INTERNATIONAL TELECOMMUNICATION UNION

ITU-T Q.772

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

(03/93)

SPECIFICATIONS OF SIGNALLING SYSTEM No. 7

SIGNALLING SYSTEM No. 7 – TRANSACTION CAPABILITIES INFORMATION ELEMENT DEFINITIONS

ITU-T Recommendation Q.772

(Previously "CCITT Recommendation")

FOREWORD

The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of the International Telecommunication Union. 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, established the topics for study by the ITU-T Study Groups which, in their turn, produce Recommendations on these topics.

ITU-T Recommendation Q.772 was revised by the ITU-T Study Group XI (1988-1993) and was approved by the WTSC (Helsinki, March 1-12, 1993).

NOTES

1 As a consequence of a reform process within the International Telecommunication Union (ITU), the CCITT ceased to exist as of 28 February 1993. In its place, the ITU Telecommunication Standardization Sector (ITU-T) was created as of 1 March 1993. Similarly, in this reform process, the CCIR and the IFRB have been replaced by the Radiocommunication Sector.

In order not to delay publication of this Recommendation, no change has been made in the text to references containing the acronyms "CCITT, CCIR or IFRB" or their associated entities such as Plenary Assembly, Secretariat, etc. Future editions of this Recommendation will contain the proper terminology related to the new ITU structure.

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

CONTENTS

SIGNALLING SYSTEM No. 7 – TRANSACTION CAPABILITIES INFORMATION ELEMENT DEFINITIONS

(Melbourne 1988; modified at Helsinki 1993)

1 General

This Recommendation describes the individual information elements and parameters used within Transaction Capabilities messages. The encoding and formatting of these elements are shown in Recommendation Q.773.

The meaning of each information element is described in general terms.

The TCAP message format consists of three parts, namely the transaction portion, the dialogue portion and the component portion. Information in the component portion concerns individual operations and their replies. The transaction portion contains protocol control information for the transaction sublayer. The dialogue portion is concerned with the application context and, as an option, user information (i.e. data, which are not components).

For a more detailed analysis of the architecture, see Figure 1/Q.771 and associated text.

2 Transaction portion

The transaction portion of a TC message may contain the following information elements, namely:

2.1 Message type

Five types of messages are defined for the transaction portion as follows:

2.1.1 **unidirectional**: This message is used when there is no need to establish a transaction with another peer TR-User.

2.1.2 **begin**: This message is used to initiate a transaction with another peer TR-User.

2.1.3 **end**: This message is used to terminate a transaction with another peer TR-User.

2.1.4 **continue**: This message is used to complete the establishment of a transaction and to continue an established transaction.

2.1.5 **abort**: This message is used to terminate a transaction following an abnormal situation detected by the transaction sublayer (the service provider), or to abort a transaction by the TR-User (the service user).

2.2 Transaction IDs

Transaction IDs are independently assigned by each of the two nodes communicating via a transaction, enabling each node to uniquely identify the transaction and associate the entire contents of the message with that particular transaction. There are two types of Transaction IDs, namely:

2.2.1 **originating transaction ID**: The Originating Transaction ID is assigned by the node sending a message, and is used to identify the transaction at that end.

2.2.2 **destination transaction ID**: The Destination Transaction ID identifies the transaction at the receiving end.

2.3 P-abort cause: This is used when the transaction sub-layer aborts a transaction.

P-Abort cause definitions are as follows:

2.3.1 unrecognized message type: The message type is not one of those defined in 2.1.1 to 2.1.5 above.

2.3.2 unrecognized transaction ID: A transaction ID has been received for which a transaction does not exist at the receiving node.

2.3.3 badly formatted transaction portion: The transaction portion of the received message does not conform to the X.209 encoding rules as outlined in 4.1/Q.773.

2.3.4 incorrect transaction portion: The elemental structure within the transaction portion of the received message, does not conform to the rules for the transaction portion defined in 3.1/Q.773.

2.3.5 resource limitation: Sufficient resources are not available to start a transaction.

TABLE 1/Q.772

Example mapping of P-Abort scenarios to P-Abort cause values

Transaction sub-layer	
P-Abort cause	Example reason
Unrecognized message type (syntax error) (The message type tag is unknown i.e. not defined in Recommendation Q.773)	The combination of class, form and value does not correspond to a known tag, i.e. message type is not Begin, Continue, End, Uni-directional or Abort
Unrecognized transaction ID. (A Transaction ID has been received which is derivable, but for which a transaction does not exist at the receiving node, i.e. a Continue message has been received with an unrecognized (destination) Transaction ID.)	Destination Transaction ID unassigned
Badly formatted transaction portion (encoding error) (The Transaction Sub-layer has received a message which does not conform to the encoding rules defined in 4.1/Q.773)	Length indicator value has less than 128 octets, but not coded short form
	Malformed tag for an information element, other than Message Type (e.g. the class and code indicates integer while the form indicates constructed encoding)
	Length indicator value does not correspond to length of message
Incorrect transaction portion (syntax error) (The elemental structure within the Transaction portion of the received message does not conform to the rules for the Transaction Portion defined in 3/Q.773)	Combination of Origin and Destination Transactions ID does not conform to message type
	Component Portion Tag present, but no components
	Message does not contain all the mandatory information elements defined in Recommendation Q.773 for the message type
	The order of the received information elements within the message does not conform to Recommendation Q.773 for the message type
Resource limitation (Insufficient resources for this TR and/or TR-User)	Congestion
	No Transaction ID are available for allocation to the new transaction establishment request

2.4 **dialogue portion**: This is used to pass Information related to application context and, as an option, user information (i.e. data, which are not components).

2.5 **component portion**: This contains components. When no components are transferred, this information element is not present.

3 Component Portion

The Component Portion contains the following types of information elements. Components within a message are delivered to the user at the receiving end in the same order in which they were received from the user at the originating end.

3.1 Component type

There are five types of components that may be present in the Component Portion of a TC message. The four Protocol Data Units (PDUs) defined in Recommendation X.229 are used, namely:

TCAP component	X.229 PDU
Invoke	ROIV
Return result (last)	RORS
Return error	ROER
Reject	RORJ

The remaining component type – Return Result (Not Last) – is defined by TCAP.

These component types are defined as follows:

3.1.1 **invoke**: The invoke component requests that an operation be performed. It may be linked to another operation invocation previously sent by the other end. In this case it is known as a "Linked Invoke".

3.1.2 **return result (not last)**: When TC uses a connectionless Network Service, it may be necessary for the TC-User to segment the result of an operation if the two peer TC-users use a network service that does not provide segmenting/reassembling of user data. In this case the Return Result (Not Last) component is used to convey each segment of the result except the last, which is conveyed in a Return Result (Last) component.

The Return Result (Not Last) facility is allowed ONLY if the result is too large to fit into a single Return Result (Last) component. Note that the use of the Return Result (Not Last) facility implies that the operation has completed successfully.

3.1.3 **return result (last)**: The Return Result (Last) component reports successful completion of an operation. It may contain the last segment of a result or, in the case of an unsegmented result, it contains the entire result.

3.1.4 **return error**: The Return Error component reports that an operation has not been successfully completed.

3.1.5 **reject**: The Reject component reports the receipt and rejection of an incorrect component, other than a Reject component. The possible causes for rejecting a component are defined by the Problem Code element in 3.7.

3.2 **invoke ID**: An Invoke ID is used as a reference number to identify uniquely an operation invocation. It is present in the Invoke component and in any reply to the Invoke (Return Result, Return Error or Reject), enabling the reply to be correlated with the invoke.

3.3 linked ID: A Linked ID is included in an invoke component by a node when it responds to an operation invocation with a linked operation invocation. The node receiving the Linked ID uses it for correlation purposes, in the same way that it uses the invoke ID in Return Result, Return Error and Reject components.

3.4 operation code: The Operation Code element indicates the precise operation to be invoked, and is present in an Invoke component type. It is also present in the Return Result (Last/Not Last) components if the results contain parameters.

The operation code may be given a local value (i.e. integer) which then identifies the operation within a limited domain; or it may be a global value (i.e. object identifier) which makes the operation uniquely identifiable across all applications.

The actual operation codes, the definition of the operations and their associated parameters, are defined in relevant ASE specifications. The component sublayer does not set or examine the operation code value, nor parameters which are present, nor the parameter values.

3.5 parameter: The Parameter element contains one or several user information elements accompanying a component. The information elements themselves are defined in relevant ASE specifications.

3.6 error code: The Error Code element contains the reason why an operation cannot be completed successfully. It is present only in a Return Error component. As with operations, errors may be local or global. These errors and associated parameters are defined in relevant ASE specifications.

3.7 problem code: The Problem code element contains the reason for the rejection of a component, and one such element is present in a Reject component. Four problem code elements are defined, namely:

3.7.1 general problem: This element contains one of the problem codes which apply to the component sub-layer in general, and which do not relate to any specific component type. All of these are generated by the component sub-layer. They are:

3.7.1.1 unrecognized component: The component type is not recognized as being one of those defined in 3.1.

3.7.1.2 mistyped component: The elemental structure of a component does not conform to the structure of that component as defined in 3.1/Q.773.

3.7.1.3 badly structured component: The contents of the component do not conform to the encoding rules defined in 4.1/Q.773.

3.7.2 invoke problem: This element contains one of the problem codes that relate only to the invoke component type. They are:

3.7.2.1 duplicate invoke ID: The invoke ID is that of a previously invoked operation which has not been completed. This code is generated by the TC-User.

3.7.2.2 unrecognized operation: The operation code is not one of those agreed by the two TC-User.

3.7.2.3 mistyped parameter: Signifies that the type of parameter in an invoke component is not that agreed by the two TC-Users.

3.7.2.4 resource limitation: Sufficient resources are not available to perform the requested operation. This code is generated by the TC-User

3.7.2.5 initiating release: The requested operation cannot be invoked because the dialogue is about to be released. This code is generated only by the TC-User.

3.7.2.6 unrecognized linked ID: The linked ID does not correspond to an active invoke operation. This code is generated only by the component sub-layer.

3.7.2.7 linked response unexpected: The operation referred to by the linked ID is not an operation for which linked invokes are allowed. This code is generated only by the TC-User.

TABLE 2/Q.772

Example mapping of general problem reject scenarios to general problem reject types

Component sub-layer	
General problem	Example reason
Unrecognized Component (The component type is not recognized as being one of those defined in 3.1)	Component Type Tag not recognized as Invoke, Return Result Not Last, Return Error, Reject or Return Result Last
Mistyped Component (The elemental structure of a component does not conform to the structure of that component as defined in 3.1/Q.773)	Missing Invoke ID element
	Operation code element expected but not present
	Return Error Component received with no Error Code Element
	The order of the received information elements within the component does not conform to Recommendation Q.773 for that Component type
Badly structured Component (The contents of the component do not conform to the encoding rules defined in 4.1/Q.773)	Length indicator value less than 128 octets, but not coded short form

3.7.2.8 unexpected linked operation: The operation referred to by the linked ID does not allow this linked operation. This code is generated only by the TC-User.

3.7.3 return result problem: This element contains one of the problem codes which relate only to the return result component type. They are:

3.7.3.1 unrecognized invoke ID: No operation with the specified invoke ID is in progress. This code is generated by the component sub-layer.

3.7.3.2 return result unexpected: The invoked operation does not report success. This code is generated by the component sub-layer.

3.7.3.3 mistyped parameter: Signifies that the type of parameter in the return result component is not that agreed by the two TC-Users.

3.7.4 return error problem: This element contains one of the problem codes that relate only to the return error component type. They are:

3.7.4.1 unrecognized invoke ID: No operation with the specified invoke ID is in progress. This code is generated by the component sub-layer.

3.7.4.2 return error unexpected: The invoked operation does not report failure. This code is generated by the component sub-layer.

3.7.4.3 unrecognized error: The error code is not one of those agreed by the two TC-User.

3.7.4.4 unexpected error: The received error is not one of those that the invoked operation may report. This code is generated by the TC-User.

3.7.4.5 mistyped parameter: Signifies that the type parameter in a Return Error component is not that agreed by the two TC-Users.

4 Dialogue portion

The dialogue portion contains a dialogue control application protocol data unit (APDU) or user information.

4.1 Dialogue Control APDUs

Each dialogue control APDU defined is compatible with the OSI ACSE APDUs defined in Recommendation X.227.

The mapping between Dialogue Control APDUs and those of the OSI ACSE are shown below.

Dialogue Control APDU	ACSE APDU
Dialogue Request	AARQ
Dialogue Response	AARE
Dialogue Abort	ABRT
Dialogue Uni	AUDT

4.1.1 Dialogue Request (AARQ) APDU: The Dialogue Request (AARQ) APDU is used by the initiating TC-User at the start of a transaction to convey the Application Context Name and, as an option, user information (i.e. data, which are not components) to the peer TC-User.

4.1.2 Dialogue Response (AARE) APDU The Dialogue Response (AARE) APDU is used by the responding TC-User in the first backward message to inform the originating TC-User on whether or not the dialogue is accepted.

4.1.3 Dialogue Abort (ABRT) APDU: The Dialogue Abort (ABRT) APDU is used by the component sublayer to inform its peer of the receipt of an abnormal (syntactically invalid or inopportune) dialogue portion APDU. It is also used by the TC-Users to terminate a dialogue due to an abnormal situation.

4.1.4 Dialogue Uni (AUDT) APDU: The Dialogue Uni (AUDT) APDU is used to convey the Application Context Name and, as an option, user information (i.e. data, which are not components) for the situation where there is no need to establish a dialogue between two TC-User.

4.2 Dialogue Portion Information Elements

4.2.1 application context name: This parameter, of the form OBJECT IDENTIFIER, is a reference to an explicitly defined set of the TC-User Application Service Elements (ASEs), related options and any other necessary information for the interworking of two TC-Users during an instance of communication.

4.2.2 protocol version: The protocol version information element indicates the versions of the Dialogue Portion that can be supported. It is a bitstring, where each bit that is set to one indicates the version of the dialogue portion that is supported. Bit 0 represents version 1, bit 1 represents version 2, etc. The last bit set to one in the bitstring is the highest selected version. When the Protocol version parameter is absent, it implies “version 1” which is the version corresponding to this Recommendation.

4.2.3 user information: User information corresponds to any information exchanged between two TC-Users. Its meaning depends on the Application Context Name that accompanies it or is in place during its use. For example, this parameter may be used to carry information that further refines the application context by providing the "versions" of the ASEs that are referenced, "initialization" information on the ASEs, etc. Its meaning and use are therefore outside the scope of these Recommendations.

4.2.4 result: This parameter is set by the component sublayer to provide the initiating TC-User with the result of the request to establish a dialogue. Its value is set based on the dialogue handling primitive (and its accompanying parameters) used by the responding TC-User in response to the request for a dialogue. It takes the values "accepted" or "rejected (permanent)". The use of the value "rejected (transient)" is for further study.

4.2.5 result source diagnostic: This parameter identifies the creating source of the Result parameter and qualifies the result with some diagnostic information. The value of this parameter is set by the component sublayer and takes the symbolic values "dialogue service user" or "dialogue service provider". If the Result parameter takes the value "accepted", this parameter's value is set to "dialogue service user".

The "dialogue service user" can further qualify the result with a diagnostic with values of "null" or "no reason given" (for the case where no diagnostic is offered) or "application-context-name-not-supported" (for the case when the dialogue is refused). The "dialogue service provider" can further qualify the result with a diagnostic with values of "null" or "no reason given" (for the case where no diagnostic is offered) or "no common-dialogue-portion" (for the case, supporting the future evolution of these Recommendations, when the dialogue portions of the peer TC are different).

4.2.6 abort source: This parameter identifies whether the abnormal release of the dialogue is due to a request by the TC-User or the initiated by the dialogue portion for which it takes the values, respectively, of "dialogue service user" or "dialogue service provider".

Printed in Switzerland

Geneva, 1994