ENUMERATED {
1012        conditional                     (1),
1013        retry                           (2),
1014        unfilled                        (3),
1015        locations-provided              (4),
1016        will-supply                     (5),
1017        hold-placed                     (6),
1018        estimate                        (7)
1019        }

1020    Transaction-Type ::= ENUMERATED  {
1021        simple                          (1),
1022        chained                         (2),
1023        partitioned                     (3)
1024        }

1025    Transportation-Mode ::= ILL-String

1026    Unable-To-Perform ::= ENUMERATED {
1027        not-available                   (1),
1028        resource-limitation             (2),
1029        other                           (3)
1030        }

1031    Unfilled-Results ::= SEQUENCE {
1032        reason-unfilled                 [0]  IMPLICIT Reason-Unfilled,
1033        locations                       [1]  IMPLICIT SEQUENCE OF Location-Info OPTIONAL
1034        }

1035    Units-Per-Medium-Type ::= SEQUENCE {
1036        medium                          [0]  Supply-Medium-Type,
1037        no-of-units                     [1]  INTEGER (1..9999)
1038        }

1039    User-Error-Report ::= CHOICE {
1040        already-forwarded               [0]  IMPLICIT Already-Forwarded,
1041        intermediary-problem            [1]  IMPLICIT Intermediary-Problem,
1042        security-problem                [2]  Security-Problem,
1043        unable-to-perform               [3]  IMPLICIT Unable-To-Perform
```

```
1044        }
1045
1046    Will-Supply-Results ::= SEQUENCE {
1047        reason-will-supply              [0]  ENUMERATED {
1048                                                in-use-on-loan              (1),
1049                                                in-process                  (2),
1050                                                on-order                    (6),
1051                                                at-bindery                  (8),
1052                                                on-hold                     (19),
1053                                                being-processed-for-supply  (26),
1054                                                other                       (27),
1055                                                responder-specific          (28),
1056                                                electronic-delivery         (30)
1057                                                },
1058        supply-date                     [1]  ISO-Date OPTIONAL,
1059        return-to-address               [2]  Postal-Address OPTIONAL,
1060        locations                       [3]  IMPLICIT SEQUENCE OF Location-Info OPTIONAL,
1061        electronic-delivery-service     [4] Electronic-Delivery-Service OPTIONAL
1062                                        -- if present, this must be one of the services proposed by
1063                                        -- the requester
1064        }

1065    EDIFACTString ::= VisibleString (FROM ("A"|"B"|"C"|"D"|"E"|"F"|"G"|"H"|
1066        "I"|"J"|"K"|"L"|"M"|"N"|"O"|"P"|"Q"|"R"|"S"|"T"|"U"|
1067        "V"|"W"|"X"|"Y"|"Z"|"a"|"b"|"c"|"d"|"e"|"f"|"g"|"h"|
1068        "i"|"j"|"k"|"l"|"m"|"n"|"o"|"p"|"q"|"r"|"s"|"t"|"u"|
1069        "v"|"w"|"x"|"y"|"z"|"1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|
1070        "9"|"0"|" "|"."|","|"-"|"("|")"|"/"|"="|"!"|""""|"%"|"&"|
1071        "*"|";"|"<"|">"|""""|"+"|":"|"?"))

1072    END
```

10 Conformance

10.1 Static Conformance

An implementation claiming conformance to this part of ISO 10161 shall be capable of:

a. following the procedures defined for one of the following:
   - requester role,
   - responder role,
   - intermediary role,
   - any combination of the above;

b. supporting simple transactions as a minimum when acting in the role of requester or responder; and supporting one or both of chained and partitioned transactions when acting in the role of intermediary;

c. supporting at least one of "loan" and "copy/non-returnable" service types;

d. support the invocation of the following services for the requester role: mandatory services ILL-REQUEST, CONDITIONAL-REPLY, RECEIVED, LOST and STATUS-OR-ERROR-REPORT when supporting any service type; and conditional service RETURNED when supporting the "loan" service type (see note 1);

e. support the invocation of the following services for the responder role: mandatory services SHIPPED, ILL-ANSWER, CANCEL-REPLY, LOST and STATUS-OR-ERROR-REPORT when supporting any service type; and conditional services RECALL, CHECKED-IN, OVERDUE, and RENEW-ANSWER when supporting the "loan" service type (see note 1);

f. support the invocation of the following services for the intermediary role: mandatory services ILL-REQUEST, CONDITIONAL-REPLY, STATUS-QUERY, STATUS-OR-ERROR REPORT, SHIPPED, RECEIVED, ILL-ANSWER, CANCEL, CANCEL-REPLY, LOST, DAMAGED and MESSAGE when supporting any service type; and conditional services RECALL, OVERDUE, CHECKED-IN, RENEW, RENEW-ANSWER and RETURNED when supporting the "loan" service type (see note 1);

g. receiving APDUs with data defined for all types as defined in clause 9;

h. supporting all mandatory types for transmitted APDUs as defined in clause 9 (see note 2);

i. defining data for optional types which are supported by the implementation (see note 3).

NOTES

1 Optional services for the requester and responder roles include the DAMAGED, MESSAGE and STATUS-QUERY services. When the optional services are not supported, then, although an implementation might not be capable of transmitting the APDUs associated with the service, it shall be capable of receiving the associated APDUs. When the conditional services are not supported, an implementation need not be capable of transmitting or accepting the APDUs associated with the service.

2 When supporting a mandatory type, the implementation shall always define data for that type. If a structured type is mandatory, but is made up only of optional types, then at least one of the optional types shall be present. If a structured type is optional, but is constructed from a mandatory type, then the type is mandatory only if the structured type is present.

3 Supporting an optional type refers to the capability of the implementation to define data for the supported type when transmitting APDUs. The data for the optional type need not always be present. The conditions for the presence of data within the optional type is a local implementation issue. Not supporting an optional type indicates that the implementation is not capable of providing data defined for the type. Note that the implementation shall still be able to receive the unsupported optional types.

10.2 Dynamic Conformance

An implementation that claims conformance to this part of ISO 10161 shall exhibit external behaviour consistent with:

a. having implemented an ILL ASE as defined by clause 8 of this part of ISO 10161;

b. its stated capabilities and requirements regarding optional messages, as indicated in the ILL-REQUEST, ILL-ANSWER and SHIPPED APDUs;

c. having implemented the capability to always send and receive the APDUs SHIPPED, RECEIVED, RETURNED and CHECKED-IN when acting in the role of intermediary;

d. encoding APDUs as defined in clause 9 of this part of ISO 10161. More than one encoding scheme can be applied to the values of the data types that are defined using ASN.1. The mandatory encoding scheme is the Basic Encoding Rules for Abstract Syntax Notation One defined in ISO/IEC 8825. An additional possibility is the encoding scheme defined by the EDIFACT standard, ISO 9735, and annex B.

10.3 Protocol Implementation Conformance Statement Requirements

Each implementation must provide a Protocol Implementation Conformance Statement (PICS). The PICS must state:

a. which roles are supported;
b. which service types are supported;
c. which transaction types are supported;
d. which services are supported;
e. which APDUs are supported;
f. which APDU data types are supported;
g. the requirements for other systems regarding optional APDUs to allow interworking, i.e. a system may require that other systems must support all optional APDUs when interworking with that system;
h. which encoding rules are supported;
i. which version or versions of the protocol are supported.

Annex A
(normative)
ILL State Tables

A.1 General

The ILL protocol defines a set of states through which an ILL-transaction progresses.

In this annex state tables are presented for each of the requester, responder and intermediary. These tables show the interrelationship between the state of an ILL Protocol Machine (ILLPM), the incoming events that occur in the protocol, the actions taken, and, finally, the resultant state of the ILLPM.

This annex contains the following tables:

a. Table A.1 specifies the abbreviated name and description of each incoming event.

b. Table A.2 specifies the abbreviated name and description of each outgoing event.

c. Table A.3 specifies the predicates.

d. Table A.4 specifies the state table for the requester for the processing phase of an ILL-transaction. This table fully describes all states for an ILL-transaction involving a non-returnable item such as a photocopy.

e. Table A.5 specifies the state table for the requester for the tracking phase of an ILL-transaction. This table applies only to ILL-transactions involving a returnable item such as a monograph.

   NOTE — For convenience in presentation, this table is split into two parts, tables A.5a and A.5b. This division has no general significance.

f. Table A.6 specifies the state table for the responder for the processing phase. This table applies to all ILL-service-types, including requests for returnable and non-returnable items.

g. Table A.7 specifies the state table for the responder for the tracking phase. This table applies to all ILL-service-types, including requests for returnable and non-returnable items.

h. Table A.8 specifies the state table for the intermediary acting in the role of requester.

i. Table A.9 specifies the state table for the intermediary acting in the role of responder.

A.2 Conventions

The intersection of an incoming event (row) and a state (column) forms a cell.

In the state table, a blank cell represents the combination of an incoming event and a state that is not defined for the ILL protocol.

A non-blank cell represents an incoming event and state that is defined for the ILL protocol. Such a cell contains an action list. An action list may be either mandatory or conditional.

A mandatory action list contains:

a. an outgoing event;

b. optionally, a change to a local protocol variable or timer; and

c. a resultant state.

A conditional action list contains:

a. a predicate expression comprising predicates and Boolean operators (^ represents the Boolean NOT); and

b. a mandatory action list. This list is used only if the predicate expression is true.

Separate rows are provided for original and repeated events to distinguish between possibly different action lists. Note that separate rows are not provided for repeated events associated with the MESSAGE, STATUS-QUERY, STATUS-OR-ERROR-REPORT and DAMAGED services and APDUs since these are not repeatable; each such occurrence is considered to be an original event.

Original and repeated service request events are distinguished on the basis of the following criteria:

a. For a requester or responder, an original service request has no value for the "date-time-of-original-service" field. A repeated event has a value defined for that field; a repeated service request can only be invoked while there has been no state change within the originating application-entity-invocation since the original service request event.

b. For an intermediary, an original service request may have a value for "date-time-of-original-service" as long as there was no immediately preceding service request of the same type. A repeated service request has a value defined for the "date-time-of-original-service" and was immediately preceded by a service request of the same type with the same date and time value

in either the "date-time-of-original-service" or "date-time-of-this-service" field. It is possible for a repeated service request to be invoked in a state different from the original request, due to the possibility of incoming events.

NOTE — A service request followed by one or more of MESSAGE, STATUS-QUERY, STATUS-OR-ERROR-REPORT or DAMAGED, then followed by another service request of the same type as the first, is considered to immediately precede the second occurrence of the same service.

Original and repeated APDU events are distinguished on the basis that a repeated APDU satisfies the following criteria:

a. a value is provided for the type "date-and-time-of-original-service" and the "service-date-time" structured type; and

b. this value is the same as the REPEAT-TIME-STAMP protocol variable.

A.3 Actions to be Taken by the ILL Protocol

The ILL protocol state tables define the action to be taken by the ILL protocol machine in terms of an outgoing event, changes to local protocol variables or timers and the resultant state of the ILLPM.

A.3.1 Invalid Intersections

Blank cells indicate an invalid intersection of an incoming event and state. If such an intersection occurs, it is considered to be a protocol error. The ILLPM ignores the incoming event, sends a STATUS-OR-ERROR-REPORT APDU to notify the sender of the protocol error, and makes no state changes. Additional local actions, such as the notification of the service-user, may also be taken, but are not specified in this part of ISO 10161.

A.3.2 Valid Intersections

If the intersection of the state and incoming event is valid, one of the following actions is taken:

a. if a cell contains a mandatory action list, the ILLPM takes the action specified;

b. if a cell contains a conditional action list and if the predicate expression is true, the ILLPM takes the actions specified. If the predicate expression is not true, then the action and state change specified for that cell are not taken.

NOTE — For some combinations of state and input events, different actions and state changes are possible, depending on the value of the predicate. In such cases, both possibilities are reflected in the state tables on separate rows.

A.4 Relationship to Supporting Services

The state tables defined in this annex do not take into account interactions with supporting services, as these depend on mappings defined outside of this part of ISO 10161. For each such mapping that is defined, additional states and state transitions may be required.

Table A.1 — Incoming Event List

Abbreviated Name	Description
ILLreq	ILL-REQUEST.request
FWDreq	FORWARD.request
ANSreq-CO	ILL-ANSWER.request : result = CONDITIONAL
ANSreq-RY	ILL-ANSWER.request : result = RETRY
ANSreq-UN	ILL-ANSWER.request : result = UNFILLED
ANSreq-LP	ILL-ANSWER.request : result = LOCATIONS-PROVIDED
ANSreq-WS	ILL-ANSWER.request : result = WILL-SUPPLY
ANSreq-HP	ILL-ANSWER.request : result = HOLD PLACED
ANSreq-ES	ILL-ANSWER.request : result = ESTIMATE
C-REPreq +	CONDITIONAL-REPLY.request : answer = yes
C-REPreq -	CONDITIONAL-REPLY.request : answer = no
CANreq	CANCEL.request
CARreq +	CANCEL-REPLY.request: answer = yes
CARreq -	CANCEL-REPLY.request: answer = no
SHIreq	SHIPPED.request
RCVreq	RECEIVED.request
RCLreq	RECALL.request
DUEreq	OVERDUE.request
RETreq	RETURNED.request
RENreq	RENEW.request
REAreq +	RENEW-ANSWER.request : answer = yes
REAreq -	RENEW-ANSWER.request : answer = no
CHKreq	CHECKED-IN.request
LSTreq	LOST.request
DAMreq	DAMAGED.request
MSGreq	MESSAGE.request
STQreq	STATUS-QUERY.request
STRreq	STATUS-OR-ERROR-REPORT.request
ILL	receive ILL-REQUEST APDU
FWD	receive FORWARD-NOTIFICATION APDU
ANS-CO	receive ILL-ANSWER APDU : result = CONDITIONAL
ANS-RY	receive ILL-ANSWER APDU : result = RETRY

Table A.1 — Incoming Event List *(concluded)*

Abbreviated Name	Description
ANS-UN	receive ILL-ANSWER APDU : result = UNFILLED
ANS-LP	receive ILL-ANSWER APDU : result = LOCATIONS-PROVIDED
ANS-WS	receive ILL-ANSWER APDU : result = WILL-SUPPLY
ANS-HP	receive ILL-ANSWER APDU : result = HOLD PLACED
ANS-ES	receive ILL-ANSWER APDU : result = ESTIMATE
C-REP +	receive CONDITIONAL-REPLY APDU : answer = yes
C-REP -	receive CONDITIONAL-REPLY APDU : answer = no
CAN	receive CANCEL APDU
CAR +	receive CANCEL-REPLY APDU: answer = yes
CAR -	receive CANCEL-REPLY APDU: answer = no
SHI	receive SHIPPED APDU
RCV	receive RECEIVED APDU
RCL	receive RECALL APDU
DUE	receive OVERDUE APDU
RET	receive RETURNED APDU
REN	receive RENEW APDU
REA +	receive RENEW-ANSWER APDU : answer = yes
REA -	receive RENEW-ANSWER APDU : answer = no
CHK	receive CHECKED-IN APDU
LST	receive LOST APDU
DAM	receive DAMAGED APDU
MSG	receive MESSAGE APDU
STQ	receive STATUS-QUERY APDU
STR	receive STATUS-OR-ERROR-REPORT APDU
EXP	receive EXPIRED APDU
EXPIRY timeout	Transaction Timer Expiry

Table A.2 — Outgoing Event List

Abbreviated Name	Description
ILLind	ILL-REQUEST.indication
FWDind	FORWARD.indication
ANSind-CO	ILL-ANSWER.indication : result = CONDITIONAL
ANSind-RY	ILL-ANSWER.indication : result = RETRY
ANSind-UN	ILL-ANSWER.indication : result = UNFILLED
ANSind-LP	ILL-ANSWER.indication : result = LOCATIONS-PROVIDED
ANSind-WS	ILL-ANSWER.indication : result = WILL-SUPPLY
ANSind-HP	ILL-ANSWER.indication : result = HOLD PLACED
ANSind-ES	ILL-ANSWER.indication : result = ESTIMATE
C-REPind +	CONDITIONAL-REPLY.indication : answer = yes
C-REPind -	CONDITIONAL-REPLY.indication : answer = no
CANind	CANCEL.indication
CARind +	CANCEL-REPLY.indication : answer = yes
CARind -	CANCEL-REPLY.indication : answer = no
SHIind	SHIPPED.indication
RCVind	RECEIVED.indication
RCLind	RECALL.indication
DUEind	OVERDUE.indication
RETind	RETURNED.indication
RENind	RENEW.indication
REAind +	RENEW-ANSWER.indication : answer = yes
REAind -	RENEW-ANSWER.indication : answer = no
CHKind	CHECKED-IN.indication
LSTind	LOST.indication
DAMind	DAMAGED.indication
MSGind	MESSAGE.indication
STQind	STATUS-QUERY.indication
STRind	STATUS-OR-ERROR-REPORT.indication
ILL	send ILL-REQUEST APDU
FWD	send FORWARD-NOTIFICATION APDU
ANS-CO	send ILL-ANSWER APDU : result = CONDITIONAL
ANS-RY	send ILL-ANSWER APDU : result = RETRY

Table A.2 — Outgoing Event List *(concluded)*

Abbreviated Name	Description
ANS-UN	send ILL-ANSWER APDU : result = UNFILLED
ANS-LP	send ILL-ANSWER APDU : result = LOCATIONS-PROVIDED
ANS-WS	send ILL-ANSWER APDU : result = WILL-SUPPLY
ANS-HP	send ILL-ANSWER APDU : result = HOLD PLACED
ANS-ES	send ILL-ANSWER APDU : result = ESTIMATE
C-REP +	send CONDITIONAL-REPLY APDU : answer = yes
C-REP -	send CONDITIONAL-REPLY APDU : answer = no
CAN	send CANCEL APDU
CAR +	send CANCEL-REPLY APDU : answer = yes
CAR -	send CANCEL-REPLY APDU : answer = no
SHI	send SHIPPED APDU
RCV	send RECEIVED APDU
RCL	send RECALL APDU
DUE	send OVERDUE APDU
RET	send RETURNED APDU
REN	send RENEW APDU
REA +	send RENEW-ANSWER APDU : answer = yes
REA-	send RENEW-ANSWER APDU : answer = no
CHK	send CHECKED-IN APDU
LST	send LOST APDU
DAM	send DAMAGED APDU
MSG	send MESSAGE APDU
STQ	send STATUS-QUERY APDU
STR	send STATUS-OR-ERROR-REPORT APDU
EXP	send EXPIRED APDU

Table A.3 — Predicates

Code	Meaning
p1	returns TRUE if the transaction-type parameter of the ILL-REQUEST service is "simple"
p2	returns TRUE if the transaction-type parameter of the ILL-REQUEST service is chained, the CHAIN protocol variable is TRUE and the transaction-id is valid for a sub-transaction
p3	returns TRUE if the transaction-type parameter of the ILL-REQUEST service is partitioned, the PART protocol variable is TRUE and the transaction-id is valid for a sub-transaction
p4	returns TRUE if the FWD protocol variable is TRUE
p5	returns TRUE if the RETURN protocol variable is TRUE
p6	returns TRUE if the CHAIN protocol variable is TRUE
p7	returns TRUE if a received APDU is in sequence
p8	returns TRUE if the most recent event that caused a state change is NOT (DUEreq, DUE)

NOTE — For repeated events, whether a request or an incoming APDU, the state tables do not include a predicate since the predicate was evaluated for the original event.

Table A.4 — Requester State Table — Processing Phase

	IDLE	PENDING	NOT-SUPPLIED	CONDITIONAL	CANCEL-PENDING	CANCELLED	SHIPPED
ILLreq	p1 ILL PENDING						
ILLreq repeat		ILL PENDING					
C-REPreq +				C-REP + PENDING			
C-REPreq + repeat		C-REP + PENDING					
C-REPreq -				C-REP - NOT-SUPPLIED			
C-REPreq - repeat			C-REP - NOT-SUPPLIED				
CANreq		CAN CANCEL-PENDING					
CANreq repeat					CAN CANCEL-PENDING		
RCVreq		RCV (opt) set RETURN var RECEIVED			RCV (opt) set RETURN var RECEIVED		RCV(opt) set RETURN var RECEIVED
RCVreq repeat							
LSTreq		LST LOST			LST LOST		LST LOST
LSTreq repeat							
MSGreq		MSG PENDING	MSG NOT-SUPPLIED	MSG CONDITIONAL	MSG CANCEL-PENDING	MSG CANCELLED	MSG SHIPPED
STQreq		STQ PENDING	STQ NOT-SUPPLIED	STQ CONDITONAL	STQ CANCEL-PENDING	STQ CANCELLED	STQ SHIPPED
STRreq		STR PENDING	STR NOT-SUPPLIED	STR CONDITIONAL	STR CANCEL-PENDING	STR CANCELLED	STR SHIPPED
FWD		FWDind PENDING			FWDind PENDING		
FWD repeat		FWDind PENDING					

Table A.4 — Requester State Table — Processing Phase *(continued)*

	IDLE	PENDING	NOT-SUPPLIED	CONDITIONAL	CANCEL-PENDING	CANCELLED	SHIPPED
ANS-CO		p7 ANSind-CO CONDITIONAL	ANSind-CO NOT-SUPPLIED	ANSind-CO CONDITIONAL	p7 ANSind-CO CANCEL-PENDING	ANSind-CO CANCELLED	
ANS-CO		^p7 ANSind-CO PENDING	ANSind-CO NOT-SUPPLIED	ANSind-CO CONDITIONAL	^p7 ANSind-CO CANCEL-PENDING	ANSind-CO CANCELLED	
ANS-CO repeat		ANSind-CO PENDING	ANSind-CO NOT-SUPPLIED	ANSind-CO CONDITIONAL	ANSind-CO CANCEL-PENDING	ANSind-CO CANCELLED	
ANS-RY		ANSind-RY NOT-SUPPLIED	ANSind-RY NOT-SUPPLIED		ANSind-RY NOT-SUPPLIED		
ANS-RY repeat			ANSind-RY NOT-SUPPLIED				
ANS-UN		ANSind-UN NOT-SUPPLIED	ANSind-UN NOT-SUPPLIED		ANSind-UN NOT-SUPPLIED		
ANS-UN repeat			ANSind-UN NOT-SUPPLIED				
ANS-LP		ANSind-LP NOT-SUPPLIED	ANSind-LP NOT-SUPPLIED		ANSind-LP NOT-SUPPLIED		
ANS-LP repeat			ANSind-LP NOT-SUPPLIED				
ANS-WS		ANSind-WS PENDING	ANSind-WS NOT-SUPPLIED	ANSind-WS CONDITIONAL	ANSind-WS CANCEL-PENDING		ANSind-WS SHIPPED
ANS-WS repeat		ANSind-WS PENDING	ANSind-WS NOT-SUPPLIED	ANSind-WS CONDITIONAL	ANSind-WS CANCEL-PENDING		ANSind-WS SHIPPED
ANS-HP		ANSind-HP PENDING	ANSind-HP NOT-SUPPLIED	ANSind-HP CONDITIONAL	ANSind-HP CANCEL-PENDING		ANSind-HP SHIPPED
ANS-HP repeat		ANSind-HP PENDING	ANSind-HP NOT-SUPPLIED	ANSind-HP CONDITIONAL	ANSind-HP CANCEL-PENDING		ANSind-HP SHIPPED
ANS-ES		ANSind-ES NOT-SUPPLIED	ANSind-ES NOT-SUPPLIED		ANSind-ES NOT-SUPPLIED		
ANS-ES repeat			ANSind-ES NOT-SUPPLIED				
CAR +					CARind + CANCELLED	CARind + CANCELLED	
CAR + repeat						CARind + CANCELLED	
CAR - repeat		CARind - PENDING					Carind - SHIPPED
CAR -		CARind - PENDING			CARind - PENDING		CARind - SHIPPED

Table A.4 — Requester State Table — Processing Phase *(concluded)*

	IDLE	PENDING	NOT-SUPPLIED	CONDITIONAL	CANCEL-PENDING	CANCELLED	SHIPPED
SHI		SHIind SHIPPED			SHIind SHIPPED		SHIind SHIPPED
SHI repeat							SHIind SHIPPED
MSG		MSGind PENDING	MSGind NOT-SUPPLIED	MSGind CONDITIONAL	MSGind CANCEL-PENDING	MSGind CANCELLED	MSGind SHIPPED
STQ		STQind PENDING	STQind NOT-SUPPLIED	STQind CONDITIONAL	STQind CANCEL-PENDING	STQind CANCELLED	STQind SHIPPED
STR		STRind PENDING	STRind NOT-SUPPLIED	STRind CONDITIONAL	STRind CANCEL-PENDING	STRind CANCELLED	STRind SHIPPED
EXP		EXPind NOT-SUPPLIED	EXPind NOT-SUPPLIED	EXPind NOT-SUPPLIED	EXPind NOT-SUPPLIED		
LST		LSTind LOST			LSTind LOST		LSTind LOST
LST repeat							

Table A.5a — Requester State Table — Tracking Phase (Part 1)

	PENDING	CANCEL PENDING	RECEIVED	RENEW/ PENDING	RENEW/ OVERDUE	SHIPPED
RCVreq	RCV (opt) set RETURN var RECEIVED	RCV (opt) set RETURN var RECEIVED				RCV (opt) set RETURN var RECEIVED
RCVreq repea t			RCV (opt) RECEIVED			
RET req			p5 RET (opt) RETURNED	RET (opt) RETURNED	RET (opt) RETURNED	
RETreq repeat						
RENreq			p5 REN RENEW/ PENDING		REN RENEW/ OVERDUE	
RENreq repeat				REN RENEW/ PENDING	REN RENEW/ OVERDUE	
LSTreq	LST LOST	LST LOST	p5 LST LOST	LST LOST	LST LOST	LST LOST
LSTreq repeat						
DAMreq			DAM RECEIVED	DAM RENEW/ PENDING	DAM RENEW/ OVERDUE	
MSGreq	MSG PENDING	MSG CANCEL-PENDING	MSG RECEIVED	MSG RENEW/ PENDING	MSG RENEW/ OVERDUE	MSG SHIPPED
STQreq	STQ PENDING	STQ CANCEL-PENDING	STQ RECEIVED	STQ RENEW/ PENDING	STQ RENEW/ OVERDUE	STQ SHIPPED
STRreq	STR PENDING	STR CANCEL-PENDING	STR RECEIVED	STR RENEW/ PENDING	STR RENEW/ OVERDUE	STR SHIPPED
ANS-WS	ANSind-WS PENDING	ANSind-WS CANCEL-PENDING	ANSind-WS RECEIVED	ANSind-WS RENEW/ PENDING	ANSind-WS RENEW/ OVERDUE	ANSind-WS SHIPPED
ANS-WS repeat	ANSind-WS PENDING	ANSind-WS CANCEL-PENDING	ANSind-WS RECEIVED	ANSind-WS RENEW/ PENDING	ANSind-WS RENEW/ OVERDUE	ANSind-WS SHIPPED
ANS-HP	ANSind-HP PENDING	ANSind-HP CANCEL-PENDING	ANSind-HP RECEIVED	ANSind-HP RENEW/ PENDING	ANSind-HP RENEW/ OVERDUE	ANSind-HP SHIPPED
ANS-HP repeat	ANSind-HP PENDING	ANSind-HP CANCEL-PENDING	ANSind-HP RECEIVED	ANSind-HP RENEW/ PENDING	ANSind-HP RENEW/ OVERDUE	ANSind-HP SHIPPED
CAR -	CARind - PENDING	CARind - PENDING	CARind - RECEIVED	CARind - RENEW/ PENDING	CARind - RENEW/ OVERDUE	CARind - SHIPPED
CAR - repeat	CARind - PENDING		CARind - RECEIVED	CARind - RENEW/ PENDING	CARind - RENEW/ OVERDUE	CARind - SHIPPED

Table A.5a — Requester State Table — Tracking Phase (Part 1) *(continued)*

	PENDING	CANCEL PENDING	RECEIVED	RENEW/ PENDING	RENEW/ OVERDUE	SHIPPED
SHI	SHIind SHIPPED	SHIind SHIPPED	SHIind RECEIVED	SHIind RENEW/ PENDING	SHIind RENEW/ OVERDUE	SHIind SHIPPED
SHI repeat			SHIind RECEIVED	SHIind RENEW/PENDING	SHIind RENEW/ OVERDUE	SHIind SHIPPED
RCL	RCLind RECALL	RCLind RECALL	p5 RCLind RECALL	RCLind RECALL	RCLind RECALL	p5 RCLind RECALL
RCL repeat						
DUE	DUEind NOT-RCVD/ OVERDUE	DUEind NOT-RCVD/ OVERDUE	p5 DUEind OVERDUE	DUEind RENEW/ OVERDUE	p7 and p8 DUEind OVERDUE	p5 DUEind NOT-RCVD/ OVERDUE
DUE	DUEind NOT-RCVD/ OVERDUE	DUEind NOT-RCVD/ OVERDUE	p5 DUEind OVERDUE	DUEind RENEW/ OVERDUE	^p7 DUEind RENEW/ OVERDUE	p5 DUEind NOT-RCVD/ OVERDUE
DUE repeat					DUEind RENEW/ OVERDUE	
LST	LSTind LOST	LSTind LOST				LSTind LOST
LST repeat						
DAM						
MSG	MSGind PENDING	MSGind CANCEL-PENDING	MSGind RECEIVED	MSGind RENEW/ PENDING	MSGind RENEW/ OVERDUE	MSGind SHIPPED
STQ	STQind PENDING	STQind CANCEL-PENDING	STQind RECEIVED	STQind RENEW/ PENDING	STQind RENEW/ OVERDUE	STQind SHIPPED
STR	STRind PENDING	STRind CANCEL-PENDING	STRind RECEIVED	STRind RENEW/ PENDING	STRind RENEW/ OVERDUE	STRind SHIPPED
REA +			p5 REAind RECEIVED +	REAind + RECEIVED	REAind + RECEIVED	
REA + repeat			REAind + RECEIVED			
REA -			p5 REAind - RECEIVED	REAind - RECEIVED	REAind - OVERDUE	
REA - repeat			REAind - RECEIVED			

Table A.5a — Requester State Table — Tracking Phase (Part 1) *(concluded)*

	PENDING	CANCEL PENDING	RECEIVED	RENEW/ PENDING	RENEW/ OVERDUE	SHIPPED
CHK	CHKind RETURNED	CHKind RETURNED	p5 CHKind RETURNED	CHKind RETURNED	CHKind RETURNED	CHKind RETURNED
CHK repeat						

Table A.5b — Requester State Table — Tracking Phase (Part 2)

	NOT-RCVD/OVERDUE	OVERDUE	RETURNED	LOST	RECALL
RCVreq	RCV (opt) set RETURN var OVERDUE				RCV (opt) RECALL
RCVreq repeat					
RETreq		RET (opt) RETURNED	RET (opt) RETURNED		RET (opt) RETURNED
RETreq repeat			RET (opt) RETURNED		
RENreq		REN RENEW/OVERDUE			
RENreq repeat					
LSTreq	LST LOST	LST LOST	LST LOST	LST LOST	LST LOST
LSTreq repeat				LST LOST	
DAMreq		DAM OVERDUE	DAM RETURNED		DAM RECALL
MSGreq	MSG NOT-RCVD/ OVERDUE	MSG OVERDUE	MSG RETURNED	MSG LOST	MSG RECALL
STQreq	STQ NOT-RCVD/ OVERDUE	STQ OVERDUE	STQ RETURNED	STQ LOST	STQ RECALL
STRreq	STR NOT-RCVD/ OVERDUE	STR OVERDUE	STR RETURNED	STR LOST	STR RECALL
ANS-WS	ANSind-WS NOT-RCVD/ OVERDUE	ANSind-WS OVERDUE	ANSind-WS RETURNED	ANSind-WS LOST	ANSind-WS RECALL
ANS-WS repeat	ANSind-WS NOT-RCVD/ OVERDUE	ANSind-WS OVERDUE	ANSind-WS RETURNED	ANSind-WS LOST	ANSind-WS RECALL
ANS-HP	ANSind-HP NOT-RCVD/ OVERDUE	ANSind-HP OVERDUE	ANSind-HP RETURNED	ANSind-HP LOST	ANSind-HP RECALL
ANS-HP repeat	ANSind-HP NOT-RCVD/ OVERDUE	ANSind-HP OVERDUE	ANSind-HP RETURNED	ANSind-HP LOST	ANSind-HP RECALL
CAR -	CARind - NOT-RCVD/ OVERDUE	CARind - OVERDUE	CARind - RETURNED	CARind - LOST	CARind - RECALL
CAR - repeat	CARind - NOT-RCVD/ OVERDUE	CARind- OVERDUE	CARind - RETURNED	CARind - LOST	CARind - RECALL
SHI	SHIind NOT-RCVD/ OVERDUE	SHIind OVERDUE	SHIind RETURNED	SHIind LOST	SHIind RECALL
SHI repeat	SHIind NOT-RCVD/ OVERDUE	SHIind OVERDUE	SHIind RETURNED	SHIind LOST	SHIind RECALL

Table A.5b — Requester State Table — Tracking Phase (Part 2) *(concluded)*

	NOT-RCVD/OVERDUE	OVERDUE	RETURNED	LOST	RECALL
RCL	RCLind RECALL	RCLind RECALL	RCLind RETURNED	RCLind LOST	RCLind RECALL
RCL repeat			RCLind RETURNED	RCLind LOST	RCLind RECALL
DUE		DUEind OVERDUE	DUEind RETURNED	DUEind LOST	DUEind RECALL
DUE repeat	DUEind NOT-RCVD/ OVERDUE	DUEind OVERDUE	DUEind RETURNED	DUEind LOST	DUEind RECALL
LST	LSTind LOST		LSTind LOST	LSTind LOST	LSTind LOST
LST repeat				LSTind LOST	
DAM			DAMind RETURNED		
MSG	MSGind NOT-RCVD/ OVERDUE	MSGind OVERDUE	MSGind RETURNED	MSGind LOST	MSGind RECALL
STQ	STQind NOT-RCVD/ OVERDUE	STQind OVERDUE	STQind RETURNED	STQind LOST	STQind RECALL
STR	STRind NOT-RCVD/OVERDUE	STRind OVERDUE	STRind RETURNED	STRind LOST	STRind RECALL
REA +			REAind + RETURNED	REAind + LOST	REAind + RECALL
REA + repeat			REAind + RETURNED	REAind + LOST	REAind + RECALL
REA -		REAind - OVERDUE	REAind - RETURNED	REAind - LOST	REAind - RECALL
REA - repeat		REAind - OVERDUE	REAind - RETURNED	REAind - LOST	REAind - RECALL
CHK	CHKind RETURNED	CHKind RETURNED	CHKind RETURNED		CHKind RETURNED
CHK repeat			CHKind RETURNED		

Table A.6 — Responder State Table — Processing Phase

	IDLE	IN-PROCESS	NOT-SUPPLIED	CONDITIONAL	CANCEL-PENDING	CANCELLED	FORWARD
ILL	ILLind set FWD var set CHAIN var set PART var set EXPIRY timer IN-PROCESS	ILLind IN-PROCESS	ILLind NOT-SUPPLIED	ILLind CONDITIONAL			ILLind FORWARD
ILL repeat		ILLind IN-PROCESS	ILLind NOT-SUPPLIED	ILLind CONDITIONAL			ILLind FORWARD
FWDreq		p4 ILL FWD disable EXPIRY timer FORWARD					
FWDreq repeat							ILL FWD FORWARD
ANS-COreq		ANS-CO reset EXPIRY timer CONDITIONAL					
ANS-COreq repeat				ANS-CO CONDITIONAL			
ANS-RYreq		ANS-RY disable EXPIRY timer NOT-SUPPLIED					
ANS-RYreq repeat			ANS-RY NOT-SUPPLIED				
ANS-UNreq		ANS-UN disable EXPIRY timer NOT-SUPPLIED					
ANS-UNreq repeat			ANS-UN NOT-SUPPLIED				
ANS-LPreq		ANS-LP disable EXPIRY timer NOT-SUPPLIED					
ANS-LPreq repeat			ANS-LP NOT-SUPPLIED				

Table A.6 — Responder State Table — Processing Phase *(continued)*

	IDLE	IN-PROCESS	NOT-SUPPLIED	CONDITIONAL	CANCEL-PENDING	CANCELLED	FORWARD
ANS-WSreq		ANS-WS disable EXPIRY timer IN-PROCESS					
ANS-WSreq repeat		ANS-WS IN-PROCESS					
ANS-HPreq		ANS-HP disable EXPIRY timer IN-PROCESS					
ANS-HPreq repeat		ANS-HP IN-PROCESS					
ANS-ESreq		ANS-ES disable EXPIRY timer NOT-SUPPLIED					
ANS-ESreq repeat			ANS-ES NOT-SUPPLIED				
CARreq +					CAR + CANCELLED		
CARreq + repeat						CAR + CANCELLED	
CARreq -					CAR - IN-PROCESS		
CARreq - repeat		CAR - IN-PROCESS					
SHIreq		SHI (opt) set RETURN var disable EXPIRY timer SHIPPED					
SHIreq repeat							
MSGreq		MSG IN-PROCESS	MSG NOT-SUPPLIED	MSG CONDITIONAL	MSG CANCEL-PENDING	MSG CANCELLED	MSG FORWARD
STQreq		STQ IN-PROCESS	STQ NOT-SUPPLIED	STQ CONDITIONAL	STQ CANCEL-PENDING	STQ CANCELLED	STQ FORWARD
STRreq		STR IN-PROCESS	STR NOT-SUPPLIED	STR CONDITIONAL	STR CANCEL-PENDING	STR CANCELLED	STR FORWARD
C-REP +		C-REPind + IN-PROCESS	C-REPind + NOT-SUPPLIED	C-REPind + reset EXPIRY timer IN-PROCESS			

Table A.6 — Responder State Table — Processing Phase *(concluded)*

	IDLE	IN-PROCESS	NOT-SUPPLIED	CONDITIONAL	CANCEL-PENDING	CANCELLED	FORWARD
C-REP + repeat		C-REPind + IN-PROCESS	C-REPind + NOT-SUPPLIED				
C-REP -			C-REPind - NOT-SUPPLIED	C-REPind - NOT-SUPPLIED			
C-REP - repeat			C-REPind - NOT-SUPPLIED				
CAN		p7 CANind CANCEL-PENDING	CANind NOT-SUPPLIED	CANind CANCEL-PENDING	CANind CANCEL-PENDING	CANind CANCELLED	CANind FORWARD
CAN		^p7 CANind IN-PROCESS	CANind NOT-SUPPLIED	CANind CANCEL-PENDING	CANind CANCEL-PENDING	CANind CANCELLED	CANind FORWARD
CAN repeat		CANind IN-PROCESS	CANind NOT-SUPPLIED	CANind CANCEL-PENDING	CANind CANCEL-PENDING	CANind CANCELLED	CANind FORWARD
MSG		MSGind IN-PROCESS	MSGind NOT-SUPPLIED	MSGind CONDITIONAL	MSGind CANCEL-PENDING	MSGind CANCELLED	MSGind FORWARD
STQ		STQind IN-PROCESS	STQind NOT-SUPPLIED	STQind CONDITIONAL	STQind CANCEL-PENDING	STQind CANCELLED	STQind FORWARD
STR		STRind IN-PROCESS	STRind NOT-SUPPLIED	STRind CONDITIONAL	STRind CANCEL-PENDING	STRind CANCELLED	STRind FORWARD
EXPIRY Timeout		EXPind EXP NOT-SUPPLIED		EXPind EXP NOT-SUPPLIED			

Table A.7 — Responder State Table — Tracking Phase

	SHIPPED	RENEW/ PENDING	RENEW/ OVERDUE	OVERDUE	RECALL	CHECKED-IN	LOST
SHIreq							
SHIreq repeat	SHI SHIPPED						
CHKreq	p5 CHK (opt) CHECKED-IN	CHK (opt) CHECKED-IN	CHK (opt) CHECKED-IN	CHK (opt) CHECKED-IN	CHK (opt) CHECKED-IN		
CHKreq repeat						CHK (opt) CHECKED-IN	
RCLreq	p5 RCL RECALL	RCL RECALL	RCL RECALL	RCL RECALL			
RCLreq repeat					RCL RECALL		
DUEreq	p5 DUE OVERDUE	DUE RENEW/OVERDUE		p8 DUE OVERDUE			
DUEreq repeat			DUE RENEW/OVERDUE	DUE OVERDUE			
LSTreq	LST LOST	LST LOST	LST LOST	LST LOST	LST LOST		LST LOST
LSTreq repeat							LST LOST
DAMreq						DAM CHECKED-IN	
MSGreq	MSG SHIPPED	MSG RENEW / PENDING	MSG RENEW/ OVERDUE	MSG OVERDUE	MSG RECALL	MSG CHECKED-IN	MSG LOST
STQreq	STQ SHIPPED	STQ RENEW/ PENDING	STQ RENEW/ OVERDUE	STQ OVERDUE	STQ RECALL	STQ CHECKED-IN	STQ LOST
STRreq	STR SHIPPED	STR RENEW/ PENDING	STR RENEW/ OVERDUE	STR OVERDUE	STR RECALL	STR CHECKED-IN	STR LOST
REAreq +		REA + SHIPPED	REA + SHIPPED				
REAreq + repeat	REA + SHIPPED						
REAreq -		REA - SHIPPED	REA - OVERDUE				
REAreq - repeat	REA - SHIPPED			REA - OVERDUE			
ILL	ILLind SHIPPED			ILLind OVERDUE	ILLind RECALL	ILLind CHECKED-IN	ILLind LOST

Table A.7 — Responder State Table — Tracking Phase *(concluded)*

ILL repeat	ILLind SHIPPED			ILLind OVERDUE	ILLind RECALL	ILLind CHECKED-IN	ILLind LOST
CAN	CANind SHIPPED	CANind RENEW/ PENDING	CANind RENEW/ OVERDUE	CANind OVERDUE	CANind RECALL	CANind CHECKED-IN	CANind LOST
CAN repeat	CANind SHIPPED	CANind RENEW/ PENDING	CANind RENEW/ OVERDUE	CANind OVERDUE	CANind RECALL	CANind CHECKED-IN	CANind LOST
RCV	RCVind SHIPPED	RCVind RENEW/ PENDING	RCVind RENEW/ OVERDUE	RCVind OVERDUE	RCVind RECALL	RCVind CHECKED-IN	RCVind LOST
RCV repeat	RCVind SHIPPED	RCVind RENEW/ PENDING	RCVind RENEW/ OVERDUE	RCVind OVERDUE	RCVind RECALL	RCVind CHECKED-IN	RCVind LOST
RET	RETind SHIPPED	RETind RENEW/ PENDING	RETind RENEW/ OVERDUE	RETind OVERDUE	RETind RECALL	RETind CHECKED-IN	RETind LOST
RET repeat	RETind SHIPPED	RETind RENEW/ PENDING	RETind RENEW/ OVERDUE	RETind OVERDUE	RETind RECALL	RETind CHECKED-IN	RETind LOST
REN	p7 RENind RENEW/PENDING	RENind RENEW/ PENDING	RENind RENEW/OVERDUE	RENind RENEW/ OVERDUE	RENind RECALL	RENind CHECKED-IN	RENind LOST
REN	^p7 RENind SHIPPED	RENind RENEW/ PENDING	RENind RENEW/OVERDUE	RENind RENEW/ OVERDUE	RENind RECALL	RENind CHECKED-IN	RENind LOST
REN repeat	RENind SHIPPED	RENind RENEW/ PENDING	RENind RENEW/ OVERDUE	RENind OVERDUE	RENind RECALL	RENind CHECKED-IN	RENind LOST
LST	LSTind LOST	LSTind LOST	LSTind LOST	LSTind LOST	LSTind LOST	LSTind CHECKED-IN	LSTind LOST
LST repeat						LSTind CHECKED-IN	LSTind LOST
DAM	DAMind SHIPPED	DAMind RENEW/ PENDING	DAMind RENEW/ OVERDUE	DAMind OVERDUE	DAMind RECALL	DAMind CHECKED-IN	DAMind LOST
MSG	MSGind SHIPPED	MSGind RENEW/ PENDING	MSGind RENEW/ OVERDUE	MSGind OVERDUE	MSGind RECALL	MSGind CHECKED-IN	MSGind LOST
STQ	STQind SHIPPED	STQind RENEW/ PENDING	STQind RENEW/ OVERDUE	STQind OVERDUE	STQind RECALL	STQind CHECKED-IN	STQind LOST
STR	STRind SHIPPED	STRind RENEW/ PENDING	STRind RENEW/ OVERDUE	STRind OVERDUE	STRind RECALL	STRind CHECKED-IN	STRind LOST

Table A.8 — Intermediary State Table — Requester Role

	IDLE	PENDING	NOT-SUPPLIED	CONDITIONAL	CANCEL-PENDING	CANCELLED	SHIPPED
ILLreq	p2 or p3 ILL update CHAIN var update PART var PENDING						
ILLreq repeat		ILL PENDING	ILL NOT-SUPPLIED	ILL CONDITIONAL			ILL SHIPPED
C-REPreq +			C-REP + NOT-SUPPLIED	C-REP + PENDING			
C-REPreq + repeat		C-REP + PENDING	C-REP + NOT-SUPPLIED				
C-REPreq -			C-REP - NOT-SUPPLIED	C-REP - NOT-SUPPLIED			
C-REPreq - repeat			C-REP - NOT-SUPPLIED				
CANreq		CAN CANCEL-PENDING	CAN NOT-SUPPLIED	CAN CANCEL-PENDING			CAN SHIPPED
CANreq repeat		CAN PENDING	CAN NOT-SUPPLIED		CAN CANCEL-PENDING	CAN CANCELLED	CAN SHIPPED
RCVreq		RCV (opt) set RETURN var SHIPPED			RCV (opt) set RETURN var SHIPPED		p6 RCV (opt) set RETURN var SHIPPED
RCVreq repeat							RCV (opt) SHIPPED
RETreq		RET SHIPPED					p6 RET SHIPPED
RETreq repeat							RET SHIPPED
RENreq		REN SHIPPED					p6 REN SHIPPED
RENreq repeat							REN SHIPPED
LSTreq		LST SHIPPED					p6 LST SHIPPED
LSTreq repeat							LST SHIPPED

Table A.8 — Intermediary State Table — Requester Role *(continued)*

	IDLE	PENDING	NOT-SUPPLIED	CONDITIONAL	CANCEL-PENDING	CANCELLED	SHIPPED
DAMreq		DAM SHIPPED			DAM SHIPPED		p6 DAM SHIPPED
MSGreq		MSG PENDING	MSG NOT-SUPPLIED	MSG CONDITIONAL	MSG CANCEL-PENDING	MSG CANCELLED	MSG SHIPPED
STQreq		STQ PENDING	STQ NOT-SUPPLIED	STQ CONDITIONAL	STQ CANCEL-PENDING	STQ CANCELLED	STQ SHIPPED
STRreq		STR PENDING	STR NOT-SUPPLIED	STR CONDITIONAL	STR CANCEL-PENDING	STR CANCELLED	STR SHIPPED
FWD		FWDind PENDING			FWDind PENDING		
FWD repeat		FWDind PENDING					
ANS-CO		p7 ANSindCO CONDITIONAL	ANSind-CO NOT-SUPPLIED	ANSind-CO CONDITIONAL	ANSind-CO CANCEL-PENDING	ANSind-CO CANCELLED	
ANS-CO		^p7 ANSind-CO PENDING	ANSind-CO NOT-SUPPLIED	ANSind-CO CONDITIONAL	ANSind-CO CANCEL-PENDING	ANSind-CO CANCELLED	
ANS-CO repeat		ANSind-CO PENDING	ANSind-CO NOT-SUPPLIED	ANSind-CO CONDITIONAL	ANSind-CO CANCEL-PENDING	ANSind-CO CANCELLED	
ANS-RY		ANSind-RY NOT-SUPPLIED	ANSind-RY NOT-SUPPLIED		ANSind-RY NOT-SUPPLIED		
ANS-RY repeat			ANSind-RY NOT-SUPPLIED				
ANS-UN		ANSind-UN NOT-SUPPLIED	ANSind-UN NOT-SUPPLIED		ANSind-UN NOT-SUPPLIED		
ANS-UN repeat			ANSind-UN NOT-SUPPLIED				
ANS-LP		ANSind-LP NOT-SUPPLIED	ANSind-LP NOT-SUPPLIED		ANSind-LP NOT-SUPPLIED		
ANS-LP repeat			ANSind-LP NOT-SUPPLIED				
ANS-WS		ANSind-WS PENDING	ANSind-WS NOT-SUPPLIED	ANSind-WS CONDITIONAL	ANSind-WS CANCEL-PENDING	ANSind CANCELLED	WSANSind-WS SHIPPED
ANS-WS repeat		ANSind-WS PENDING	ANSind-WS NOT-SUPPLIED	ANSind-WS CONDITIONAL	ANSind-WS CANCEL-PENDING	ANSind-WS CANCELLED	ANSind-WS SHIPPED

Table A.8 — Intermediary State Table — Requester Role *(continued)*

	IDLE	PENDING	NOT-SUPPLIED	CONDITIONAL	CANCEL-PENDING	CANCELLED	SHIPPED
ANS-HP		ANSind-HP PENDING	ANSind-HP NOT-SUPPLIED	ANSind-HP CONDITIONAL	ANSind-HP CANCEL-PENDING	ANSind-HP CANCELLED	ANSind-HP SHIPPED
ANS-HP repeat		ANSind-HP PENDING	ANSind-HP NOT-SUPPLIED	ANSind-HP CONDITIONAL	ANSind-HP CANCEL-PENDING	ANSind-HP CANCELLED	ANSind-HP SHIPPED
ANS-ES		ANSind-ES NOT-SUPPLIED	ANSind-ES NOT-SUPPLIED		ANSind-ES NOT-SUPPLIED		
ANS-ES repeat			ANSind-ES NOT-SUPPLIED				
CAR +					CARind + CANCELLED	CARind + CANCELLED	
CAR + repeat						CARind + CANCELLED	
CAR -		CARind - PENDING			CARind - PENDING		CARind - SHIPPED
CAR - repeat		CARind - PENDING					CARind - SHIPPED
SHI		SHIind SHIPPED			SHIind SHIPPED		SHIind SHIPPED
SHI repeat							SHIind SHIPPED
DUE		DUEind SHIPPED					p6 DUEind SHIPPED
DUE repeat							DUEind SHIPPED
REA +							p6 REAind + SHIPPED
REA + repeat							REAind + SHIPPED
REA -							p6 REAind - SHIPPED
REA - repeat							REAind - SHIPPED
LST		LSTind SHIPPED			LSTind SHIPPED		p6 LSTind SHIPPED
LST repeat							LSTind SHIPPED

Table A.8 — Intermediary State Table — Requester Role *(concluded)*

	IDLE	PENDING	NOT-SUPPLIED	CONDITIONAL	CANCEL-PENDING	CANCELLED	SHIPPED
DAM							p6 DAMind SHIPPED
RCL		RCLind SHIPPED					p6 RCLind SHIPPED
RCL repeat							RCLind SHIPPED
MSG		MSGind PENDING	MSGind NOT-SUPPLIED	MSGind CONDITIONAL	MSGind CANCEL-PENDING	MSGind CANCELLED	MSGind SHIPPED
STQ		STQind PENDING	STQind NOT-SUPPLIED	STQind CONDITIONAL	STQind CANCEL-PENDING	STQind CANCELLED	STQind SHIPPED
STR		STRind PENDING	STRind NOT-SUPPLIED	STRind CONDITIONAL	STRind CANCEL-PENDING	STRind CANCELLED	STRind SHIPPED
EXP		EXPind NOT-SUPPLIED	EXPind NOT-SUPPLIED	EXPind NOT-SUPPLIED	EXPind NOT-SUPPLIED		
CHK		CHKind SHIPPED			CHKind SHIPPED		p6 CHKind SHIPPED
CHK repeat							CHKind SHIPPED

Table A.9 — Intermediary State Table — Responder Role

	IDLE	IN-PROCESS	NOT-SUPPLIED	CONDITIONAL	CANCEL-PENDING	CANCELLED	FORWARD	SHIPPED
ILL	ILLind set FWD var set CHAIN var set PART var set EXPIRY timer IN-PROCESS	ILLind IN-PROCESS	ILLind NOT-SUPPLIED	ILLind CONDITIONAL			ILLind FORWARD	ILLind SHIPPED
ILL repeat		ILLind IN-PROCESS	ILLind NOT-SUPPLIED	ILLind CONDITIONAL			ILLind FORWARD	ILLind SHIPPED
FWDreq		p4 ILL FWD disable EXPIRY timer FORWARD						
FWDreq repeat							ILL FWD FORWARD	
ANS-COreq		ANS-CO reset EXPIRY timer CONDITIONAL			ANS-CO CANCEL-PENDING	ANS-CO CANCELLED		
ANS-COreq repeat		ANS-CO disable EXPIRY timer IN-PROCESS	ANS-CO NOT-SUPPLIED	ANS-CO CONDITIONAL	ANS-CO CANCEL-PENDING	ANS-CO CANCELLED		
ANS-RYreq		ANS-RY disable EXPIRY timer NOT-SUPPLIED			ANS-RY disable EXPIRY timer NOT-SUPPLIED			
ANS-RYreq repeat			ANS-RY NOT-SUPPLIED					
ANS-UNreq		ANS-UN disable EXPIRY timer NOT-SUPPLIED			ANS-UN disable EXPIRY timer NOT-SUPPLIED			
ANS-UNreq repeat			ANS-UN NOT-SUPPLIED					

Table A.9 — Intermediary State Table — Responder Role *(continued)*

	IDLE	IN-PROCESS	NOT-SUPPLIED	CONDITIONAL	CANCEL-PENDING	CANCELLED	FORWARD	SHIPPED
ANS-LPreq		ANS-LP disable EXPIRY timer NOT-SUPPLIED			ANS-LP disable EXPIRY timer NOT-SUPPLIED			
ANS-LPreq repeat			ANS-LP NOT-SUPPLIED					
ANS-WSreq		ANS-WS disable EXPIRY timer IN-PROCESS	ANS-WS NOT-SUPPLIED	ANS-WS CONDITIONAL	ANS-WS CANCEL-PENDING	ANS-WS CANCELLED		ANS-WS SHIPPED
ANS-WSreq repeat		ANS-WS IN-PROCESS	ANS-WS NOT-SUPPLIED	ANS-WS CONDITIONAL	ANS-WS CANCEL-PENDING	ANS-WS CANCELLED		ANS-WS SHIPPED
ANS-HPreq		ANS-HP disable EXPIRY timer IN-PROCESS	ANS-HP NOT-SUPPLIED	ANS-HP CONDITIONAL	ANS-HP CANCEL-PENDING	ANS-HP CANCELLED		ANS-HP SHIPPED
ANS-HPreq repeat		ANS-HP IN-PROCESS	ANS-HP NOT-SUPPLIED	ANS-HP CONDITIONAL	ANS-HP CANCEL-PENDING	ANS-HP CANCELLED		ANS-HP SHIPPED
ANS-ESreq		ANS-E disable EXPIRY timer NOT-SUPPLIED			ANS-ES disable EXPIRY timer NOT-SUPPLIED			
ANS-ESreq repeat			ANS-ES NOT-SUPPLIED					
CARreq +					CAR + CANCELLED			
CARreq + repeat						CAR + CANCELLED		
CARreq -					CAR - IN-PROCESS			CAR - SHIPPED
CARreq - repeat		CAR - IN-PROCESS						CAR - SHIPPED
SHIreq		SHI set RETURN var disable EXPIRY timer SHIPPED			SHI set RETURN variable disable EXPIRY timer SHIPPED			SHI SHIPPED
SHIreq repeat								SHI SHIPPED

Table A.9 — Intermediary State Table — Responder Role *(continued)*

	IDLE	IN-PROCESS	NOT-SUPPLIED	CONDITIONAL	CANCEL-PENDING	CANCELLED	FORWARD	SHIPPED
MSGreq		MSG IN-PROCESS	MSG NOT-SUPPLIED	MSG CONDITIONAL	MSG CANCEL-PENDING	MSG CANCELLED	MSG FORWARD	MSG SHIPPED
STQreq		STQ IN-PROCESS	STQ NOT-SUPPLIED	STQ CONDITIONAL	STQ CANCEL-PENDING	STQ CANCELLED	STQ FORWARD	STQ SHIPPED
STRreq		STR IN-PROCESS	STR NOT-SUPPLIED	STR CONDITIONAL	STR CANCEL-PENDING	STR CANCELLED	STR FORWARD	STR SHIPPED
CHKreq		CHK (opt) SHIPPED			CHK (opt) SHIPPED			p6 CHK (opt) SHIPPED
CHKreq repeat								CHK (opt) SHIPPED
DUEreq		DUE SHIPPED						p6 DUE SHIPPED
DUEreq repeat								DUE SHIPPED
REAreq +								p6 REA + SHIPPED
REAreq + repeat								REA + SHIPPED
REAreq -								p6 REA - SHIPPED
REAreq - repeat								REA - SHIPPED
RCLreq		RCL SHIPPED						p6 RCL SHIPPED
RCLreq repeat								RCL SHIPPED
LSTreq		LST SHIPPED			LST SHIPPED			p6 LST SHIPPED
LSTreq repeat								LST SHIPPED
DAMreq								p6 DAM SHIPPED

Table A.9 — Intermediary State Table — Responder Role *(continued)*

	IDLE	IN-PROCESS	NOT-SUPPLIED	CONDITIONAL	CANCEL-PENDING	CANCELLED	FORWARD	SHIPPED
C-REP +		C-REPind + IN-PROCESS	C-REPind + NOT-SUPPLIED	C-REPind + reset EXPIRY timer IN-PROCESS				
C-REP + repeat		C-REPind + IN-PROCESS	C-REPind + NOT-SUPPLIED					
C-REP -			C-REPind - NOT-SUPPLIED	C-REPind - NOT-SUPPLIED				
C-REP repeat -			C-REPind - NOT-SUPPLIED					
CAN		P7 CANind CANCEL-PENDING	CANind NOT-SUPPLIED	CANind CANCEL-PENDING	CANind CANCEL-PENDING	CANind CANCELLED	CANind FORWARD	CANind SHIPPED
CAN		^P7 CANind IN-PROCESS	CANind NOT-SUPPLIED	CANind CANCEL-PENDING	CANind CANCEL-PENDING	CANind CANCELLED	CANind FORWARD	CANind SHIPPED
CAN repeat		CANind IN-PROCESS	CANind NOT-SUPPLIED	CANind CANCEL-PENDING	CANind CANCEL-PENDING	CANind CANCELLED	CANind FORWARD	CANind SHIPPED
MSG		MSGind IN-PROCESS	MSGind NOT-SUPPLIED	MSGind CONDITIONAL	MSGind CANCEL-PENDING	MSGind CANCELLED	MSGind FORWARD	MSGind SHIPPED
STQ		STQind IN-PROCESS	STQind NOT-SUPPLIED	STQind CONDITIONAL	STQind CANCEL-PENDING	STQind CANCELLED	STQind FORWARD	STQind SHIPPED
STR		STRind IN-PROCESS	STRind NOT-SUPPLIED	STRind CONDITIONAL	STRind CANCEL-PENDING	STRind CANCELLED	STRind FORWARD	STRind SHIPPED
RCV		RCVind SHIPPED			RCVind SHIPPED			p6 RCVind SHIPPED
RCV repeat								RCVind SHIPPED
RET		RETind SHIPPED						p6 RETind SHIPPED
RET repeat								RETind SHIPPED
REN		RENind SHIPPED						p6 RENind SHIPPED

Table A.9 — Intermediary State Table — Responder Role *(concluded)*

	IDLE	IN-PROCESS	NOT-SUPPLIED	CONDITIONAL	CANCEL-PENDING	CANCELLED	FORWARD	SHIPPED
REN repeat								RENind SHIPPED
LST		LSTind SHIPPED						p6 LSTind SHIPPED
LST repeat								LSTind SHIPPED
DAM		DAMind SHIPPED			DAMind SHIPPED			p6 DAMind SHIPPED
EXPIRY Timeout		EXPind EXP NOT-SUPPLIED		EXPind EXP NOT-SUPPLIED				

Annex B
(normative)
Transfer Syntax

More than one encoding scheme can be applied to the values of data types that are defined using ASN.1. The mandatory encoding scheme is the Basic Encoding Rules for Abstract Syntax Notation One defined in ISO/IEC 8825. An additional possibility is the encoding scheme defined by the EDIFACT standard, ISO 9735.

This annex defines rules for encoding ILL APDUs using EDIFACT. If EDIFACT encoding is used as the ILL transfer syntax (i.e. as a supplement to ASN.1), the encoding rules provided in this annex must be applied.

All the data transmitted between two sites at one time is defined to be an interchange. An interchange is in turn made up of one or more messages. Each ILL APDU to be transferred is encoded as a single EDIFACT message; each EDIFACT message contains exactly one ILL APDU.

B.1 EDIFACT Features Not Supported

The following EDIFACT features are not supported. They are not to be used in an interchange.

-Use of functional groups.

NOTE — Functional groups are used to convey groups of messages of the same type; this is an unnecessary feature that is distinct from the capability to include multiple messages, possibly of different types, within a single transmission. This latter capability is also available and can be used.

-Explicit indication of segment nesting.

B.2 Character Set and Syntax Level

Of the two syntax levels defined by ISO 9735, level B shall be used with the corresponding character set. However, the delimiter characters as defined in level A will be applied, i.e. the characters "+':?". Of the two choices defined for ILL-String in clause 9, only the EDIFACT-String can be used with the EDIFACT encoding.

B.3 EDIFACT Interchange

EDIFACT defines an interchange as being the unit of data that is transferred between two sites. ILL APDUs are encoded as part of an EDIFACT interchange.

An interchange is made up of a number of segments. Each segment begins with a three character segment code, followed in sequence by the segment code delimiter, the segment content, and the end of segment delimiter. The end of segment delimiter is defined to be the "'" character. The segment code delimiter is defined to be the "+" character.

The segment content is made up of one or more data elements. Data elements are separated from each other by the "+" character. A data element is made up of a string of EDIFACTString characters, or by two or more sub-elements. The order of the data elements and sub-elements within a segment is fixed.

Optional data elements which contain no data need not be transmitted if they appear as the last data element of a segment. For example, a segment with segment code "ABC", made up of three data elements, the first containing the data "12345" and the second and third optional and empty, can be transmitted in any of the three following forms:

ABC+12345++'

ABC+12345+'

ABC+12345'

Sub-elements are separated from each other by the ":" character. A sub-element is made up of a string of EDIFACTString characters. It may be repeated within an element up to the number of times specified by the corresponding ASN.1 specification in clause 9.

Optional sub-elements which contain no data need not be transmitted if they appear as the last sub-elements of a data element. For example, a data element which is preceded and followed by data elements of the same segment and which is made up of two sub-elements, the first containing the data "12345" and the second optional and empty, can be transmitted in either of the two following forms:

...+12345:+...

...+12345+...

The character "?" is an escape character. If the data element or sub-element is to contain one of the special characters "+':?" then the special character must be preceded by a "?".

The following is the definition of an interchange in Bacchus-Naur form. (All literal characters are enclosed by double-quotation marks).

interchange ::= segment_list

segment_list ::= segment | segment segment_list

segment ::= segment_code "+" segment_content "'"

segment_code ::= alpha alpha alpha

segment_content ::= data_element | data_element "+" segment_content

data_element ::= content_string | content_string ":" data_element

content_string ::= string_part | string_part content_string

string_part ::= normal_string | special_character

special_character ::= "??" | "?+" | "?:" | "?'"

normal_string ::= {A variable length string of 0 or more EDIFACTString characters excluding "?+:'"}

alpha ::= {any character a-z or A-Z}

Each interchange contains exactly one Service String Advice segment (segment code UNA) and one Interchange Header segment (segment code UNB), which must be the first and second segments of the interchange, respectively.

Following the Service String Advice and the Interchange Header segments, the interchange contains one or more messages. Each message contains exactly one Message Header segment (segment code UNH) and it must be the first segment of a message.

A message contains exactly one Message Trailer segment (segment code UNT) and it must be the last segment of the message.

A Message Header segment, the next Message Trailer segment, and all segments between the Message Header and the Message Trailer constitute a single message.

ASN.1 data types, including the APDUs, that are defined as SEQUENCE data types are mapped to a sequence of EDIFACT segments. The order of the segments is important.

The segment following the last message must be an Interchange Trailer segment (segment code UNZ). There is exactly one Interchange Trailer segment per interchange, and it must be the last segment of the interchange.

B.4 Segments

The segments which make up an interchange are classified as control segments or as data segments.

The control segments are the Service String Advice, Interchange Header, Message Header, Message Trailer, and Interchange Trailer. These segments do not contain any ILL APDU content information as specified in clause 9. The structure of these segments is given in clause B.5.

The data segments are those segments of a message between the Message Header and the Message Trailer. The structure of the data segments is created by applying the rules of clause B.6 to the ASN.1 specification of clause 9.

The data value of an ILL APDU is completely contained within the data segments of a single message.

B.5 Control Segment Encoding

B.5.1 Control Segment Construction

This subclause defines the construction of the control segments. Each segment is assigned a name and a three character segment code used for identification within the EDIFACT interchange. A description of the segment is provided, followed by a list of data elements contained within the segment. These data elements are described in B.5.2. All data elements of the control segments are mandatory.

Segment name:	Service String Advice
Segment code:	UNA
Description:	Defines the characters selected for use as delimiters and indicators in an interchange. This must be the first segment in the interchange and must appear immediately before the Interchange Header (UNB)
Representation:	It must be represented as: UNA:+.?(space)'

Segment name:	Interchange Header
Segment code:	UNB
Description:	Indicates the start of an interchange. Must be the second segment of the interchange.

Data elements:	Syntax Identifier
	Interchange Sender
	Interchange Recipient
	Date/Time of Preparation
	Interchange Control Reference
	Communications Agreement ID

Segment name:	Interchange Trailer
Segment code:	UNZ
Description:	Indicates the end of an interchange. Must be the last segment of an interchange.
Data elements:	Interchange Control Count
	Interchange Control Reference

Segment name:	Message Header
Segment code:	UNH
Description:	Indicates the start of a message.
Data elements:	Message Reference
	Number Message Identifier

Segment name:	Message Trailer
Segment code:	UNT
Description:	Indicates the end of a message.
Data elements:	Number Of Segments In The Message
	Message Reference Number

B.5.2 Data Elements of Control Segments

Data Element:	Syntax Identifier
Description:	Identifies the syntax rules. Consists of two sub-elements: Identifier and Version Number. Both sub-elements are mandatory.
Representation:	Identifier is represented by the four characters "UNOB". Version Number is represented by the single character "1".

Data Element:	Interchange Sender
Description:	Identifies the sender of the interchange. Contains one of the components of the ASN.1 system-id type.
Representation:	EDIFACTString, length 35, variable.

Data Element:	Interchange Recipient
Description:	Identifies the recipient of the interchange. Contains one of the components of the ASN.1 system-id type.
Representation:	EDIFACTString, length 35, variable

Data Element:	Date/Time of Preparation
Description:	Date and time of interchange preparation. Consists of two sub-elements:Date and Time. The date is six characters long in the format YYMMDD, while the time is four characters long in the format HHMM.
Representation:	Date:Numeric string, length = 6, fixed. Time:Numeric string, length = 4, fixed.
Data Element:	Interchange Control Reference
Description:	A unique reference assigned to the interchange by the sender.
Representation:	EDIFACTString, length = 14, variable.
Data Element:	Communications Agreement ID
Description:	Identifies the communication agreement which governs the information content of the interchange. It has the value ISO-10161-ILL-1 where the 10161 represents the ISO number of the ILL protocol standard. This identifier corresponds to the ILL APDU definition identifier specified in clause 9 of this part of ISO 10161.
Data Element:	Interchange Control Count
Description:	Total number of messages in the interchange.
Representation:	Numeric string, length = 6, variable.
Data Element:	Message Identifier
Description:	Identifies the type and version of ILL APDU which is contained within the message. The APDU type is a 6 character string as defined in B.6.1. The version number is a three character numeric string.
Representation:	Type -EDIFACTString, length = 6, fixed. Version Number -EDIFACTString, length = 3, variable.
Data Element:	Message Reference Number
Description:	Provides a unique message reference. It is not related to any ILL APDU types and, therefore, is specific to the EDIFACT encoding.
Representation:.	EDIFACTString, length = 14, variable
Data Element:	Number Of Segments In The Message
Description:	A count of the total number of segments in the message, including the Message Header and the Message Trailer.
Representation:	Numeric string, length = 6, variable.

B.6 Encoding ILL APDUs using EDIFACT

B.6.1 Mapping ASN.1 Data Types to Messages

Each message within a EDIFACT interchange corresponds to a single ILL APDU. The following table lists the ASN.1 types which are mapped onto EDIFACT messages (i.e. ILL APDUs) and the Message Identifier value to which they correspond.

ASN.1 TYPE	EDIFACT MESSAGE IDENTIFIER VALUE
Ill-Request	ILLREQ
Forward-Notification	FORNOT
Shipped	SHIPED
ILL-Answer	ILLANS
Conditional-Reply	CONREP
Cancel	CANCEL
Cancel-Reply	CNLREP
Received	RCEIVD
Recall	RECALL
Returned	RETRND
Checked-In	CHKDIN
Overdue	OVERDU
Renew	RENEWL
Renew-Answer	RENANS
Lost	LOSTIT
Damaged	DAMAGE
Message	MESSAG
Status-Query	STATQY
Status-Or-Error-Report	STATRP
Expired	EXPIRD

B.6.2 Mapping ASN.1 Types to EDIFACT Segments

This subclause provides a mapping from the ASN.1 types into EDIFACT segments, data elements, and sub-elements. This mapping makes use of implicit nesting and implicit repeated segments.

Each EDIFACT segment is given the name of the corresponding ASN.1 type identifier, i.e. if a type identifier in the ASN.1 specification of clause 9 is defined as a segment name within this subclause, then the corresponding ASN.1 type is represented in EDIFACT as a segment. The segment is optional if the corresponding type in the ASN.1 specification is declared optional.

Following the segment name is the three character segment code used to identify the segment within the EDIFACT interchange.

Following the segment code is a list of the data elements which make up the segment. The names of these data elements correspond to the ASN.1 value type identifiers, i.e. if an ASN.1 type identifier of clause 9 appears in the data element list of this subclause then it is encoded in EDIFACT as an EDIFACT data element. The data element is optional if the corresponding type in the ASN.1 specification is declared optional.

An ASN.1 type mapped to an EDIFACT data element either corresponds to an ASN.1 simple type (ENUMERATED, INTEGER, EDIFACT-String, BOOLEAN or OBJECT IDENTIFIER), or contains EDIFACT sub-elements.

If an EDIFACT data element is made up of sub-elements, then these sub-elements are listed after the data element, and are enclosed in braces ({ }), except in the case of data elements defined as EXTERNAL, in which case the structure is to be defined externally. The names of the EDIFACT sub-elements correspond to the ASN.1 type identifiers, i.e. if the ASN.1 type

identifier appears in the sub-element list, then it is encoded in EDIFACT as an EDIFACT sub-element. The sub-element is optional if the corresponding type in the ASN.1 specification is declared optional.

An ASN.1 type mapped into an EDIFACT sub-element must correspond to an ASN.1 simple type (ENUMERATED, INTEGER, EDIFACT-String, BOOLEAN or OBJECT IDENTIFIER).

The order of the listing of the data elements corresponds to the order in which these data elements must appear within the segment.

The order of the listing of the sub-elements corresponds to the order in which these sub-elements must appear within the data element.

Any default values to be assumed if the data element or sub-element are not present are given in the ASN.1 specification.

ASN.1 types defined as EXTERNAL are mapped to EDIFACT data elements with the first sub-element defined as an object identifier. The remaining sub-elements are defined externally, but must correspond to the ASN.1 simple types supported in this specification.

Segment Name: account-number

Segment Code: ACN

Data Elements: account-number

Segment Name: already-tried-list

Segment Code: ATL

Data Elements: --none; this segment includes the
--subordinate segment

system-id

--which is repeated for each entry in the
--list

Segment Name: answer

Segment Code: ANS

Data Elements: answer

Segment Name: billing-address

Segment Code: BAD

Data Elements: --none; this segment comprises other
--segments rather than actual data
--elements. It includes the following
--segments:

postal-address

system-address

Segment Name: client-id

Segment Code: CID

Data Elements: client-name

client-status

client-identifier

Segment Name: conditional-results

Segment Code: CRE

Data Elements: --none; this segment comprises other
--segments rather than actual data
--elements. It includes the following
--segments:

conditions
date-for-reply
location-info
--this last subordinate segment may be
--repeated

Segment Name: conditions

Segment Code: CON

Data Elements: conditions

Segment Name: copyright-compliance

Segment Code: COC

Data Elements: copyright-compliance

Segment Name: correlation-information

Segment Code: COI

Date Elements: correlation-information

Segment Name: cost-estimate

Segment Code: CST

Data Elements: cost-estimate

Segment Name: cost-info-type

Segment Code: CIT

Data Elements: account-number
maximum-cost
{
currency-code
monetary-value
}
reciprocal-agreement
will-pay-fee
payment-provided

Segment Name: current-state

Segment Code: CUS

Data Elements: current-state

Segment Name: date-checked-in

Segment Code: DCI

Data Elements: date-checked-in

Segment Name: date-due

Segment Code: DUE

Data Elements: date-due-field
renewable

Segment name: date-for-reply

Segment code: DFR

Data Elements: date-for-reply

Segment Name: date-received

Segment Code: DRC

Data Elements: date-received

Segment Name: date-returned

Segment Code: DRT

Data Elements: date-returned

Segment Name: delivery-address

Segment Code: DAD

Data Elements:
-- none; this segment comprises other
-- segments rather than actual data
-- elements. It includes the following
-- segments:

postal-address

system-address

Segment Name: [delivery-service]

-- Note this segment has been renamed
-- physical-delivery. To maintain
-- compatibility with existing systems, the
-- segment code remains DLS. Where an
-- APDU requires a choice of physical-
-- delivery or electronic-delivery-service,
-- one DLS or (repeated) EDY segments
-- will be used.

Segment Name: desired-due-date

Segment Code: DDD

Data Elements: desired-due-date

Segment Name: electronic-delivery-service

Segment Code: EDY

Data Elements: e-delivery-mode

e-delivery-parameters

document-type-id

document-type-parameters

e-delivery-description

--the next three elements together form
--the e-delivery-details
--only one of e-delivery-address or
--e-delivery-id will be present

e-delivery-address

{
telecom-service-identifier
telecom-service-address
}

--the next two elements together form the
--e-delivery-id

person-or-institution-symbol

{
person-symbol
institution-symbol
--only one of these sub-elements may
--be present
}

name-of-person-or-institution

{
name-of-person
name-of-institution
--only one of these sub-elements may
--be present
}

name-or-code

delivery-time

--this segment may be repeated

Segment Name: error-report

Segment Code: ERR

Data Elements: --none; this segment comprises other
--segments rather than actual data
--elements. It includes the following
--segments:

correlation-information

report-source

user-error-report

provider-error-report

Segment Name: estimate-results

Segment Code: ESR

Data Elements: --none; this segment comprises other
--segments rather than actual data
--elements. It includes the following
--segments:

cost-estimate

location-info

--this last subordinate segment may be
--repeated

Segment Name: estimated-date-available

Segment Code: EDA

Data Elements: estimated-date-available

Segment Name: extension

Segment Code: EXT

Data Elements: --none; this segment comprises other
--segments rather than actual data
--elements. It includes the following
--segments:

extension-identifier

extension-critical

--this segment will also contain one or
--more other segments which encode the
--data elements of the protocol extension,
--i.e. of the type "item";
-- the segments which may be included
-- here are defined by the value of
-- "extension-identifier".

-- No extensions are defined for version 1
-- or 2 of the protocol

Segment Name: extension-critical

Segment Code: ECR

Data Elements: critical

Segment Name: extension-identifier

Segment Code: EID

Date Elements: identifier

Segment Name: forward-flag

Segment Code: FWD

Data Elements: forward-flag

Segment Name: forward-note

Segment Code: FWN

Data Elements: forward-note

Segment Name: history-report

Segment Code: HRP

Data Elements: date-requested

author

title

author-of-article

title-of-article

date-of-last-transition

most-recent-service

date-of-most-recent-service

--the next two data elements together form
--the initiator-of-most-recent-service:

person-or-institution-symbol

{
person-symbol
institution-symbol
--only one of these sub-elements may
--be present
}

name-of-person-or-institution

{
name-of-person
name-of-institution
--only one of these sub-elements may
--be present