



INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

Q.773

(06/97)

SERIES Q: SWITCHING AND SIGNALLING

Specifications of Signalling System No. 7 – Transaction
capabilities application part

Transaction capabilities formats and encoding

ITU-T Recommendation Q.773

(Previously CCITT Recommendation)

ITU-T Q-SERIES RECOMMENDATIONS
SWITCHING AND SIGNALLING

SIGNALLING IN THE INTERNATIONAL MANUAL SERVICE	Q.1–Q.3
INTERNATIONAL AUTOMATIC AND SEMI-AUTOMATIC WORKING	Q.4–Q.59
FUNCTIONS AND INFORMATION FLOWS FOR SERVICES IN THE ISDN	Q.60–Q.99
CLAUSES APPLICABLE TO ITU-T STANDARD SYSTEMS	Q.100–Q.119
SPECIFICATIONS OF SIGNALLING SYSTEMS No. 4 AND No. 5	Q.120–Q.249
SPECIFICATIONS OF SIGNALLING SYSTEM No. 6	Q.250–Q.309
SPECIFICATIONS OF SIGNALLING SYSTEM R1	Q.310–Q.399
SPECIFICATIONS OF SIGNALLING SYSTEM R2	Q.400–Q.499
DIGITAL EXCHANGES	Q.500–Q.599
INTERWORKING OF SIGNALLING SYSTEMS	Q.600–Q.699
SPECIFICATIONS OF SIGNALLING SYSTEM No. 7	Q.700–Q.849
General	Q.700
Message transfer part (MTP)	Q.701–Q.709
Signalling connection control part (SCCP)	Q.711–Q.719
Telephone user part (TUP)	Q.720–Q.729
ISDN supplementary services	Q.730–Q.739
Data user part	Q.740–Q.749
Signalling System No. 7 management	Q.750–Q.759
ISDN user part	Q.760–Q.769
Transaction capabilities application part	Q.770–Q.779
Test specification	Q.780–Q.799
Q3 interface	Q.800–Q.849
DIGITAL SUBSCRIBER SIGNALLING SYSTEM No. 1	Q.850–Q.999
General	Q.850–Q.919
Data link layer	Q.920–Q.929
Network layer	Q.930–Q.939
User-network management	Q.940–Q.949
Stage 3 description for supplementary services using DSS 1	Q.950–Q.999
PUBLIC LAND MOBILE NETWORK	Q.1000–Q.1099
INTERWORKING WITH SATELLITE MOBILE SYSTEMS	Q.1100–Q.1199
INTELLIGENT NETWORK	Q.1200–Q.1999
BROADBAND ISDN	Q.2000–Q.2999

For further details, please refer to ITU-T List of Recommendations.

ITU-T RECOMMENDATION Q.773**TRANSACTION CAPABILITIES FORMATS AND ENCODING****Summary**

This Recommendation has been revised for some corrections to the ASN.1 and tabular descriptions of TC messages and parameters. It also provides, in Annex A, a description of the TC using the ASN.1 notation defined in Recommendations Q.680 to Q.683 and some definitions from Recommendation X.880.

Source

ITU-T Recommendation Q.773 was revised by ITU-T Study Group 11 (1997-2000) and was approved under the WTSC Resolution No. 1 procedure on the 5th of June 1997.

FOREWORD

ITU (International Telecommunication Union) is the United Nations Specialized Agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of the ITU. The ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Conference (WTSC), which meets every four years, establishes the topics for study by the ITU-T Study Groups which, in their turn, produce Recommendations on these topics.

The approval of Recommendations by the Members of the ITU-T is covered by the procedure laid down in WTSC Resolution No. 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

INTELLECTUAL PROPERTY RIGHTS

The ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. The ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, the ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

© ITU 1998

All rights reserved. No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the ITU.

CONTENTS

	Page
1 Introduction	1
2 Description conventions	1
3 Abstract syntax description	1
3.1 TC-Messages	1
3.2 Dialogue portion	4
3.2.1 Structured dialogue	4
3.2.2 Unstructured dialogue	6
4 Message representation.....	7
4.1 Encoding rules	7
4.1.1 Specification of encoding rules	7
4.1.2 Overview of encoding rules	7
4.1.3 Transmission order	11
4.2 Message encoding	12
4.2.1 Transaction Portion	12
4.2.2 Component Portion	17
4.2.3 Dialogue Portion	22
Annex A.....	30

Recommendation Q.773**TRANSACTION CAPABILITIES FORMATS AND ENCODING***(Melbourne 1988; revised in 1993 and 1997)***1 Introduction**

This Recommendation provides the format and encoding of Transaction Capabilities (TC) messages. These encoding rules are described in 4.1 and are based on a consistent subset of the encoding rules specified in Recommendation X.209 as contained in the *Blue Book*.

2 Description conventions

This Recommendation uses the Abstract Syntax Notation One (ASN.1) defined in Recommendation X.208 and the description method of other Q.700-Series Recommendations (tabular method). In the case of misalignment between the tabular and the ASN.1 description, the latter takes precedence over the tabular representation.

3 Abstract syntax description**3.1 TC-Messages**

The following module defines the type of TC messages:

TCAPMessages { ccitt recommendation q 773 modules (2) messages (1) version2 (2) }

DEFINITIONS ::=

BEGIN

EXPORTS OPERATION, ERROR, Component, InvokeId Type;

-- Transaction Portion fields

MessageType ::=	CHOICE {		
	unidirectional	[APPLICATION 1]	IMPLICIT Unidirectional,
	begin	[APPLICATION 2]	IMPLICIT Begin,
	end	[APPLICATION 4]	IMPLICIT End,
	continue	[APPLICATION 5]	IMPLICIT Continue,
	abort	[APPLICATION 7]	IMPLICIT Abort }
Unidirectional ::=	SEQUENCE	{ dialoguePortion components	DialoguePortion OPTIONAL, ComponentPortion }
Begin ::=	SEQUENCE	{ otid dialoguePortion components	OrigTransactionID, DialoguePortion OPTIONAL ComponentPortion OPTIONAL }
End ::=	SEQUENCE	{ dtid dialoguePortion components	DestTransactionID, DialoguePortion OPTIONAL, ComponentPortion OPTIONAL }

Continue ::= SEQUENCE { otid OrigTransactionID,
dtid DestTransactionID,
dialoguePortion DialoguePortion OPTIONAL,
components ComponentPortion OPTIONAL }

Abort ::= SEQUENCE {
dtid DestTransactionID,
reason CHOICE
{ p-abortCause P-AbortCause,
u-abortCause DialoguePortion } OPTIONAL
}

-- NOTE – When the Abort Message is generated by the Transaction sublayer, a p-Abort Cause must be present. The u-abortCause may be generated by the component sublayer in which case it is an ABRT APDU, or by the TC-User in which case it could be either an ABRT APDU or data in some user-defined abstract syntax.

DialoguePortion ::= [APPLICATION 11] EXTERNAL

-- The dialogue portion carries the dialogue control PDUs as value of the external data type.
-- The direct reference should be set to { ccitt recommendation q 773 as (1) dialogue-as (1) version (1) }
-- if structured dialogue is used and to { ccitt recommendation q 773 as (1) unidialogue-as (2) version (1) }
-- if unstructured dialogue is used or any user defined abstract syntax name when only user information is carried (e.g. when user information is sent in a 1988 Abort message).

OrigTransactionID ::= [APPLICATION 8] IMPLICIT OCTET STRING (SIZE (1..4))

DestTransactionID ::= [APPLICATION 9] IMPLICIT OCTET STRING (SIZE (1..4))

P-AbortCause ::= [APPLICATION 10] IMPLICIT INTEGER {
unrecognizedMessageType (0),
unrecognizedTransactionID (1),
badlyFormattedTransactionPortion (2),
incorrectTransactionPortion (3),
resourceLimitation (4) }

-- COMPONENT PORTION. The last field in the transaction portion of the TCAP message is the Component Portion.
-- The Component Portion may be absent.

ComponentPortion ::= [APPLICATION 12] IMPLICIT SEQUENCE SIZE (1..MAX) OF Component

-- Component Portion fields

-- COMPONENT TYPE. Recommendation X.229 defines four Application Protocol Data Units (APDUs).
-- TCAP adds returnResultNotLast to allow for the segmentation of a result.

Component ::= CHOICE {
invoke [1] IMPLICIT Invoke,
returnResultLast [2] IMPLICIT ReturnResult,
returnError [3] IMPLICIT ReturnError,
reject [4] IMPLICIT Reject,
returnResultNotLast [7] IMPLICIT ReturnResult }

-- The Components are sequences of data elements.

Invoke ::= SEQUENCE {
invokeID InvokeIdType,
linkedID [0] IMPLICIT InvokeIdType OPTIONAL,
operationCode OPERATION,
parameter ANY DEFINED BY operationCode OPTIONAL }

-- ANY is filled by the single ASN.1 data type following the keyword *PARAMETER* or the keyword *ARGUMENT*
 -- in the type definition of a particular operation.

```
ReturnResult ::= SEQUENCE {
    invokeID          InvokeIdType,
    result            SEQUENCE {
        operationCode OPERATION,
        parameter     ANY DEFINED BY operationCode
    } OPTIONAL
}
```

-- ANY is filled by the single ASN.1 data type following the keyword *RESULT* in the type definition
 -- of a particular operation.

```
ReturnError ::= SEQUENCE {
    invokeID          InvokeIdType,
    errorCode         ERROR,
    parameter        ANY DEFINED BY errorCode OPTIONAL
}
```

-- ANY is filled by the single ASN.1 data type following the keyword *PARAMETER* in the type definition
 -- of a particular error.

```
Reject ::= SEQUENCE {
    invokeID CHOICE {
        derivable          InvokeIdType,
        not-derivable      NULL },
    problem CHOICE {
        generalProblem     [0] IMPLICIT GeneralProblem,
        invokeProblem      [1] IMPLICIT InvokeProblem,
        returnResultProblem [2] IMPLICIT ReturnResultProblem,
        returnErrorProblem [3] IMPLICIT ReturnErrorProblem }
}
```

InvokeIdType ::= INTEGER (-128..127)

-- OPERATIONS

-- Operations are specified with the *OPERATION MACRO*.
 -- When an operation is specified, the valid parameter set, results, and errors for that operation are indicated.
 -- Default values and optional parameters are permitted.

OPERATION MACRO ::=

BEGIN

```
TYPE NOTATION ::= Parameter Result Errors LinkedOperations
VALUE NOTATION ::= value (VALUE CHOICE {
    localValue INTEGER,
    globalValue OBJECT IDENTIFIER } )

Parameter ::= ArgKeyword NamedType | empty
ArgKeyword ::= "ARGUMENT" | "PARAMETER"
Result ::= "RESULT" ResultType | empty
Errors ::= "ERRORS" "{"ErrorNames"}" | empty
LinkedOperations ::= "LINKED" "{"LinkedOperationNames"}" | empty
ResultType ::= NamedType | empty
ErrorNames ::= ErrorList | empty
ErrorList ::= Error | ErrorList "," Error
Error ::= value (ERROR)
    -- shall reference an error value
| type -- shall reference an error type
    -- if no error value is specified
```

```

LinkedOperationNames ::= OperationList | empty
OperationList        ::= Operation | OperationList "," Operation
Operation             ::= value (OPERATION)
                        -- shall reference an operation value
                        | type -- shall reference an operation type if
                        -- no operation value is specified
NamedType             ::= identifier type | type
END

-- ERRORS

-- Errors are specified with the ERROR MACRO.
-- When an error is specified, the valid parameters for that error are indicated.
-- Default values and optional parameters are permitted.

ERROR MACRO           ::=

BEGIN
  TYPE NOTATION        ::= Parameter
  VALUE NOTATION       ::= value (VALUE CHOICE {
                        localValue INTEGER,
                        globalValue OBJECT IDENTIFIER } )
  Parameter            ::= "PARAMETER" NamedType | empty
  NamedType            ::= identifier type | type
END

-- PROBLEMS

GeneralProblem        ::= INTEGER {
                        unrecognizedComponent (0),
                        mistypedComponent (1),
                        badlyStructuredComponent (2) }

InvokeProblem         ::= INTEGER {
                        duplicateInvokeID (0),
                        unrecognizedOperation (1),
                        mistypedParameter (2),
                        resourceLimitation (3),
                        initiatingRelease (4),
                        unrecognizedLinkedID (5),
                        linkedResponseUnexpected (6),
                        unexpectedLinkedOperation (7) }

ReturnResultProblem   ::= INTEGER {
                        unrecognizedInvokeID (0),
                        returnResultUnexpected (1),
                        mistypedParameter (2) }

ReturnErrorProblem    ::= INTEGER {
                        unrecognizedInvokeID (0),
                        returnErrorUnexpected (1),
                        unrecognizedError (2),
                        unexpectedError (3),
                        mistypedParameter (4) }

END -- TCAPMessages

```

3.2 Dialogue portion

3.2.1 Structured dialogue

The following module defines the type DialoguePDU whose values form the abstract syntax for the Dialogue APDUs used for the structured dialogue.

DialoguePDUs { ccitt recommendation q 773 modules (2) dialoguePDUs(2) version1 (1) }

DEFINITIONS ::=

BEGIN

EXPORTS dialogue-as-id, DialoguePDU;

-- abstract syntax name for structured dialogue APDUs

dialogue-as-id OBJECT IDENTIFIER ::= { ccitt recommendation q 773 as (1)
dialogue-as (1) version1 (1) }

DialoguePDU ::= CHOICE {

dialogueRequest AARQ-apdu,
dialogueResponse AARE-apdu,
dialogueAbort ABRT-apdu }

AARQ-apdu ::= [APPLICATION 0]
protocol-version

application-context-name
user-information

IMPLICIT SEQUENCE {
[0] IMPLICIT BIT STRING { version1 (0) }
DEFAULT { version1 },
[1] OBJECT IDENTIFIER,
[30] IMPLICIT SEQUENCE OF EXTERNAL
OPTIONAL }

AARE-apdu ::= [APPLICATION 1]
protocol-version

application-context-name
result
result-source-diagnostic
user-information

IMPLICIT SEQUENCE {
[0] IMPLICIT BIT STRING { version1 (0) }
DEFAULT { version1 },
[1] OBJECT IDENTIFIER,
[2] Associate-result,
[3] Associate-source-diagnostic,
[30] IMPLICIT SEQUENCE OF EXTERNAL
OPTIONAL }

-- RLRQ PDU is currently not used.
-- It is included for completeness only.

RLRQ-apdu ::= [APPLICATION 2]
reason
user-information

IMPLICIT SEQUENCE {
[0] IMPLICIT Release-request-reason OPTIONAL,
[30] IMPLICIT SEQUENCE OF EXTERNAL
OPTIONAL }

-- RLRE PDU is currently not used.
-- It is included for completeness only

RLRE-apdu ::= [APPLICATION 3]
reason
user-information

IMPLICIT SEQUENCE {
[0] IMPLICIT Release-response-reason OPTIONAL,
[30] IMPLICIT SEQUENCE OF EXTERNAL
OPTIONAL }

ABRT-apdu ::= [APPLICATION 4]
abort-source
user-information

IMPLICIT SEQUENCE {
[0] IMPLICIT ABRT-source,
[30] IMPLICIT SEQUENCE OF EXTERNAL
OPTIONAL }

ABRT-source ::= INTEGER {

dialogue-service-user (0),
dialogue-service-provider (1) }

Associate-result ::= INTEGER {

accepted (0),
reject-permanent (1) }

```

Associate-source-diagnostic ::= CHOICE {
    dialogue-service-user [1] INTEGER {
        null (0),
        no-reason-given (1),
        application-context-name-not-supported (2) },
    dialogue-service-provider [2] INTEGER {
        null (0),
        no-reason-given (1),
        no-common-dialogue-portion (2) }
}

```

-- Release-request-reason is currently not used.
 -- It is included for completeness only.

```

Release-request-reason ::= INTEGER { normal (0),
                                     urgent (1),
                                     user-defined (30)
                                   }

```

-- Release-response-reason is currently not used.
 -- It is included for completeness only.

```

Release-response-reason ::= INTEGER { normal (0),
                                     not-finished (1),
                                     user-defined (30) }

```

END -- DialoguePDUs

3.2.2 Unstructured dialogue

The following module defines the type UnidialoguePDU whose values form the abstract syntax for the dialogue APDUs used for the unstructured dialogue.

UnidialoguePDUs { ccitt recommendation q 773 modules (2) unidialoguePDUs (3) version1 (1) }

DEFINITIONS ::=

BEGIN

EXPORTS uniDialogue-as-id, UniDialoguePDU;

-- Abstract syntax name for unstructured dialogue APDUs

```

uniDialogue-as-id OBJECT IDENTIFIER ::= { ccitt recommendation q 773 as (1)
                                         unidialogue-as (2) version1 (1) }

```

```

UniDialoguePDU ::= CHOICE { unidialoguePDU
                             AUDT-apdu }

```

```

AUDT-apdu ::= [APPLICATION 0]
  protocol-version      IMPLICIT SEQUENCE {
                        [0] IMPLICIT BIT STRING {version1 (0) }
                        DEFAULT { version1 },
                        [1] OBJECT IDENTIFIER,
                        [30] IMPLICIT SEQUENCE OF EXTERNAL
                        OPTIONAL }
  application-context-name
  user-information

```

END -- UNIDialoguePDU

4 Message representation

A TCAP message is structured as a single constructor information element. It consists of a Transaction Portion which contains information elements used by the Transaction sublayer, and a Component Portion which contains information elements used by the Component sublayer related to components and, optionally, the Dialogue Portion which contains the Application Context and user information, which are not components. One of the Transaction Portion elements is called the Component Portion, and it contains the Component sublayer information elements. Each Component is a constructor information element.

4.1 Encoding rules

4.1.1 Specification of encoding rules

The encoding rules for TC messages are those described in Recommendation X.209 with the following restrictions:

- when the definite form is used for length encoding, a data value of length less than 128 octets must have the length encoded in the short form;
- when the long form is employed to code a length, the minimum number of octets shall be used to code the length field;
- OCTET STRING values and BIT STRING values must be encoded in a primitive form.

These encoding rules apply to TC messages as well as to the other abstract syntaxes defined in this Recommendation.

4.1.2 Overview of encoding rules

The following subclauses summarize the principles of these encoding rules. In case of misalignment between these subclauses and 4.1.1, the latter takes precedence.

4.1.2.1 General message structure

Each information element within a TCAP message has the same structure. An information element consists of three fields, which always appear in the following order. The Tag distinguishes one type from another and governs the interpretation of the Contents. The Length specifies the length of the Contents. The Contents is the substance of the element, containing the primary information the element is intended to convey. Figure 1 shows an overview of a TCAP message and an information element.

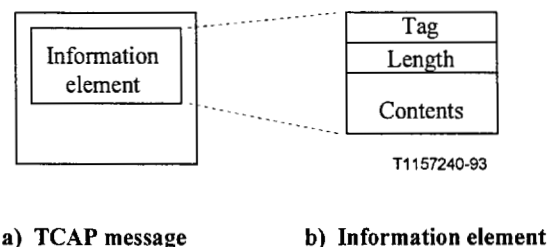


Figure 1/Q.773 – Structure of TCAP message and information element

Each field is coded using one or more octets. Octets are labelled as shown in Figure 2. Bits in an octet are labelled as shown in Figure 3, with bit A the least significant.